

**Oracle® Coherence**

Management Guide

Release 3.7.1

**E22842-01**

September 2011

Explains key management concepts and provides instructions for using Coherence management features.

Oracle Coherence Management Guide, Release 3.7.1

E22842-01

Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

Primary Author: Joe Ruzzi

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 is software or related documentation that 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 or hardware 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 that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware 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

<b>Preface</b> .....	vii
Audience .....	vii
Documentation Accessibility .....	vii
Related Documents .....	vii
Conventions .....	viii
<b>1 Introduction to Coherence Management</b>	
<b>Conceptual Overview of Coherence Management</b> .....	1-1
<b>Management Reporting in Coherence</b> .....	1-2
<b>Overview of Management Configuration</b> .....	1-2
<b>Managing Coherence with Enterprise Manager</b> .....	1-3
<b>2 Using JMX to Manage Coherence</b>	
<b>Configuring JMX Management</b> .....	2-1
Enabling Remote JMX Management on a Cluster Member .....	2-1
Enabling Local JMX Management on a Cluster Member .....	2-2
Enabling JMX Management When Using the Startup Scripts .....	2-2
Stopping a Cluster Member from being Managed Remotely .....	2-3
Disabling JMX Management .....	2-3
Filtering MBeans .....	2-4
Configuring Management Refresh .....	2-4
Setting the Management Refresh Expiry .....	2-5
Setting the Management Refresh Policy .....	2-5
Setting the Management Refresh Timeout .....	2-6
<b>Accessing Coherence MBeans</b> .....	2-6
Accessing MBeans Using Java VisualVM .....	2-7
Accessing MBeans Using the JConsole Utility .....	2-7
Accessing MBeans Using the HTML Adapter Application .....	2-9
Setting Up the Coherence MBeanConnector .....	2-10
<b>3 Registering Custom MBeans</b>	
<b>Registering Custom MBeans Declaratively</b> .....	3-1
Creating an MBean Configuration File .....	3-2
Registering MBeans Using a Class Name .....	3-2
Registering MBeans From an MBean Factory .....	3-2

Registering MBeans Using a Query.....	3-3
Specifying a Different MBean Configuration Override File.....	3-3
Restricting MBeans to the Local MBean Server.....	3-3
<b>Registering Custom MBeans Programmatically.....</b>	<b>3-4</b>

## 4 Using JMX Reporting

Overview of JMX Reporting.....	4-1
Enabling JMX Reporting On a Cluster Member.....	4-2
Switching the Default Report Group Configuration File.....	4-3
Changing the Report Timestamp Format.....	4-4
Administering JMX Reporting Using the Reporter MBean.....	4-4
Running JMX Reporting in a Distributed Configuration.....	4-6

## 5 Creating Custom Reports

Overview of Creating Custom Reports.....	5-1
<b>Constructing a Report Configuration File.....</b>	<b>5-2</b>
Specifying General Report Characteristics.....	5-2
file-name Macros.....	5-2
file-name Macro Examples.....	5-3
Specifying Data Columns.....	5-3
How to Include an Attribute.....	5-3
How to Include Part of the Key.....	5-3
How to Include Information from Composite Attributes.....	5-4
How to Include Information from Multiple MBeans.....	5-4
Including Multiple MBean Information Example.....	5-4
How to Use Report Macros.....	5-5
How to Include Constant Values.....	5-5
Including Queries in a Report.....	5-6
Using Filters to Construct Reports.....	5-7
Using Functions to Construct a Report.....	5-9
Function Examples.....	5-9
Using Aggregates to Construct a Report.....	5-11
Aggregate Examples.....	5-11
Constructing Delta Functions.....	5-12
Delta Function Examples.....	5-12
<b>Creating a Custom Report Group Configuration File.....</b>	<b>5-13</b>
Specifying the Report Refresh Frequency.....	5-13
Specifying the Output Directory for Reports.....	5-13
Specifying the Report List.....	5-14
<b>Configuring Custom Reports to Generate.....</b>	<b>5-14</b>

## 6 Analyzing Reporter Content

Understanding the Cache Size Report.....	6-1
Understanding the Cache Usage Report.....	6-2
Understanding the Flash Journal Report.....	6-3
Understanding the Management Report.....	6-4

Understanding the Memory Status Report .....	6-4
Understanding the Network Health Detail Report .....	6-5
Understanding the Network Health Report .....	6-6
Understanding the Node List Report .....	6-7
Understanding the Proxy Report.....	6-8
Understanding the Ram Journal Report .....	6-8
Understanding the Service Report.....	6-9

## A Coherence MBeans Reference

MBean Index .....	A-1
CacheMBean.....	A-3
ClusterMBean .....	A-7
ClusterNodeMBean .....	A-9
ConnectionManagerMBean .....	A-14
ConnectionMBean.....	A-15
FlashJournalRM.....	A-17
ManagementMBean.....	A-19
PointToPointMBean .....	A-20
RamJournalRM .....	A-22
ReporterMBean.....	A-24
ServiceMBean .....	A-26
StorageManagerMBean.....	A-30
TransactionManagerMBean .....	A-33

## B Report File Configuration Reference

Report File Deployment Descriptor .....	B-1
Element Reference.....	B-3
column .....	B-4
filter .....	B-6
filters.....	B-7
params .....	B-8
query .....	B-9
report.....	B-11
report-config .....	B-12
row.....	B-13

## C Report Group Configuration Reference

Report Group Configuration Deployment Descriptor .....	C-1
Element Reference.....	C-3
init-param.....	C-4
init-params .....	C-5
report-config .....	C-6
report-group.....	C-7
report-list .....	C-8



Welcome to *Oracle Coherence Management Guide*. This document explains key management concepts and provides instructions for using Coherence management features.

## Audience

This guide is intended for the following audiences:

- Primary Audience – Application Developers and Administrators who want to learn how to manage a Coherence Environment
- Secondary Audience – System Architects who want to understand the options and architecture for managing Coherence.

The audience must be familiar with Java and JMX to use this guide. In addition, the examples in this guide require the installation and use of the Oracle Coherence product.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Related Documents

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

- *Oracle Coherence Administrator's Guide*
- *Oracle Coherence Developer's Guide*
- *Oracle Coherence Client Guide*
- *Oracle Coherence Getting Started Guide*

- *Oracle Coherence Integration Guide for Oracle Coherence*
- *Oracle Coherence Security Guide*
- *Oracle Coherence Tutorial for Oracle Coherence*
- *Oracle Coherence User's Guide for Oracle Coherence\*Web*
- *Oracle Coherence Java API Reference*
- *Oracle Coherence C++ API Reference*
- *Oracle Coherence .NET API Reference*
- *Oracle Coherence Release Notes for Oracle Coherence*

## Conventions

The following text conventions are used in this document:

<b>Convention</b>	<b>Meaning</b>
<b>boldface</b>	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.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

---

---

# Introduction to Coherence Management

This chapter provides an introduction to Coherence management features. Coherence management is implemented using Java Management Extensions (JMX). JMX is a Java standard for managing and monitoring Java applications and services. Refer to the following article for details on JMX:

<http://java.sun.com/developer/technicalArticles/J2SE/jmx.html>

The following sections are included in this chapter:

- [Conceptual Overview of Coherence Management](#)
- [Management Reporting in Coherence](#)
- [Overview of Management Configuration](#)
- [Managing Coherence with Enterprise Manager](#)

## Conceptual Overview of Coherence Management

Coherence provides a JMX framework to manage clusters. The framework relies on one or more cluster members that are configured to host an MBean server. The MBean servers are responsible for managing the managed objects of all the other cluster members. The framework allows access to management information from any cluster member and is fault tolerant should a JMX member fail. The management framework is disabled by default and must be explicitly enabled by configuring at least one cluster member to host an MBean server. See [Chapter 2, "Using JMX to Manage Coherence,"](#) for more information on enabling JMX management in Coherence.

### Coherence MBeans

Coherence managed objects are registered to an MBean server using the `com.tangosol.net.management.Registry` interface. The interface is specific to managing Coherence clustered resources and is an abstraction of the basic JMX registration APIs. Though the interface is closely related to the JMX infrastructure, it is independent from `javax.management.*` classes. The interface enables remote management support for cluster members that are not collocated with any JMX services and allows Coherence MBeans to be registered in an MBean server that is either collocated or remote in relation to the managed object.

[Appendix A, "Coherence MBeans Reference,"](#) provides a list of all Coherence MBeans and describes each of the attributes and operations that are exposed for the managed resources. Some managed resources only have a single instance for each cluster member; while, some managed resources (such as the `CacheMBean` MBean) may have multiple MBean instances for each cluster member. In addition, an MBean is only

registered if there is at least one managed resource operational. In the case of the CacheMBean MBean, a cache must be started before the MBean is registered.

### Custom MBeans

Coherence provides the ability to manage and monitor custom MBeans within the management framework. Custom MBeans are any dynamic or standard MBeans that are specific to an application. The MBeans are registered either declaratively in an XML file or programmatically using the `Registration` interface. This allows an application's MBeans to be managed or monitored from any JVM, member, or end-point within the cluster. See [Chapter 3, "Registering Custom MBeans,"](#) for more information on registering custom MBeans.

### MBean Consoles

Any MBean-capable console can interact with Coherence MBeans. Coherence includes support for both the Java Management & Monitoring Console (jconsole) that is distributed with the JDK and the JMX HTTPAdapter Web application that is included as part of the JMX reference implementation. See ["Accessing Coherence MBeans"](#) on page 2-6 for more information on using these consoles to interact with Coherence MBeans.

## Management Reporting in Coherence

The Coherence Reporter provides management reports that are used to view management information over time. The reports are text files that are constructed from data that is obtained from the Coherence MBeans. The reports are automatically updated at a configured time interval and provide a historical context that is not possible simply by monitoring the MBeans. The reports are most often used to identify trends that are valuable for troubleshooting and planning.

Coherence includes many predefined reports out-of-box. The reports can be customized or new reports can be created as required. Reporting functionality is disabled by default and must be explicitly enabled. In addition, only a subset of the reports are initially configured to be generated. See [Chapter 4, "Using JMX Reporting,"](#) for details on enabling reporting in Coherence. Also, see [Chapter 6, "Analyzing Reporter Content,"](#) for detailed information on the predefined reports.

## Overview of Management Configuration

Coherence management is configured using several configuration files. See *Oracle Coherence Developer's Guide* for detailed information on Coherence configuration. The files include:

- Operational Override File – The `tangosol-coherence-override.xml` file is used to override the operational deployment descriptor, which is used to specify the operational and run-time settings that are used to create, configure and maintain clustering, communication, and data management services. The management settings are defined within the `<management-config>` node. See *Oracle Coherence Developer's Guide* for more information on the available settings for management.
- MBean Configuration Override File – The `custom-mbeans.xml` file is the default MBean configuration override file and is used to declaratively define custom MBeans. Custom MBeans can also be defined within the operational override file. However, the MBean configuration override file is typically used instead.

- report configuration files – A set of report configuration files that each results in the creation of a report file that displays management information for a particular set of metrics. Report configuration files must be referenced in a report group configuration file to be used at run time. The default report configuration files are located in the `/reports` directory of the `coherence.jar` and are referenced by the default report group configuration file. Custom report configuration files can be created as required. See [Appendix B, "Report File Configuration Reference,"](#) for details about the report file configuration elements.
- report group configuration file – A report group configuration file is used to list the name and location of report definition files and the output directory where reports are written. The name and location of this file is defined in the operational deployment descriptor. By default, the `report-group.xml` file is used and is located in the `/reports` directory of the `coherence.jar`. Additional report group configuration file are provided and custom report group files can be created as required. See [Appendix C, "Report Group Configuration Reference,"](#) for details about the report group configuration elements.
- `management-config.xml` – The management configuration file is used to configure the management invocation service instance that is used by the Coherence JMX management framework. The file is located in the root of the `coherence.jar` file and can be overridden by placing a `management-config.xml` file in the classpath before the `coherence.jar` file. This configuration file is not defined by an XSD. The file must have a `<config>` root element and supports the same subelements that are available for an `<invocation-scheme>` element. The management configuration file settings are not typically modified and it is generally recommended to leave the default settings.

## Managing Coherence with Enterprise Manager

Oracle Enterprise Manager Grid Control includes the Management Pack for Oracle Coherence, which is used to manage and monitor Coherence clusters. The management pack helps administrators pro-actively monitor the performance of their Coherence clusters and reduces the time needed to identify and diagnose performance problems within their application environments. The key benefits include:

- Manage complexity by modeling the entire cluster as a single target
- Real-time and historical performance monitoring for caches and nodes for faster diagnostics and resolution times
- Monitor caches in context of applications to analyze dependency
- Proactive monitoring using thresholds and alerts
- Reduce risk using automated provisioning and lifecycle management
- Change run-time configuration to quickly tune cache performance

Oracle Enterprise Manager Grid Control, which includes the Management Pack for Oracle Coherence, can be downloaded from OTN:

<http://www.oracle.com/technetwork/oem/grid-control/downloads/index.html?ssSourceSiteId=ocomen>

See the *Getting Started Guide for Oracle Coherence* in the Oracle Enterprise Manager Documentation Library for detailed instructions on configuring and using the management pack.

<http://www.oracle.com/technetwork/oem/grid-control/documentation/index.html>

---

---

## Using JMX to Manage Coherence

This chapter provides instructions for enabling and configuring JMX management within a cluster and describes how to access Coherence MBeans.

The following sections are included in this chapter:

- [Configuring JMX Management](#)
- [Accessing Coherence MBeans](#)

### Configuring JMX Management

JMX management is configured within the `<management-config>` element in a `tangosol-coherence-override.xml` file or by setting management system properties at startup. See *Oracle Coherence Developer's Guide* for detailed information on each of the elements that can be used within the `<management-config>` element.

The following topics are included in this section:

- [Enabling Remote JMX Management on a Cluster Member](#)
- [Enabling Local JMX Management on a Cluster Member](#)
- [Enabling JMX Management When Using the Startup Scripts](#)
- [Stopping a Cluster Member from being Managed Remotely](#)
- [Disabling JMX Management](#)
- [Filtering MBeans](#)
- [Configuring Management Refresh](#)

### Enabling Remote JMX Management on a Cluster Member

Remote JMX management allows one or more cluster members to each host an MBean server that is responsible for the managed objects of all cluster members. Accessing the MBean servers on these cluster members shows management information for all cluster members. The use of dedicated JMX cluster members is a common pattern because it avoids loading JMX software into every single cluster member while still providing fault-tolerance should a single JMX member fail.

To enable remote JMX management on a cluster member, set the `<managed-nodes>` element to `all` or `remote-only`. For example:

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

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

```

xsi:schemaLocation="http://xmlns.oracle.com/coherence/
coherence-operational-config coherence-operational-config.xsd">
<management-config>
  <managed-nodes system-property="tangosol.coherence.management">all
  </managed-nodes>
</management-config>
</coherence>

```

The `remote-only` setting starts an MBean server that only manages remote MBeans. The `all` setting starts an MBean server that manages remote MBeans and local (within the same JVM) MBeans.

The `tangosol.coherence.management` system property is used to enable remote JMX management instead of using the operational override file. For example:

```
-Dtangosol.coherence.management=all
```

## Enabling Local JMX Management on a Cluster Member

Local JMX management constrains an MBean server to only manage the MBeans that are local (within the same JVM) to the cluster member. Accessing the MBean server on the cluster member only shows local management information. However, the member's MBeans can still be managed by a member that has been enabled for remote JMX management. Local JMX Management is typically used for extend clients or transient cluster clients.

To enable local JMX management on a cluster member, set the `<managed-nodes>` element to `local-only`. For example:

```

<?xml version='1.0'?>

<coherence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.oracle.com/coherence/coherence-operational-config"
  xsi:schemaLocation="http://xmlns.oracle.com/coherence/
  coherence-operational-config coherence-operational-config.xsd">
  <management-config>
    <managed-nodes system-property="tangosol.coherence.management">local-only
    </managed-nodes>
  </management-config>
</coherence>

```

The `tangosol.coherence.management` system property is used to enable local JMX management instead of using the operational override file. For example:

```
-Dtangosol.coherence.management=local-only
```

## Enabling JMX Management When Using the Startup Scripts

As a convenience, the `COHERENCE_HOME/bin/cache-server` and `COHERENCE_HOME/bin/coherence` startup scripts include a `-jmx` argument that enables JMX management on a cluster member and also enables access to the MBean server. For example:

```
cache-server -jmx
```

The argument automatically sets the following system properties, which can be changed as required within the script. The properties are, by default, set to enable

remote JMX management on the member and also allow the member to be managed remotely.

```
-Dcom.sun.management.jmxremote
-Dtangosol.coherence.management=all
-Dtangosol.coherence.management.remote=true
```

The `com.sun.management.jmxremote` system property is required to allow remote access to an MBean server.

## Stopping a Cluster Member from being Managed Remotely

By default, all cluster members allow their MBeans to be managed by a remote MBean server. To restrict remote management of a member's MBeans, the `<allow-remote-management>` element must be set to `false`. For example:

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

<coherence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.oracle.com/coherence/coherence-operational-config"
  xsi:schemaLocation="http://xmlns.oracle.com/coherence/
coherence-operational-config coherence-operational-config.xsd">
  <management-config>
    <allow-remote-management
      system-property="tangosol.coherence.management.remote">false
    </allow-remote-management>
  </management-config>
</coherence>
```

The `tangosol.coherence.management.remote` system property is used to disable remote management instead of using the operational override file. For example:

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

## Disabling JMX Management

To disable JMX management on a cluster member, set the `<managed-nodes>` element to `none`. For example:

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

<coherence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.oracle.com/coherence/coherence-operational-config"
  xsi:schemaLocation="http://xmlns.oracle.com/coherence/
coherence-operational-config coherence-operational-config.xsd">
  <management-config>
    <managed-nodes system-property="tangosol.coherence.management">none
  </managed-nodes>
  </management-config>
</coherence>
```

Disabling JMX management on a member does not stop the member from being remotely managed. The following example disables JMX Management and stops the member from being remotely managed:

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

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

```

xsi:schemaLocation="http://xmlns.oracle.com/coherence/
coherence-operational-config coherence-operational-config.xsd">
<management-config>
  <managed-nodes system-property="tangosol.coherence.management">none
  </managed-nodes>
  <allow-remote-management
    system-property="tangosol.coherence.management.remote">>false
  </allow-remote-management>
</management-config>
</coherence>

```

## Filtering MBeans

The Coherence management framework provides the ability to filter MBeans before they are registered in the MBean server. An out-of-the box MBean filter is provided and custom filters can be created as required. The included MBean filter (`com.tangosol.net.management.ObjectNameExcludeFilter`) is used to exclude MBeans from being registered based on their JMX object name using standard regex patterns. As configured out-of-box, the filter excludes some platform MBeans from being registered in the management framework. MBean filters are defined using the `<mbean-filter>` element.

The following example shows the out-of-the box configuration:

```

...
<mbean-filter>
  <class-name>com.tangosol.net.management.ObjectNameExcludeFilter</class-name>
  <init-params>
    <init-param>
      <param-type>string</param-type>
      <param-value system-property="tangosol.coherence.management.exclude">
        .*type=Service,name=Management,.*
        .*type=Platform,Domain=java.lang,subType=ClassLoader,.*
        .*type=Platform,Domain=java.lang,subType=Compilation,.*
        .*type=Platform,Domain=java.lang,subType=MemoryManager,.*
        .*type=Platform,Domain=java.lang,subType=Threading,.*
      </param-value>
    </init-param>
  </init-params>
</mbean-filter>
...

```

To enable the management service or platform MBeans, remove the corresponding object names from the list of names in the `<param-value>` element. To exclude an MBean from being registered, add the MBean object name to the list.

The `tangosol.coherence.management.exclude` system property is used to filter MBeans instead of using the operational override file. For example:

```
-Dtangosol.coherence.management.exclude=.*type=Service,name=Management,.*
```

## Configuring Management Refresh

The `<refresh-expiry>`, `<refresh-policy>`, and `<refresh-timeout>` elements control the latency of management information. For example:

```

<?xml version='1.0'?>
<coherence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

xmlns="http://xmlns.oracle.com/coherence/coherence-operational-config"
xsi:schemaLocation="http://xmlns.oracle.com/coherence/
coherence-operational-config coherence-operational-config.xsd">
<management-config>
  <refresh-policy
    system-property="tangosol.coherence.management.refresh.policy">
    refresh-ahead</refresh-policy>
  <refresh-expiry
    system-property="tangosol.coherence.management.refresh.expiry">1s
  </refresh-expiry>
  <refresh-timeout
    system-property="tangosol.coherence.management.refresh.timeout">250ms
  </refresh-timeout>
</management-config>
</coherence>

```

### Setting the Management Refresh Expiry

The `<refresh-expiry>` element specifies the minimum time interval between the remote retrieval of management information from remote members. The value of this element must be in the following format:

```
[\d]+[.[.][\d]+]?[MS|ms|S|s|M|m|H|h|D|d]?
```

The first non-digits (from left to right) indicate the unit of time duration:

- MS or ms (milliseconds)
- S or s (seconds)
- M or m (minutes)
- H or h (hours)
- D or d (days)

If the value does not contain a unit, a unit of milliseconds is assumed. The default value is 1s.

The `tangosol.coherence.management.refresh.expiry` system property is used to set the expiry instead of using the operational override file. For example:

```
-Dtangosol.coherence.management.refresh.expiry=2s
```

### Setting the Management Refresh Policy

The `<refresh-policy>` element is used to specify the method which is used to refresh remote management information. Each policy uses a different refresh algorithm that can improve latency based on MBean usage patterns. [Table 2-1](#) describes each policy.

**Table 2-1 Refresh Policies**

Setting	Description
<code>refresh-ahead</code> (default)	MBeans are refreshed before they are requested based on prior usage patterns after the expiry delay has passed. This setting can reduce latency of the management information with a minor increase in network consumption. This setting is best when MBeans are accessed in a repetitive/programmatic pattern.

**Table 2-1 (Cont.) Refresh Policies**

Setting	Description
<code>refresh-behind</code>	Each MBean is refreshed after the data is accessed. This method ensures optimal response time. However, the information returned is offset by the last refresh time.
<code>refresh-expired</code>	Each MBean is refreshed from the remote member when it is accessed and the expiry delay has passed from the last refresh. This setting is best used when MBeans are accessed in a random pattern.

The `tangosol.coherence.management.refresh.policy` system property is used to set the policy instead of using the operational override file. For example:

```
-Dtangosol.coherence.management.refresh.policy=refresh-expired
```

### Setting the Management Refresh Timeout

The `<refresh-time>` element is used to specify the duration which the management member waits for a response from a remote member when refreshing MBean information. This value must be less than the `<refresh-expiry>` value. The value of this element must be in the following format:

```
[\\d]+[\\.][\\d]+?[MS|ms|S|s|M|m|H|h|D|d]?
```

The first non-digits (from left to right) indicate the unit of time duration:

- MS or ms (milliseconds)
- S or s (seconds)
- M or m (minutes)
- H or h (hours)
- D or d (days)

If the value does not contain a unit, a unit of milliseconds is assumed. The default value is 250ms.

The `tangosol.coherence.management.refresh.timeout` system property is used to set the timeout instead of using the operational override file. For example:

```
-Dtangosol.coherence.management.refresh.timeout=500ms
```

## Accessing Coherence MBeans

JMX management must be enabled on a cluster member before accessing Coherence MBeans. See "[Configuring JMX Management](#)" on page 2-1. The procedures in this section use the Coherence startup scripts and the `-jmx` argument to enable JMX management.

The following topics are included in this section:

- [Accessing MBeans Using Java VisualVM](#)
- [Accessing MBeans Using the JConsole Utility](#)
- [Accessing MBeans Using the HTML Adapter Application](#)
- [Setting Up the Coherence MBeanConnector](#)

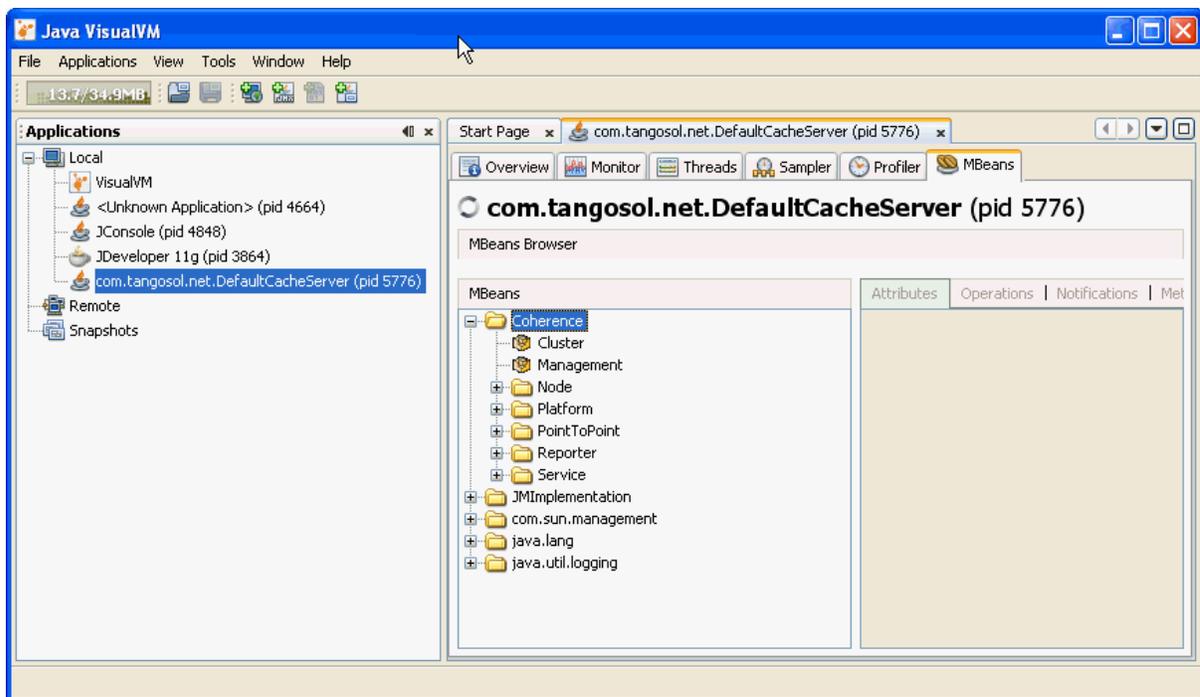
## Accessing MBeans Using Java VisualVM

Java Visual VM is a management utility included with the JDK (`JDK_HOME\bin\jvisualvm`) and is used to view and interact with Coherence MBeans.

To access Coherence MBeans using the Java VisualVM utility:

1. Start a cache server using the `COHERENCE_HOME\bin\cache-server` script and specify the `-jmx` argument to enable management on the member. For example:
 

```
COHERENCE_HOME\bin\cache-server -jmx
```
2. Start `JDK_HOME\bin\jvisualvm`. The Java VisualVM window displays.
3. From the Applications tree, click to expand **Local** if it is not already expanded and double-click the cluster member's process. The process information is displayed in a process tab on the right side of the window.
4. From the selected process tab, click the MBeans tab and expand the Coherence node to access the Coherence MBeans.



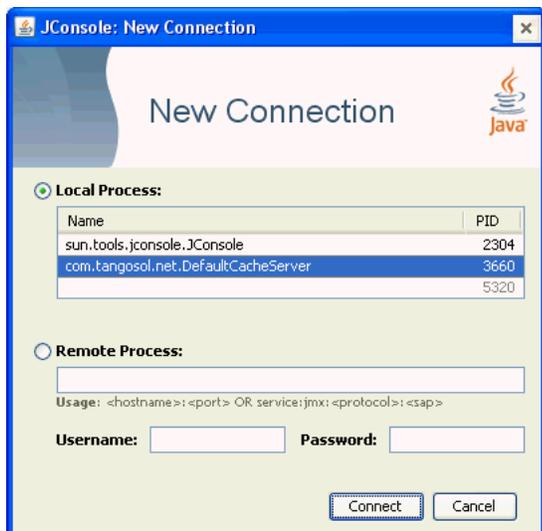
## Accessing MBeans Using the JConsole Utility

JConsole is a management utility that is included with the JDK (`JDK_HOME\bin\jconsole`) and is used to view and interact with Coherence MBeans.

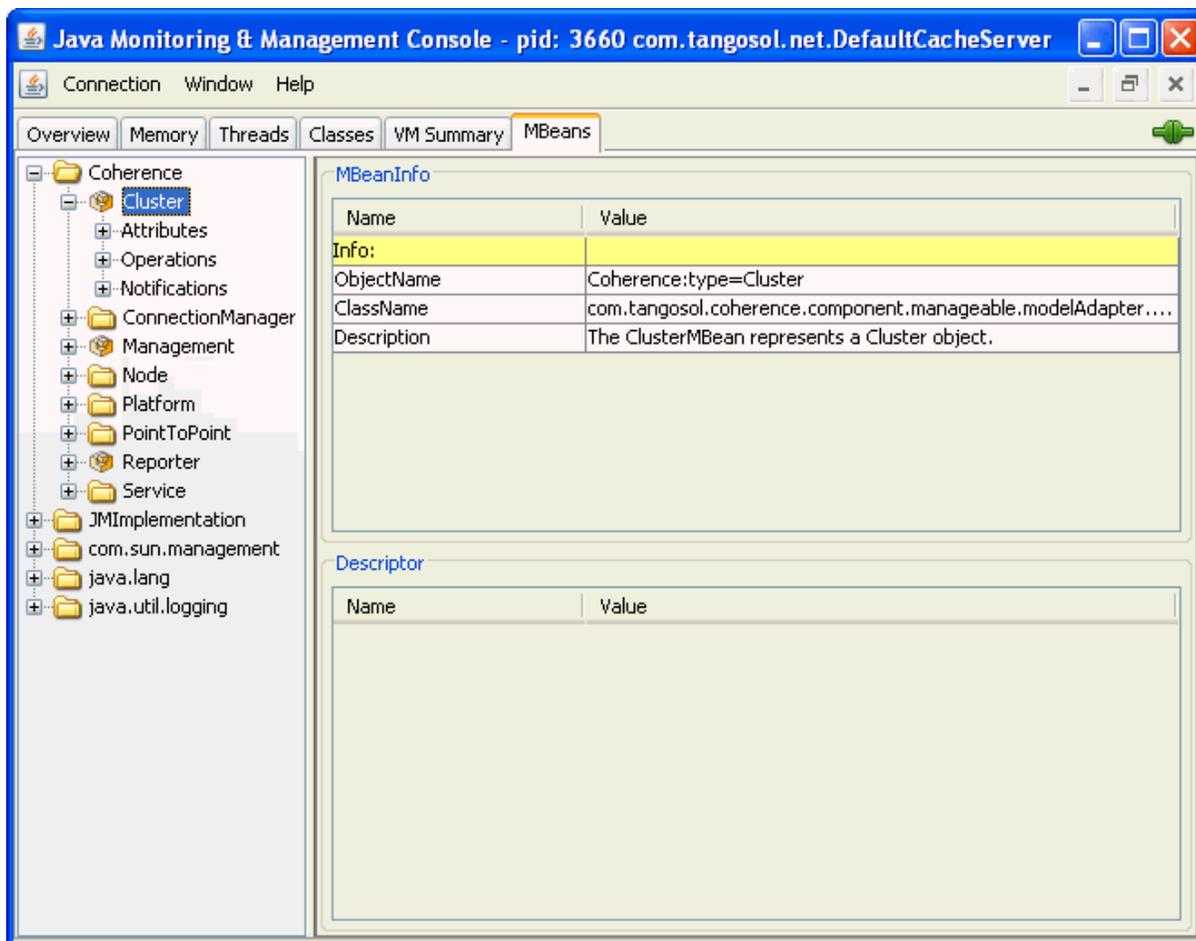
To access Coherence MBeans using the JConsole utility:

1. Start a cache server using the `COHERENCE_HOME\bin\cache-server` script and specify the `-jmx` argument to enable management on the member. For example:
 

```
COHERENCE_HOME\bin\cache-server -jmx
```
2. Start `JDK_HOME\bin\jconsole`. The Java Monitoring & Management Console window displays and the JConsole: New Connection dialog box displays.
3. From the JConsole: New Connection dialog box, click **Local Process:** and select the cluster member's process. For example:



4. Click **Connect**.
5. From the Java Monitoring & Management Console window, select the MBeans tab and expand the Coherence node to access the Coherence MBeans.



## Accessing MBeans Using the HTML Adapter Application

The HTML Adapter Web Application uses the HTTP adapter (`HtmlAdaptorServer`) that is shipped as part of the JMX reference implementation. The adapter requires both the `jmxri.jar` and `jmxtools.jar` libraries to be in the classpath. The JMX reference implementation can be downloaded from the following page:

<http://www.oracle.com/technetwork/java/javase/tech/download-jsp-141676.html>

To access Coherence MBeans using the HTML adapter:

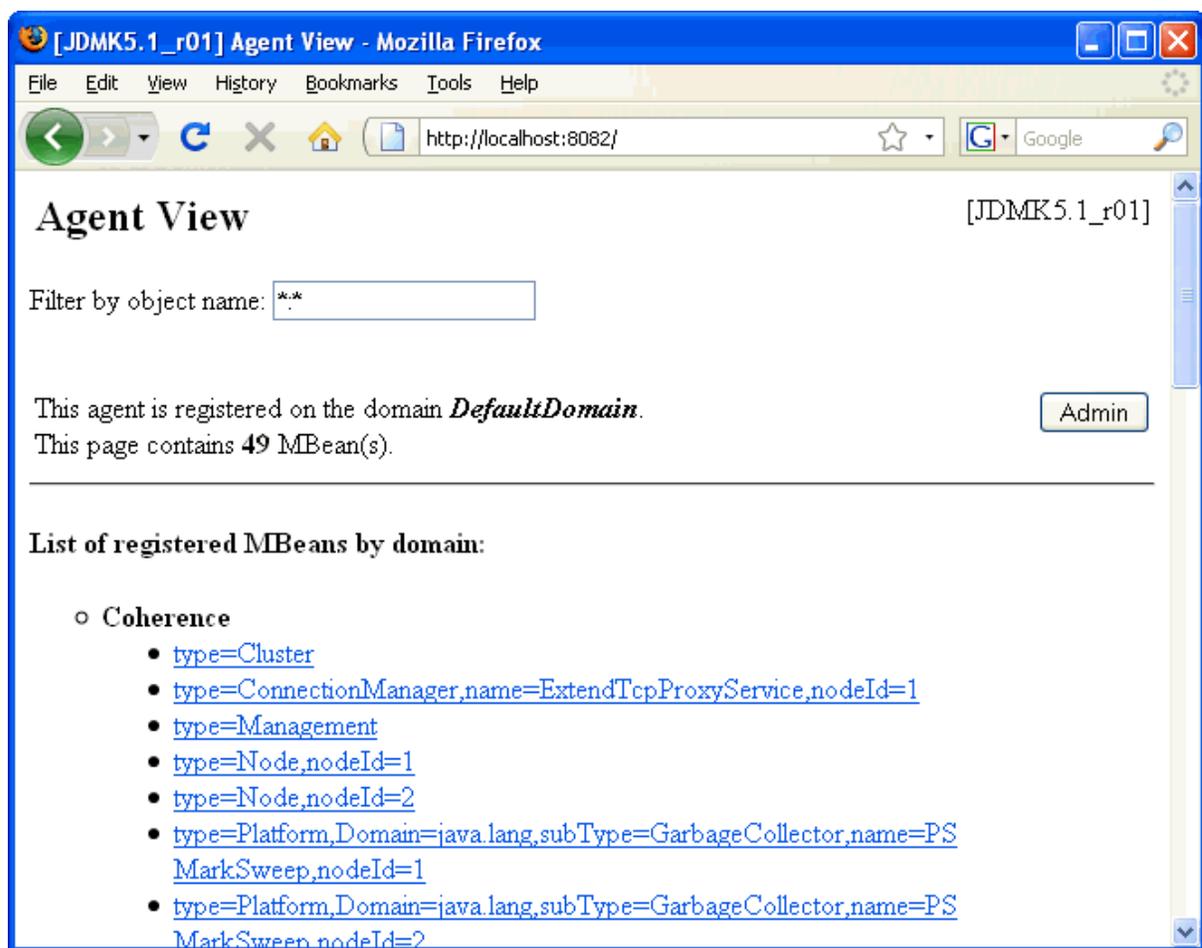
1. Edit the `COHERENCE_HOME\bin\coherence` script to include the `jmxri.jar` and `jmxtools.jar` in the classpath. For example, on Windows:

```
-cp "jmxri-1.2.1.jar;jmxtools-1.2.1.jar;%coherence_home%\lib\coherence.jar"
```

2. Start a cache factory instance using the script and specify the `-jmx` argument to enable management on the member. For example:

```
COHERENCE_HOME\bin\coherence -jmx
```

3. After the cache factory instance starts, enter `jmx 8082` at the command prompt. This starts an HTTP adapter at port 8082 on the cluster member:
4. Using a Web browser, access the adapter using the host name and port 8082 for the address.



## Setting Up the Coherence MBeanConnector

Coherence ships with a program to launch a cluster member as a dedicated MBean server host. This program provides access to Coherence MBeans by using the JMX Remote API using RMI or the HTTP server provided by Sun's JMX RI. The RMI and HTTP ports can be configured, allowing for access through a firewall. The server is started using the following command (note that it is broken up into multiple lines here only for formatting purposes; this is a single command entered on one line):

```
java -Dtangosol.coherence.management=all
      -Dcom.sun.management.jmxremote.ssl=false
      -Dcom.sun.management.jmxremote.authenticate=false
      -cp coherence.jar;jmxri.jar;jmxtools.jar
      com.tangosol.net.management.MBeanConnector [-http -rmi]
```

To allow access by using JMX RMI, include the `-rmi` flag. To allow access by using HTTP and a Web browser, include the `-http` flag. Both flags may be included; however at least one must be present for the member to start.

Table 2–2 describes optional properties that can be used for JMX RMI configuration:

**Table 2–2** *Optional Properties that can be used for JMX RMI Configuration*

Property	Description
<code>tangosol.coherence.management.remote.host</code>	The host that the JMX server binds to. Default is <code>localhost</code> . (NOTE: on Redhat Linux this may have to be changed to the host name or IP address)
<code>tangosol.coherence.management.remote.registryport</code>	The port used for the JMX RMI registry. Default is 9000.
<code>tangosol.coherence.management.remote.connectionport</code>	The port used for the JMX RMI connection. Default is 3000.

Table 2–3 describes optional properties that can be used for HTTP configuration.

**Table 2–3** *Optional Properties that can be used for Http Configuration*

Property	Description
<code>tangosol.coherence.management.remote.httpport</code>	The port used for the HTTP connection. Default is 8888.

To connect by using JConsole with default settings, use the following command:

```
jconsole service:jmx:rmi://localhost:3000/jndi/rmi://localhost:9000/server
```

To connect by using HTTP with default settings, use the following URL:

```
http://localhost:8888
```

---

**Note:** Refer to the JMX Agent documentation below to setup secure access using authentication and SSL:

<http://java.sun.com/javase/6/docs/technotes/guides/management/agent.html>

---

---

---

## Registering Custom MBeans

Coherence provides the ability to manage and monitor custom MBeans within the management framework. Custom MBeans are MBeans that are specific to an application (including MBeans for the Java platform ). Custom MBeans are registered to a Coherence MBean server either declaratively in an XML file or programmatically.

The ability to integrate custom MBeans with Coherence MBeans allows administrators to update and view system and application information for all members in a cluster from a single location.

The following sections are included in this chapter:

- [Registering Custom MBeans Declaratively](#)
- [Registering Custom MBeans Programmatically](#)

### Registering Custom MBeans Declaratively

Custom MBeans are registered declaratively within an MBean configuration override file named `custom-mbeans.xml`. At run time, Coherence uses the first instance of `custom-mbeans.xml` that is found in the classpath. See *Oracle Coherence Developer's Guide* for detailed information on configuration. MBeans are registered using either an MBean's class name, an MBean factory, or by running a query to locate MBeans.

---

---

**Note:** Custom MBeans must be found at run time. Make sure to place the MBeans (or the library that contains the MBeans) in the classpath of a Coherence member, including the JMX management-enabled member.

---

---

The following topics are included in this section:

- [Creating an MBean Configuration File](#)
- [Registering MBeans Using a Class Name](#)
- [Registering MBeans From an MBean Factory](#)
- [Registering MBeans Using a Query](#)
- [Restricting MBeans to the Local MBean Server](#)
- [Specifying a Different MBean Configuration Override File](#)

## Creating an MBean Configuration File

The `custom-mbeans.xml` file is used to override the `<mbeans>` element of the operational deployment descriptor. Therefore, the root element must be the `<mbeans>` element. See the *Oracle Coherence Developer's Guide* for a complete descriptions of the elements that can be defined within the `<mbeans>` element.

To create the MBean configuration override file:

1. Create a text file and save it as `custom-mbeans.xml`.
2. Edit the file and create an empty `<mbeans>` node as follows:

```
<mbeans>
</mbeans>
```

3. Save and close the file.
4. Make sure the location of the custom MBean configuration override file is located in the classpath at run time and precedes the `coherence.jar` library.

The following example demonstrates starting a cache server that uses a `custom-mbeans.xml` file that is located in `COHERENCE_HOME`.

```
java -cp COHERENCE_HOME;COHERENCE_HOME\lib\coherence.jar
com.tangosol.net.DefaultCacheServer
```

## Registering MBeans Using a Class Name

The `<mbean-class>` element is used to register and instantiate an MBean using the MBean's fully qualified class name. The following example registers an MBean named `com.MyMBean` and gives the MBean an object name `type=application`.

```
<mbeans>
  <mbean id="100">
    <mbean-class>com.MyMBean</mbean-class>
    <mbean-name>type=application</mbean-name>
    <enabled>>true</enabled>
  </mbean>
</mbeans>
```

As configured, the MBean can be managed remotely and includes the member where it is located as part of the object name. Therefore, the complete object name when the MBean is registered is:

```
Coherence:type=application,nodeId=<nodeId>
```

## Registering MBeans From an MBean Factory

The `<mbean-factory>` and `<mbean-accessor>` elements are used to register and instantiate an MBean from a factory class. The following example registers the Java platform's `MemoryMXBean` MBean using the `getMemoryMXBean` accessor method from the `java.lang.management.ManagementFactory` class and gives the MBean an object name `type=java,SubSystem=Memory`.

```
<mbeans>
  <mbean id="2">
    <mbean-factory>java.lang.management.ManagementFactory</mbean-factory>
    <mbean-accessor>getMemoryMXBean</mbean-accessor>
    <mbean-name>type=java,SubSystem=Memory</mbean-name>
    <enabled>>true</enabled>
```

```

    </mbean>
</mbeans>

```

As configured, the MBean can be managed remotely and includes the member where it is located as part of the object name. Therefore, the complete object name when the MBean is registered is:

```
Coherence:type=java,SubSystem=Memory,nodeId=<nodeId>
```

## Registering MBeans Using a Query

The `<mbean-query>` element is used to query an MBean server when registering and instantiating MBeans. The member's default MBean server is queried by default but an MBean server can also be specified. The following example executes the `java.lang:*` query on the member's default MBean server to find MBeans to register.

```

<mbeans>
  <mbean id="3">
    <mbean-query>java.lang:*/</mbean-query>
    <mbean-name>type=Platform</mbean-name>
    <enabled>>true</enabled>
  </mbean>
</mbeans>

```

To specify an MBean server other than the member's default MBean server, enter the name of a default domain for the MBean server using the `<mbean-server-domain>` element. For example:

```

<mbeans>
  <mbean id="3">
    <mbean-query>java.lang:*/</mbean-query>
    <mbean-server-domain>MyDomain</mbean-server-domain>
    <mbean-name>type=Platform</mbean-name>
    <enabled>>true</enabled>
  </mbean>
</mbeans>

```

## Specifying a Different MBean Configuration Override File

The `tangosol.coherence.mbeans` system property specifies an MBean configuration override file to be used instead of the default `custom-mbeans.xml` override file. For example:

```
-Dtangosol.coherence.mbeans=my-mbeans.xml
```

If the file is not located in the classpath, enter the full (or relative) path to the file in addition to the file name. The system property also supports the use of a URL when specifying the location of an MBean configuration override file.

## Restricting MBeans to the Local MBean Server

By default, custom MBeans that are registered with Coherence are visible by any cluster member that enables JMX management. To restrict the MBean from being visible to remote cluster members, set the `<local-only>` element to `true`. This registers the MBean only to the local MBean server. For example:

```

<mbeans>
  <mbean id="100">

```

```
<mbean-class>com.MyMBean</mbean-class>
<mbean-name>type=application</mbean-name>
<local-only>true</local-only>
<enabled>true</enabled>
</mbean>
</mbeans>
```

## Registering Custom MBeans Programmatically

The `com.tangosol.net.management.Registry` interface is used to register custom MBeans programmatically. The following example registers a standard MBean to Coherence using the `register` method. See *Oracle Coherence Java API Reference* for complete details of the `Registry` interface.

```
Registry registry = CacheFactory.ensureCluster().getManagement();
Custom bean = new Custom();
String sName = registry.ensureGlobalName("type=Custom");

registry.register(sName, bean);
```

### Using Static MBean Names

The example uses the `ensureGlobalName` method when adding a custom MBean to Coherence. The method is used to add the `nodeId=...` portion to the end of the MBean's `ObjectName`. The node ID is required to have unique names on the centralized MBean server. If an application requires static MBean names, use a MBean query to add MBeans from a local MBean server to the Coherence management system. The MBeans are created on the managed member with the static name and then the `,nodeId=...` portion is added to the name of the MBean when aggregating on the managing member.

To use static MBean names:

1. Register the MBeans on the local MBean server of the managed member using the `registerMBean` or `createMBean` methods before joining the cluster. See the `MBeanServer` interface JavaDocs for information on using these methods:

<http://java.sun.com/j2se/6/docs/api/javax/management/MBeanServer.html>

2. Use the `MBeanHelper.findMBeanServer()` method to obtain the same MBean server that Coherence is using.
3. Configure the `custom-mbeans.xml` file to query the newly registered MBeans. See "Registering MBeans Using a Query" on page 3-3.

---

**Note:** Refer to the Sun documentation to ensure an environment is set up properly to access the local MBean server.

<http://download.oracle.com/javase/6/docs/technotes/guides/management/agent.html>

---

---



---

## Using JMX Reporting

This chapter describes JMX reporting and provides instructions for enabling and configuring JMX reporting.

The following sections are included in this chapter:

- [Overview of JMX Reporting](#)
- [Enabling JMX Reporting On a Cluster Member](#)
- [Switching the Default Report Group Configuration File](#)
- [Changing the Report Timestamp Format](#)
- [Administering JMX Reporting Using the Reporter MBean](#)
- [Running JMX Reporting in a Distributed Configuration](#)

### Overview of JMX Reporting

JMX reports are used to view key management information over time. The reports are most often used to identify trends that are valuable for troubleshooting and planning. JMX reporting is disabled by default and must be explicitly enabled in an operational override file or by using system properties.

#### Viewing Reports

By default, ten reports are created each hour and refreshed every minute. The reports are written out to the directory where the cluster member was started. Each file is prefixed with the date and hour the report was executed in a *YYYYMMDDHH* format. This allows for easy location and purging of unwanted information. The files generated are described in [Table 4–1](#) and detailed in [Chapter 6, "Analyzing Reporter Content."](#)

**Table 4–1** *Default JMX Reports*

File Name	Description
<i>YYYYMMDDHH-cache-usage.txt</i>	Contains cache utilization (put, get, and so on) statistic for each cache. This report is only created if a cache exists on the cluster.
<i>YYYYMMDDHH-Management.txt</i>	Contains information about the Coherence management framework.
<i>YYYYMMDDHH-memory-status.txt</i>	Contains memory and garbage collection information about each member.

**Table 4–1 (Cont.) Default JMX Reports**

File Name	Description
YYYYMMDDHH-network-health-detail.txt	Contains the publisher success rates and receiver success rates for each member.
YYYYMMDDHH-network-health.txt	Contains the publisher success rates and receiver success rates for the entire grid.
YYYYMMDDHH-nodes.txt	Contains a list of members of the grid.
YYYYMMDDHH-report-proxy.txt	Contains utilization information about each proxy server in the grid.
YYYYMMDDHH-service.txt	Contains Request and Task information for each service.
YYYYMMDDHH-flashjournal.txt	Contains usage and performance information for flash-based storage.
YYYYMMDDHH-ramjournal.txt	Contains usage information for RAM-based storage.

### Understanding JMX Reporting Configuration

JMX reports are defined within a report configuration file. The predefined report configuration files are located in the `/reports` directory of the `coherence.jar` library. Each report configuration file results in a JMX report being created. For example, the `report-node.xml` configuration file is used to create the `YYYYMMDDHH-nodes.txt` report. The report configuration files use Coherence MBeans to gather the data that is used in the report. See [Appendix B, "Report File Configuration Reference,"](#) for details about the report file configuration elements.

Report group configuration files are used to control which reports are generated, how often the reports are refreshed, the directory where the reports are saved, and any parameters for a report. Three predefined report group configuration files are included out-of-box and are located in the `/reports` directory of the `coherence.jar` file:

- `report-group.xml` – configures a set of core coherence reports to be generated. This is the report group configuration file that is used by the default configuration.
- `report-web.xml` – configures a set of Coherence\*Web reports. For reports specific to Coherence\*Web, see *Oracle Coherence User's Guide for Oracle Coherence\*Web*.
- `report-all.xml` – configures all the out-of-box reports to be generated.

See [Appendix C, "Report Group Configuration Reference,"](#) for details about the report group configuration elements.

### Customizing Reports

The predefined reporting configuring files can be customized and new configuration files can be created as required. Any MBeans that are registered in the Coherence MBean server (including custom MBeans) can have their data displayed within a report. [Chapter 5, "Creating Custom Reports,"](#) for detailed instructions.

## Enabling JMX Reporting On a Cluster Member

JMX reporting must be enabled before JMX reports are created. JMX reporting can only be enabled on a cluster member where JMX management has also been enabled. See ["Configuring JMX Management"](#) on page 2-1 for details on enabling JMX

management on a cluster member. The default reporting configuration creates a set of reports each hour and refreshes the reports every minute. The reports are written out to the directory where the cluster member was started.

---

**Note:** Decide on a plan for archiving reports before enabling JMX reporting.

---

To enable JMX reporting, edit the operational override file and add an `<autostart>` element, within the `<reporter>` element, that is set to `true`. For example:

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

<coherence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.oracle.com/coherence/coherence-operational-config"
  xsi:schemaLocation="http://xmlns.oracle.com/coherence/
  coherence-operational-config coherence-operational-config.xsd">
  <management-config>
    <reporter>
      <autostart
        system-property="tangosol.coherence.management.report.autostart">true
      </autostart>
    </reporter>
  </management-config>
</coherence>
```

The `tangosol.coherence.management.report.autostart` system property is used to enable JMX reporting instead of using the operational override file. For example:

```
-Dtangosol.coherence.management.report.autostart=true
```

## Switching the Default Report Group Configuration File

Report group configuration files are used to define which reports get generated. Three report group configuration files are included out-of-box and are used to generate predefined reports: `report-group.xml` (default- generates core reports), `report-web.xml` (generates Coherence\*Web reports), and `report-all.xml` (generates all reports).

To change the report group configuration file, edit the operational override file and add a `<configuration>` element, within the `<reporter>` element, that is set to a report group configuration files. For example:

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

<coherence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.oracle.com/coherence/coherence-operational-config"
  xsi:schemaLocation="http://xmlns.oracle.com/coherence/
  coherence-operational-config coherence-operational-config.xsd">
  <management-config>
    <reporter>
      <configuration
        system-property="tangosol.coherence.management.report.configuration">
        reports/report-all.xml</configuration>
      </reporter>
    </management-config>
</coherence>
```

The `tangosol.coherence.management.report.configuration` system property is used to switch the default report group configuration file instead of using the operational override file. For example:

```
-Dtangosol.coherence.management.report.configuration=reports/report-all.xml
```

## Changing the Report Timestamp Format

Timestamps that are displayed within a report use the local time zone and a default time and date format (EEE MMM dd HH:mm:ss zzz yyyy). The time zone and time and date format can be changed as required. The time zone is entered using either a zone ID (for example, US/Eastern) or a custom ID (for example, GMT-05:00). The time and date format must follow the patterns specified by the `java.text.SimpleDateFormat` class.

To change the report timestamp format, edit the operational override file and add the `<timezone>` and `<timeformat>` elements, within the `<reporter>` element, that are set to the time zone and time and date format, respectively. For example:

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

<coherence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.oracle.com/coherence/coherence-operational-config"
  xsi:schemaLocation="http://xmlns.oracle.com/coherence/
coherence-operational-config coherence-operational-config.xsd">
  <management-config>
    <reporter>
      <timezone>
        system-property="tangosol.coherence.management.report.timezone">
          US/Eastern
        </timezone>
      <timeformat>
        system-property="tangosol.coherence.management.report.timeformat">
          MMM dd, yyyy hh:mm:ss a
        </timeformat>
      </reporter>
    </management-config>
  </coherence>
```

The `tangosol.coherence.management.report.timezone` and the `tangosol.coherence.management.report.timeformat` system properties are used to change the report timestamp format instead of using the operational override file. For example:

```
-Dtangosol.coherence.management.report.timezone=US/Eastern
-Dtangosol.coherence.management.report.timeformat=hh:mm:ss a
```

## Administering JMX Reporting Using the Reporter MBean

JMX reporting is administered using the `ReporterMBean` MBean. The MBean contains attributes and operations that are used to view JMX statistics and administer JMX reporting in real time. See "[Accessing Coherence MBeans](#)" on page 2-6 for instructions on how to access Coherence MBeans.

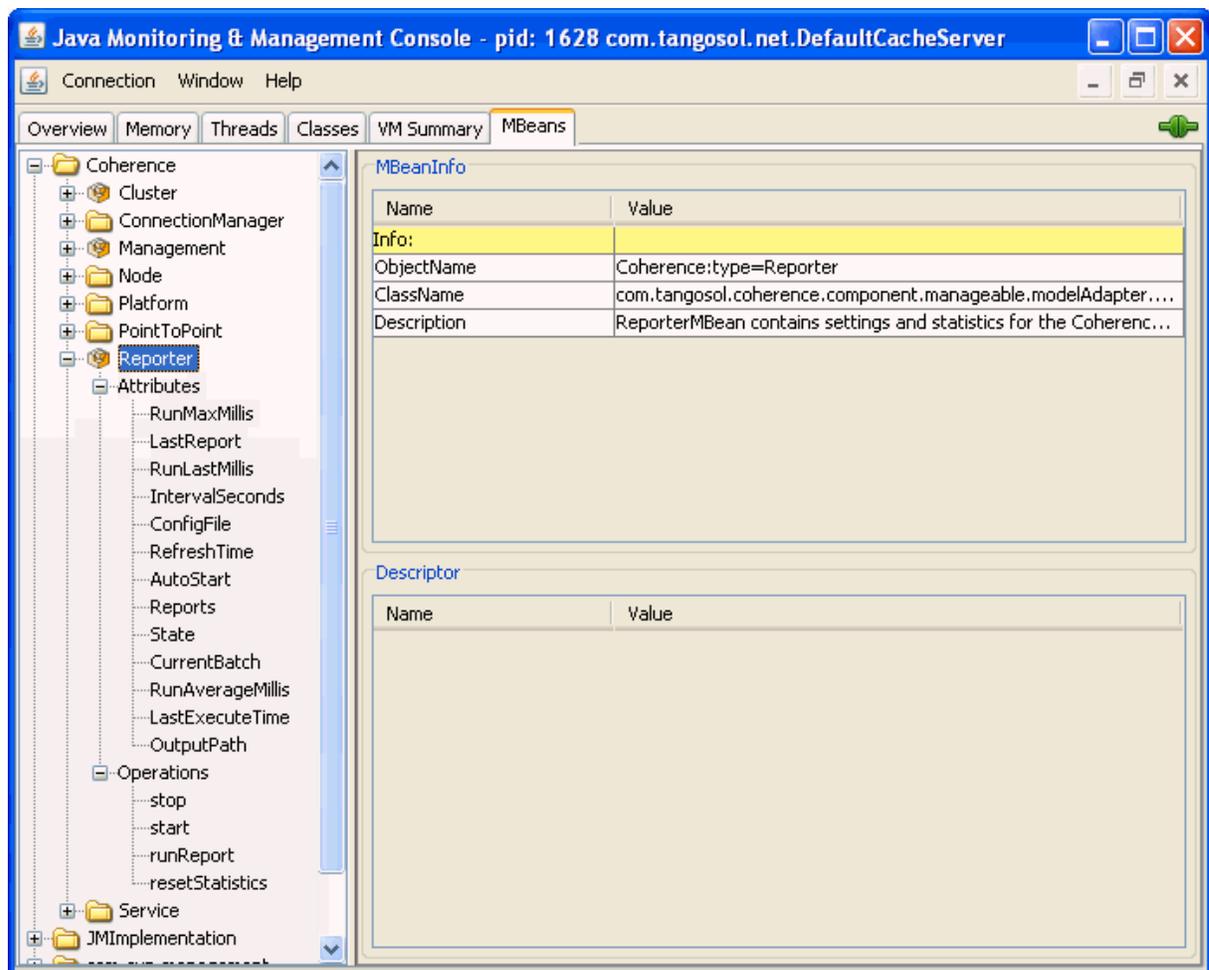
---

**Note:** Attributes that are changed using the ReporterMBean MBean are not saved and revert back to their original settings when the cluster member is restarted.

---

Figure 4–1 shows the ReporterMBean MBean’s attributes and operations within JConsole.

**Figure 4–1 ReporterMBean MBean in JConsole**



The following tasks can be performed using the ReporterMBean MBean. See ["ReporterMBean"](#) on page A-24 for a complete reference of the ReporterMBean MBean’s attributes and operations.

- Starting JMX Reporting – Use the `start` operation to start JMX reporting if it is currently stopped.
- Stopping JMX Reporting – Use the `stop` operation to stop JMX reporting. JMX reporting is restarted when the cluster member is restarted if the cluster member is configured to automatically start JMX reporting.
- Running JMX Reporting on Demand – Use the `runReport` operation to run either a single report or a report group one time. A report configuration file or report group configuration file must be specified. In addition, JMX reporting must be started to run the report(s).

- Changing the output path – Use the `OutputPath` attribute to change the location where reports are saved.
- Changing the refresh interval – Use the `IntervalSeconds` attribute to change how often the report data is refreshed.
- Changing the report configuration file – Use the `ConfigFile` attribute to change which report group configuration file to use.
- Changing the batch number – use the `CurrentBatch` attribute to change the current batch number. The next refresh of the reports is incremented from the new batch number.

## Running JMX Reporting in a Distributed Configuration

A distributed configuration is only recommended in situations where grid stability is an issue. In this configuration, the distributed reporters run independently and the execution times do not align. Therefore, grid level analysis is extremely difficult but member level analysis during periods when members may be leaving or joining the grid is still available.

When running in distributed mode, each member logs local JMX statistics while allowing for centralized management of the Reporters. To enable distributed configuration:

Start the managing member using the following system properties:

```
-Dtangosol.coherence.management.report.autostart=false  
-Dtangosol.coherence.management.report.distributed=true  
-Dtangosol.coherence.management=all  
-Dcom.sun.management.jmxremote
```

Start the cluster member to be managed using the following system properties:

```
-Dtangosol.coherence.management.report.autostart=true  
-Dtangosol.coherence.management.report.distributed=true  
-Dtangosol.coherence.management=local-only  
-Dtangosol.coherence.management.remote=true
```

---

---

## Creating Custom Reports

This chapter provides instructions for creating custom report definition files and creating custom report group configuration files.

The following sections are included in this chapter:

- [Overview of Creating Custom Reports](#)
- [Constructing a Report Configuration File](#)
- [Creating a Custom Report Group Configuration File](#)
- [Configuring Custom Reports to Generate](#)

### Overview of Creating Custom Reports

Custom reports can be created based on any MBeans residing in the Coherence MBean server domain (including custom MBeans). This allows different management data to be combined in meaningful ways that are specific to a cluster implementation. Custom reports are typically created during testing to correlate data and monitor trends when troubleshooting and planning for production.

Before creating custom reports, extract the predefined reporting configuration files that are located in the `/reports` directory of the `coherence.jar` library. Refer to the files when learning to create custom reports. The files can also be used as a starting point and modified as required.

#### Report Configuration files

JMX reporting utilizes two types of configuration files: a report configuration file and a report group configuration file.

- **Report Configuration File** – A report configuration file is used to construct a report at run time. It includes the name of the report, which MBean data should be extracted, and how the data is organized. Each report has a corresponding report configuration file. Report configuration files are XML documents that are defined by the `coherence-report-config.xsd` schema file which is packaged in the root directory of the `coherence.jar` library.
- **Report Group Configuration File** – A report group configuration file is used to configure which reports are generated at run time. It includes the name and location of each report configuration file, the output directory to which reports are saved, and how often the reports are refreshed. Report group configuration files can be created as required. Report group configuration files are XML documents that are defined by the `coherence-report-group-config.xsd` schema file which is packaged in the root directory of the `coherence.jar` library.

## Constructing a Report Configuration File

The topics in this section detail how to construct a report configuration file based on the `coherence-report-config.xsd` file. See [Appendix B, "Report File Configuration Reference,"](#) for details about the report file configuration elements.

The following topics are included in this section:

- [Specifying General Report Characteristics](#)
- [Specifying Data Columns](#)
- [Including Queries in a Report](#)
- [Using Filters to Construct Reports](#)
- [Using Functions to Construct a Report](#)
- [Using Aggregates to Construct a Report](#)
- [Constructing Delta Functions](#)

### Specifying General Report Characteristics

To correctly generate the report file, several elements must be configured. These elements are described in [Table 5–1](#).

**Table 5–1 Elements to Configure an Output File**

Element	Optional/ Required	Description
<file-name>	Required	The file name to create or update when the report is executed.
<delim>	Optional	The column delimiter for the report. Valid values are {tab}, {space} or a printable character. The default value is {tab}. If a string longer than one character is entered, the first character in the string is used.
<hide-headers>	Optional	A boolean element to determine if the headers are to be included in the report. If <code>true</code> , the column headers and the report description are not included in the file. The default value is <code>false</code> .

#### file-name Macros

There are pre-defined macros that you can use with the `file-name` element. These macros can add a member name, a batch number, or a date to the generated file name.

**Table 5–2 Macros that can be Used with the file-name Element**

Macro	Description
sequence	Adds a sequence number into the filename of the report.
date	Includes the date (with the format <code>YYYYMMDDHH</code> ), into the file name of the report. The date is used mostly when the data is only kept for a certain period and then discarded.
node	Includes the member ID into the file name string. The node setting is helpful when many members are executing the same report and the output files are integrated for analysis.

## file-name Macro Examples

The following example creates a file `200901012_network_status.txt` on January 1, 2009, during hour 12. The filename changes with the system time on the member executing the report.

```
<file-name>{{date}}_network_status.txt</file-name>
```

The following example creates a file `00012_network_status.txt` when the report is executed on member 12. Note that due to the volatile nature of the Node Id, long term storage in this manner is not recommended.

```
<file-name>{node}_network_status.txt</file-name>
```

The following example creates a file `0000000021_network_status.txt` on the 21st execution of the report. Note that due to the volatile nature of the batch, long term storage in this manner is not recommended.

```
<file-name>{sequence}_network_status.txt</file-name>
```

## Specifying Data Columns

Data columns can be sourced from JMX Attributes, `ObjectName` key part, JMX composite attributes, JMX joined attributes, Report macros, and Report Constants.

### How to Include an Attribute

To include data from MBeans returned from the query-pattern, the report must have a column with an attribute source. This is the most common item that is included in the report.

[Example 5-1](#) illustrates how to include the `RoleName` attribute from the query pattern `Coherence:type=Node, *`.

#### **Example 5-1 Including an Attribute Obtained from a Query Pattern**

```
<column id = "RoleName">
  <type>attribute</type>
  <name>RoleName</name>
  <header>Role Name</header>
</column>
```

### How to Include Part of the Key

A value that is present in an `ObjectName` key can be obtained from the `ObjectNames` returned from the query-pattern. This value can subsequently be included in the report.

[Example 5-2](#) illustrates how to include the `nodeId` key part from the query pattern `Coherence:type=Node, *`.

#### **Example 5-2 Including Part of an ObjectName Key in a Report**

```
<column id = "NodeId">
  <type>key</type>
  <name>nodeId</name>
  <header>Node Id</header>
</column>
```

## How to Include Information from Composite Attributes

JMX composite values are used to include part of a composite data attribute in a report.

[Example 5–9](#) illustrates how to include the `startTime` of the `LastGCInfo` attribute from the query pattern `java.lang:type=GarbageCollector,*`.

### **Example 5–3 Including Information from a Composite Attribute in a Report**

```
<column id="LastGCStart">
  <type>attribute</type>
  <name>LastGcInfo/startTime</name>
  <header>Last GC Start Time</header>
</column>
```

## How to Include Information from Multiple MBeans

A JMX join attribute is required when a report requires information from multiple MBeans. The major considerations when creating a join is to determine both the primary query, the join query and the foreign key. The primary query should be the query that returns the appropriate number of rows for the report. The join query pattern must reference a single MBean and cannot contain a wildcard (\*). The foreign key is determined by what attributes from the primary query that are required to complete the join query string.

The reporter feature that enables joins between MBeans is a column substitution macro. The column substitution allows for the resulting value from a column to be included as part of a string. A column substitution macro is a column ID attribute surrounded by curly braces "{}". The reporter does not check for cyclical references and fails during execution if a cycle is configured.

### **Including Multiple MBean Information Example**

You can draw information from multiple MBeans and include it in a report. This requires a `join` between the MBeans.

---



---

**Note:** The major limitation of `join` attributes is that the result of the join must have only one value.

---



---

For example, if a report requires the `TotalGets` from the `Cache` MBean (`Coherence:type=cache,*`) and `RoleName` from the `Node` MBean (`Coherence:type=Node,*`), then a `join` attribute must be used.

Since a greater number of MBeans come from the `Cache` MBean, `Coherence:type=Cache,*` is the primary query and the `RoleName` is the `join` attribute. The foreign key for this join is the `nodeId` key part from the `Cache` MBean and it must be included in the report. The configuration for this scenario is illustrated in [Example 5–4](#).

### **Example 5–4 Including Information from Multiple MBeans in a Report**

```
<column id="RoleName">
  <type>attribute</type>
  <name>RoleName</name>
  <header>Role Name</header>
  <query>
    <pattern>Coherence:type=Node,nodeId={NodeFK}</pattern>
```

```

    </query>
</column>

<column id = "NodeFK">
  <type>key</type>
  <name>nodeId</name>
  <header>Node Id</header>
</column>

```

## How to Use Report Macros

There are three report macros that can be included in a report:

- Report Time (`report-time`)—is the time and date that the report was executed. This information is useful for time series analysis.
- Report Batch/Count (`report-count`)—is a long identifier that is used to correlate information from different reports executed at the same time.
- Reporting Member (`report-node`)—is used when integrating information from the same report executed on different members or excluding the executing member information from the report.

To include the execution time into the report:

### **Example 5–5 Including Execution Time in a Report**

```

<column id = "ReportTime">
  <type>global</type>
  <name>{report-time}</name>
  <header>Report Time</header>
</column>

```

To include the Report Batch/Count:

### **Example 5–6 Including the Report Batch/Count in a Report**

```

<column id="ReportBatch">
  <type>global</type>
  <name>{report-count}</name>
  <header>batch</header>
</column>

```

To include the execution member:

### **Example 5–7 Including the Execution Member**

```

<column id="ReportNode">
  <type>global</type>
  <name>{report-node}</name>
  <header>ExecNode</header>
  <hidden>true</hidden>
</column>

```

## How to Include Constant Values

Report constants are used for either static values or report parameters. These constants can be either double or string values. Often, these are used in filters to limit the results to a particular data set or in calculations.

[Example 5–8](#) illustrates how to include a constant double of 1.0 in a report:

**Example 5-8 Including a Constant Numeric Value in a Report**

```

<column id ="One">
  <type>constant</type>
  <header>Constant1</header>
  <hidden>true</hidden>
  <data-type>double</data-type>
  <value>1.0</value>
</column>

```

[Example 5-9](#) illustrates how to include the constant string `dist-Employee` in a report:

**Example 5-9 Including a Constant String in a Report**

```

<column id ="EmployeeCacheName">
  <type>constant</type>
  <header>Employee Cache Name</header>
  <hidden>true</hidden>
  <data-type>string</data-type>
  <value>dist-Employee</value>
</column>

```

## Including Queries in a Report

The query is the foundation of the information included in a report. Each query includes a query pattern, column references, and an optional filter reference. The query pattern is a string that is a JMX `ObjectName` query string. This string can return one or more MBeans. The column references must be defined in the `<columns>` section of the report definition file. The filter reference must be defined in the `<filters>` section of the report section.

[Example 5-10](#) illustrates how to include the list all the Node IDs and RoleNames in the cluster where the RoleName equals `CoherenceServer`.

**Example 5-10 Including a List of the Cluster's NodeIDs and RoleNames in a Report**

```

<filters>
  <filter id="equalsRef">
    <type>equals</type>
    <params>
      <column-ref>RoleRef</column-ref>
      <column-ref>StringRef</column-ref>
    </params>
  </filter>
</filters>

<query>
  <pattern>Coherence:type=Node,*</pattern>
  <filter-ref>equalsRef</filter-ref>
</query>

<row>
  <column id ="NodeRef">
    <type>key</type>
    <name>nodeId</name>
    <header>Node Id</header>
  </column>

  <column id ="RoleRef">
    <name>RoleName</name>

```

```

    <header>Role</header>
  </column>

  <column id = "StringRef">
    <type>constant</type>
    <name>ConstString</name>
    <hidden>true</hidden>
    <data-type>string</data-type>
    <value>CoherenceServer</value>
  </column>

</row>

```

## Using Filters to Construct Reports

Filters limit the data returned in the Report. Filters are either comparison filters or composite filters. Comparison Filters evaluate the results of two columns while composite filters evaluate the boolean results from one or two filters. Comparison filters are `equals`, `greater`, and `less`.

Composite Filter types are `and`, `or`, and `not`. Each composite filter evaluates the filter parameters first to last and apply standard boolean logic. Composite filter evaluation uses standard short circuit logic. Cyclic references checks are not performed during execution. If a cyclic reference occurs, it creates a run-time error.

[Example 5–11](#) illustrates how to define an `equals` filter where `RoleRef` and `StringRef` are defined columns.

### **Example 5–11 Using an Equals Filter for a Report**

```

<filters>
  <filter id="equals">
    <type>equals</type>
    <params>
      <column-ref>RoleRef</column-ref>
      <column-ref>StringRef</column-ref>
    </params>
  </filter>
</filters>

```

[Example 5–12](#) illustrates how to define a filter where the number of `PacketsResent` are greater than `PacketsSent` (assuming `PacketsResent` and `PacketsSent` are valid column references).

### **Example 5–12 Defining a "Greater Than" Filter for a Report**

```

<filters>
  <filter id="greaterRef">
    <type>greater</type>
    <params>
      <column-ref>PacketsResent</column-ref>
      <column-ref>PacketsSent</column-ref>
    </params>
  </filter>
</filters>

```

[Example 5–13](#) illustrates how to define an filter where the number of `PacketsResent` are less than `PacketsSent` (assuming `PacketsResent` and `PacketsSent` are valid column references).

**Example 5–13 Defining a "Less Than" Filter for a Report**

```

<filters>
  <filter id="greaterRef">
    <type>less</type>
    <params>
      <column-ref>PacketsResent</column-ref>
      <column-ref>PacketsSent</column-ref>
    </params>
  </filter>
</filters>

```

[Example 5–14](#) illustrates how to define an and filter (assuming all `column-ref` values are valid).

**Example 5–14 Defining an "And" Filter for a Report**

```

<filters>
  <filter id="equalsRef">
    <type>equals</type>
    <params>
      <column-ref>RoleRef</column-ref>
      <column-ref>StringRef</column-ref>
    </params>
  </filter>

  <filter id="greaterRef">
    <type>greater</type>
    <params>
      <column-ref>PacketsResent</column-ref>
      <column-ref>PacketsSent</column-ref>
    </params>
  </filter>

  <filter>
    <type>and</type>
    <params>
      <filter-ref>greaterRef</filter-ref>
      <filter-ref>equalsRef</filter-ref>
    </params>
  </filter>
</filters>

```

[Example 5–15](#) illustrates how to define an or filter (assuming all `column-ref` values are valid).

**Example 5–15 Defining an "Or" Filter for a Report**

```

<filters>
  <filter id="equalsRef">
    <type>equals</type>
    <params>
      <column-ref>RoleRef</column-ref>
      <column-ref>StringRef</column-ref>
    </params>
  </filter>

  <filter id="greaterRef">
    <type>greater</type>
    <params>
      <column-ref>PacketsResent</column-ref>

```

```

        <column-ref>PacketsSent</column-ref>
    </params>
</filter>

<filter>
    <type>or</type>
    <params>
        <filter-ref>greaterRef</filter-ref>
        <filter-ref>equalsRef</filter-ref>
    </params>
</filter>
</filters>

```

**Example 5–16** illustrates how to define a `not equals` filter, where `RoleRef` and `StringRef` are defined columns.

**Example 5–16 Defining a "Not Equals" Filter for a Report**

```

<filters>
    <filter id="equals">
        <type>equals</type>
        <params>
            <column-ref>RoleRef</column-ref>
            <column-ref>StringRef</column-ref>
        </params>
    </filter>

    <filter id = "Not">
        <type>not</type>
        <params>
            <filter-ref>equals</filter-ref>
        </params>
    </filter>
</filters>

```

## Using Functions to Construct a Report

Reporter functions allow mathematical calculations to be performed on data elements within the same row of the report. The supported functions are `Add`, `Subtract`, `Multiply`, and `Divide`. Function columns can then be included as parameters into other function columns.

### Function Examples

**Example 5–17** illustrates how to add columns (`Attribute1` and `Attribute2`) and place the results into a third column (`Addition`).

**Example 5–17 Adding Column Values and Including Results in a Different Column**

```

<column id="AttributeID1">
    <name>Attribute1</name>
</column>

<column id="AttributeID2">
    <name>Attribute2</name>
</column>

<column id="Addition">
    <type>function</type>

```

```
<name>Add2Columns</name>
<header>Adding Columns</header>
<function-name>add</function-name>
<params>
  <column-ref>AttributeID1</column-ref>
  <column-ref>AttributeID2</column-ref>
</params>
</column>
```

[Example 5–18](#) illustrates how to subtract one column value (`Attribute2`) from another (`Attribute1`) and place the results into a third column (`Subtraction`).

**Example 5–18 Subtracting Column Values and Including Results in a Different Column**

```
<column id="AttributeID1">
  <name>Attribute1</name>
</column>

<column id="AttributeID2">
  <name>Attribute2</name>
</column>

<column id="Subtraction">
  <type>function</type>
  <name>Subtract2Columns</name>
  <header>Difference</header>
  <function-name>subtract</function-name>
  <params>

    <column-ref>AttributeID1</column-ref>
    <column-ref>AttributeID2</column-ref>
  </params>
</column>
```

[Example 5–19](#) illustrates how to multiply column values (`Attribute1` and `Attribute2`) place the results into a third column (`Multiplication`).

**Example 5–19 Multiplying Column Values and Including Results in a Different Column**

```
<column id="AttributeID1">
  <name>Attribute1</name>
</column>

<column id="AttributeID2">
  <name>Attribute2</name>
</column>

<column id="Multiplication">
  <type>function</type>
  <name>Multiply2Columns</name>
  <header>Multiply Columns</header>
  <function-name>multiply</function-name>
  <params>
    <column-ref>AttributeID1</column-ref>
    <column-ref>AttributeID2</column-ref>
  </params>
</column>
```

[Example 5–20](#) illustrates how to divide one column (`Attribute1`) by another (`Attribute2`) into a third column (`Division`). The result of all division is a `Double` data type.

**Example 5–20 Dividing Column Values and Including Results in a Different Column**

```
<column id="AttributeID1">
  <name>Attribute1</name>
</column>

<column id="AttributeID2">
  <name>Attribute2</name>
</column>

<column id="Division">
  <type>function</type>
  <name>Dividing2Columns</name>
  <header>Division</header>
  <function-name>divide</function-name>
  <params>
    <column-ref>AttributeID1</column-ref>
    <column-ref>AttributeID2</column-ref>
  </params>
</column>
```

## Using Aggregates to Construct a Report

Reporter aggregates allow for multiple rows to be aggregated into a single value or row. [Table 5–3](#) describes the available aggregate types.

**Table 5–3 Reporter Aggregate Types**

Type	Description
avg	Calculate the mean value for all values in the column.
max	Return the maximum value for all values in the column.
min	Return the minimum value for all values in the column.
sum	Add all the values from a column.

### Aggregate Examples

Sum the values in the `size` column

**Example 5–21 Adding the Values in a Column**

```
<column id="SumRef">
  <type>function</type>
  <column-ref>size</column-ref>
  <function-name>sum</function-name>
  <header>Sum</header>
</column>
```

Average the values in the `size` column

**Example 5–22 Calculating the Average of Values in a Column**

```
<column id="AverageRef">
  <type>function</type>
```

```
<header>Average</header>
<column-ref>size</column-ref>
<function-name>avg</function-name>
</column>
```

Find the maximum the value in the `size` column

**Example 5–23 Finding the Maximum Value in a Column**

```
<column id="MaximumRef">
  <type>function</type>
  <header>Maximum</header>
  <column-ref>size</column-ref>
  <function-name>max</function-name>
</column>
```

Find the minimum the value in the `size` column

**Example 5–24 Finding the Minimum Value in a Column**

```
<column id="MinimumRef">
  <type>function</type>
  <header>Minimum</header>
  <column-ref>size</column-ref>
  <function-name>min</function-name>
</column>
```

## Constructing Delta Functions

Many numeric attributes in the Coherence report are cumulative. These values are reset only when the `resetStatistics` operation is executed on the MBean. To determine the state of the system without resetting the statistics, the Reporter uses a delta function. The delta function subtracts the prior value of a column from the current value of a column and returns the difference.

The prior values for a report are stored in a map on the Reporter client. This map is keyed by the "delta key". By default, the delta key is the MBean name for the attribute. However, when one-to-one relationship does not exist between the MBean and the rows in the report, or the MBean name is subject to change between executions of the report, the delta key is calculated using the columns provided in the `<params>` section.

---

---

**Note:** Accuracy of Delta Functions: delta functions are only correct when the report is running as part of a report batch.

---

---

### Delta Function Examples

[Example 5–25](#) illustrates how to include a delta calculation of an attribute. (Assume `PacketsSent` is a defined column)

**Example 5–25 Delta Calculation for an Attribute**

```
<column id="DeltaPacketsSent">
  <type>function</type>
  <name>PacketsSent</name>
  <header>Delta Sent</header>
  <column-ref>PacketsSent</column-ref>
  <function-name>delta</function-name>
```

```
</column>
```

**Example 5–26** illustrates how to include a delta calculation of an attribute with an alternate delta key. (Assume `PacketsSent`, `NodeID` and `TimeStamp` are defined columns)

**Example 5–26 Delta Calculation for an Attribute with an Alternate Delta Key**

```
<column id="DeltaPacketsSent">
  <type>function</type>
  <name>PacketsSent</name>
  <header>Delta Sent</header>
  <column-ref>PacketsSent</column-ref>
  <function-name>delta</function-name>
  <params>
    <column-ref>NodeID</column-ref>
    <column-ref>TimeStamp</column-ref>
  </params>
</column>
```

## Creating a Custom Report Group Configuration File

The topics in this section detail how to create a report group configuration file based on the `coherence-report-group-config.xsd` file. This configuration file is used at run time to determine what reports get generated, how often the reports get refreshed, and where the reports are saved. If a single report can be used with different parameters, these parameters are also configured. See [Appendix C, "Report Group Configuration Reference,"](#) for details about the report group configuration elements.

The following topics are included in this section:

- [Specifying the Report Refresh Frequency](#)
- [Specifying the Output Directory for Reports](#)
- [Specifying the Report List](#)

### Specifying the Report Refresh Frequency

The `<frequency>` element specifies how often the reports are refreshed. Selecting an appropriate frequency is critical. If the frequency is too short, the reporter can generate a large amount of data and consume significant disk space. If the frequency is too long, not enough information is generated. The value is entered in either seconds(s) or minutes(m). For example:

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

<report-group xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.oracle.com/coherence/coherence-report-group-config"
  xsi:schemaLocation="http://xmlns.oracle.com/coherence/
  coherence-report-group-config coherence-report-group-config.xsd">
  <frequency>5m</frequency>
  ...
```

### Specifying the Output Directory for Reports

The `<output-directory>` element specifies the directory path to which reports are saved. The directory path is prepended to the output file name that is defined in the

report configuration file. See ["Specifying General Report Characteristics"](#) on page 5-2. The username which the member is executing must have read/write access to this path. The path can be absolute or relative to directory where the cluster member was started (. /). The following example saves the reports to the /output directory.

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

<report-group xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.oracle.com/coherence/coherence-report-group-config"
  xsi:schemaLocation="http://xmlns.oracle.com/coherence/
coherence-report-group-config coherence-report-group-config.xsd">
  <frequency>5m</frequency>
  <output-directory>/output</output-directory>
  ...
```

## Specifying the Report List

The <report-list> element specifies the name and location of any number of report configuration files. The path can be either a file or a URL. To enter a report configuration file, add a <location> element within a <report-config> element. For example:

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

<report-group xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.oracle.com/coherence/coherence-report-group-config"
  xsi:schemaLocation="http://xmlns.oracle.com/coherence/
coherence-report-group-config coherence-report-group-config.xsd">
  <frequency>5m</frequency>
  <output-directory>/output</output-directory>
  <report-list>
    <report-config>
      <location>/config/myReport.xml</location>
    </report-config>
    <report-config>
      <location>config/aSecondReport.xml</location>
    </report-config>
  </report-list>
</report-group>
```

## Configuring Custom Reports to Generate

To configure custom reports to generate, JMX reporting must be enabled and the reporter must be configured to use a custom report group configuration file. See ["Enabling JMX Reporting On a Cluster Member"](#) on page 4-2 for details on enabling JMX reporting.

To configure the reporter to use a custom report group configuration file, edit the operational override file and add a <configuration> element, within the <reporter> element, that is set to the location of a custom report group configuration file. The following example enables JMX reporting and sets a custom report group configuration file:

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

<coherence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.oracle.com/coherence/coherence-operational-config"
  xsi:schemaLocation="http://xmlns.oracle.com/coherence/
coherence-operational-config coherence-operational-config.xsd">
```

```
<management-config>
  <reporter>
    <configuration
      system-property="tangosol.coherence.management.report.configuration">
      my-report-group.xml</configuration>
    <autostart
      system-property="tangosol.coherence.management.report.autostart">true
    </autostart>
  </reporter>
</management-config>
</coherence>
```

The `tangosol.coherence.management.report.autostart` and `tangosol.coherence.management.report.configuration` system property are used to enable JMX reporting and configure a report group configuration file instead of using the operational override file. For example:

```
-Dtangosol.coherence.management.report.autostart=true
-Dtangosol.coherence.management.report.configuration=my-report-group.xml
```



---



---

## Analyzing Reporter Content

Coherence provides out-of-box reports that helps administrators and developers better analyze usage and configuration issues that may occur. For reports specific to Coherence\*Web, see "Running Performance Reports" in *Oracle Coherence User's Guide for Oracle Coherence\*Web*.

The following sections are included in this chapter:

- [Understanding the Cache Size Report](#)
- [Understanding the Cache Usage Report](#)
- [Understanding the Flash Journal Report](#)
- [Understanding the Management Report](#)
- [Understanding the Memory Status Report](#)
- [Understanding the Network Health Detail Report](#)
- [Understanding the Network Health Report](#)
- [Understanding the Node List Report](#)
- [Understanding the Proxy Report](#)
- [Understanding the Ram Journal Report](#)
- [Understanding the Service Report](#)

### Understanding the Cache Size Report

The cache size report can be executed either on demand or it can be added as part of the report batch and the Caches should have the `<unit-calculator>` subelement of `<local-scheme>` set to `BINARY`. The cache size file is a tab delimited file that is prefixed with the date and hour in `YYYYMMDDHH` format and post fixed with `-cache-size.txt`. For example `2009013101-cache-size.txt` would be created on January 31, 2009 at 1:00 AM. [Table 6-1](#) describes the content of the Cache Size report.

**Table 6-1** Contents of the Cache Size Report

Column	Data Type	Description
Batch Counter	Long	A sequential counter to help integrate information between related files. This value does reset when the reporter restarts and is not consistent across members. However, it is helpful when trying to integrate files.
Cache Name	String	The name of the cache.

**Table 6–1 (Cont.) Contents of the Cache Size Report**

Column	Data Type	Description
Cache Size	Double	The number of objects in the cache.
Memory Bytes	Double	The number of bytes consumed by the objects in the cache. This does not include indexes or over head.
MemoryMB	Double	The MB consumed by the objects in the cache. This does not include indexes or over head.
Avg Object Size	Double	The Average memory consumed by each object.

## Understanding the Cache Usage Report

The cache usage report provides information about cache usage (gets, puts, evictions, and so on). The report is a tab-delimited file that is prefixed with the date and hour in YYYYMMDDHH format and post fixed with `-cache-usage.txt`. For example, `2010013113-cache-usage.txt` would be created on January 31, 2010 1:00 pm. [Table 6–2](#) describes the content of the cache utilization report.

**Table 6–2 Contents of the Cache Usage Report**

Column	Data Type	Description
Batch Counter	Long	A sequential counter to help integrate information between related files. This value resets when the Reporter restarts, and is not consistent across members. However, it is helpful when trying to integrate files.
Report Time	Date	The system time when the report is executed.
Service	String	The name of the cache service.
Cache Name	String	The name of the cache.
Tier	String	Value can be either <code>front</code> or <code>back</code> . Describes whether the cache resides in the front-tier (local cache) or back tier (remote cache).
Total Puts	Double	The total number of puts for the cache across the cluster since the last time the report was executed.
Total Puts Milliseconds	Double	The total number of milliseconds spent per <code>put()</code> invocation ( <code>PutsMillis</code> ) across the cluster since the last time the report was executed.
Total Gets	Double	The total number of gets for the cache across the cluster since the last time the report was executed.
Total Gets Milliseconds	Double	The total number of milliseconds spent per <code>get()</code> invocation ( <code>GetsMillis</code> ) across the cluster since the last time the report was executed.
Total Hits	Double	The total number of hits for the cache across the cluster since the last time the report was executed.
Total Hits Milliseconds	Double	The total number of milliseconds spent per <code>get()</code> invocation that is a hit ( <code>HitsMillis</code> ) across the cluster since the last time the report was executed.
Total Misses	Double	The total number of misses for the cache across the cluster since the last time the report was executed.
Total Misses Milliseconds	Double	The total number of milliseconds spent per <code>get()</code> invocation that is a miss ( <code>MissesMillis</code> ) across the cluster since the last time the report was executed.

**Table 6–2 (Cont.) Contents of the Cache Usage Report**

Column	Data Type	Description
Total Writes	Double	The total number of storage writes for the cache across the cluster since the last time the report was executed.
Total Writes Milliseconds	Double	The total number of milliseconds spent in storage write operations (WritesMillis) across the cluster since the last time the report was executed.
Total Reads	Double	The total number of reads from a cache store for the cache across the cluster since the last time the report executed.
Total Read Milliseconds	Double	The total number milliseconds on cache store reads for the cache across the cluster since the last time the report executed.
Total Failures	Long	The total number of storage failures for the cache across the cluster since the last time the report was executed.
Total Queue evictions	Long	The sum of the queue link sizes across the cluster.
Cache Prunes	Long	The total number of evictions for the cache across the cluster since the last time the report was executed.
Cache Prunes Milliseconds	Long	The total number of prunes for the cache across the cluster since the last time the report was executed.
Cache Prunes Milliseconds	Long	The total number of milliseconds spent in the prune operation (PrunesMillis) across the cluster since the last time the report was executed.

## Understanding the Flash Journal Report

The flash journal report displays statistics that are used to determine how well data is being stored to flash memory. The report is a tab-delimited file that is prefixed with the date and hour in YYYYMMDDHH format and post fixed with `-flashjournal.txt`. For example, `2010013113-flashjournal.txt` would be created on January 31, 2010 1:00 pm. [Table 6–3](#) describes the content of the cache utilization report.

**Table 6–3 Contents of the Flash Journal Report**

Column	Type	Description
Batch Counter	Long	A sequential counter to help integrate information between related files. This value does reset when the reporter restarts and is not consistent across members. However, it is helpful when trying to integrate files.
Report Time	Date	The system time when the report executed.
Node ID	Long	The member for the flash journal statistics.
FileCount	Integer	The number of journal files that are currently in use.
BinaryStoreCount	Integer	The number of active <code>JournalBinaryStore</code> objects that are using this journal.
TotalDataSize	Long	The amount of data, in bytes, that are currently stored for this journal.
TotalFileSize	Long	The total size of all journal files for this journal.
BacklogCount	Integer	The number of serialized values that have yet to be persisted.

**Table 6–3 (Cont.) Contents of the Flash Journal Report**

Column	Type	Description
BacklogSize	Integer	The maximum size, in bytes, of the backlog. The backlog is the amount of serialized values that has yet to be persisted. Client threads are blocked if this limit is exceeded and remain blocked until the backlog recedes below this limit
PoolSize	Integer	The total size, in bytes, of all available buffers in the pool.

## Understanding the Management Report

The Management report contains refresh statistics that are used to determine if the management framework is providing a timely view of management data for all MBeans. The management file is a tab delimited file that is prefixed with the date and hour in YYYYMMDDHH format and post fixed with `-Management.txt`. For example `2009013113-Management.txt` would be created on January 31, 2009 at 1:00 PM. [Table 6–4](#) describes the content of the Management report.

**Table 6–4 Contents of the Management Report**

Column	Type	Description
Batch Counter	Long	A sequential counter to help integrate information between related files. This value does reset when the reporter restarts and is not consistent across members. However, it is helpful when trying to integrate files.
Report Time	Date	The system time when the report executed.
Refresh Policy	String	The refresh policy that is currently set, which is used to determine how data for remote models are being refreshed.
Refresh Time	Date	The timestamp when this model was last retrieved from a corresponding member. For local servers it is the local time.
Refresh Excess Count	Long	The number of times the MBean server predictively refreshed information and the information was not accessed.
Refresh Count	Long	The total number of snapshots retrieved since the statistics were last reset.
Refresh Prediction Count	Long	The number of times the MBean server used a predictive algorithm to refresh MBean information.
Refresh Timeout Count	Long	The number of times this management member has timed out while attempting to refresh remote MBean attributes.

## Understanding the Memory Status Report

The Memory Status report must be run as part of a report batch. The values are helpful in understanding memory consumption on each member and across the grid. For data to be included, cluster members must be configured to publish platform MBean information. The memory status file is a tab delimited file that is prefixed with the date and hour in YYYYMMDDHH format and post fixed with `-memory-status.txt`. For example `2009013115-memory-status.txt` would be created on January 31, 2009 at 3:00 PM. [Table 6–5](#) describes the content of the Memory Status report.

**Table 6–5 Contents of the Memory Status Report**

Column	Data Type	Description
Batch Counter	Long	A sequential counter to help integrate information between related files. This value does reset when the reporter restarts and is not consistent across members. However, it is helpful when trying to integrate files.
Report Time	Date	The system time when the report executed.
JVM Uptime	Long	The amount of time since the JVM was started.
Node Id	Long	The member for the memory statistics.
Gc Name	String	The name of the Garbage Collector information.
CollectionCount	Long	The number of garbage collections that have happened since the computer started.
Delta Collection Count	Long	The number of garbage collections that have occurred since the last execution of the report.
CollectTime	Long	The number of milliseconds the JVM has spent on garbage collection since the start of the JVM.
Delta Collect Time	Long	The number of milliseconds the JVM has spent on garbage collection since the last execution of the report.
Last GC Start Time	Long	The start time of the last garbage collection.
Last GC Duration Millis	Long	The total amount of time of the last garbage collection.
Last GC Stop Time	Long	The stop time of the last garbage collection.
Heap Committed	Long	The number of heap bytes committed at the time of report.
Heap Init	Long	The number of heap bytes initialized at the time of the report.
Heap Max	Long	The Maximum number of bytes used by the JVM since the start of the JVM.
Heap Used	Long	The bytes used by the JVM at the time of the report.

## Understanding the Network Health Detail Report

The Network Health report supporting member-level details for determining the health of the network communications. The network health detail file is a tab delimited file that is prefixed with the date and hour in YYYYMMDDHH format and post fixed with `-network-health-detail.txt`. For example `2009013114-network-health.txt` would be created on January 31, 2009 at 2:00PM. [Table 6–6](#) describes the content of the Network Health Detail report.

**Table 6–6 Contents of the Network Health Detail Report**

Column	Data Type	Description
Batch Counter	Long	A sequential counter to help integrate information between related files. This value does reset when the reporter restarts and is not consistent across members. However, it is helpful when trying to integrate files.
Report Time	Date	The system time when the report executed.

**Table 6–6 (Cont.) Contents of the Network Health Detail Report**

Column	Data Type	Description
RefreshTime	Date	The system time when management information was last retrieved from a corresponding node. Local servers display the local time.
Node Id	Long	The member for the network statistics.
Tx Success	Double	The publisher success rate for the member. If this value is within 2%-3% of the "Min Node Tx Success" and more than 10% less than the "Grid Tx Success" for the batch in the Network Health File, the corresponding member may be having difficulty communicating with the cluster. Constrained CPU, constrained network bandwidth or high network latency could cause this to occur.
RX Success	Double	The receiver success rate for the member. If this value is within 2%-3% of the "Min Node Rx Success" and more than 10% less than the "Grid Tx Success" for the batch in the Network Health File, the corresponding member may be having difficulty communicating with the cluster. Constrained CPU, constrained network bandwidth or high network latency could cause this to occur.
Packets Sent	Double	The total number of network packets sent by the member.
Current Packets Sent	Long	The number of packets sent by the member since the prior execution of the report.
Packets Resent	Long	The total number of network packets resent by the member. Packets are resent when the receiver of the packet receives and invalid packet or when an acknowledge packet is not sent within the appropriate amount of time.
Current Packet Resent	Long	The number of network packets resent by the member since the prior execution of the report.
PacketsRepeated	Long	The total number of packets received more than once.
Current Packets Repeated	Long	The number of packets received more than once since the last execution of the report.
Packets Received	Long	The total number of packets received by the member.
Current Packets Received	Long	The total number of packets received by the member since the last execution of the report.

## Understanding the Network Health Report

The Network Health report contains the primary aggregates for determining the health of the network communications. The network health file is a tab delimited file that is prefixed with the date and hour in YYYYMMDDHH format and post fixed with `-network-health.txt`. For example `2009013113-network-health.txt` would be created on January 31, 2009 at 1:00 PM. [Table 6–7](#) describes the content of the Network Health report.

**Table 6–7 Contents of the Network Health Report**

Column	Type	Description
Batch Counter	Long	A sequential counter to help integrate information between related files. This value does reset when the reporter restarts and is not consistent across members. However, it is helpful when trying to integrate files.
Report Time	Date	The system time when the report executed.
Min Node Rx Success	Double	The minimum receiver success rate for a member in the cluster. If this value is considerably less (10%) than the Grid Rx Success rate. Further analysis using the Network Health Detail should be done.
Grid Rx Success	Double	The receiver success rate for the grid as a whole. If this value is below 90%. Further analysis of the network health detail should be done.
Min Node Tx Success	Double	The minimum publisher success rate for a member in the cluster. If this value is considerably less (10%) than the Grid Rx Success rate. Further analysis using the Network Health Detail should be done.
Grid TX Success	Double	The publisher success rate for the grid as a whole. If this value is below 90%. Further analysis of the network health detail should be done.

## Understanding the Node List Report

Due to the transient nature of the node identifier (`nodeId`), the reporter logs out a list of members and user-defined member identity information. See "member-identity" in the *Oracle Coherence Developer's Guide*. The node list file is a tab delimited file that is prefixed with the date and hour in YYYYMMDDHH format and post fixed with `-nodes.txt`. For example `2009013101-nodes.txt` would be created on January 31, 2009 at 1:00 AM. [Table 6–8](#) describes the content of the Node List report.

**Table 6–8 Contents of the Node List Report**

Column	Data Type	Description
Batch Counter	Long	A sequential counter to help integrate information between related files. This value does reset when the reporter restarts and is not consistent across members. However, it is helpful when trying to integrate files.
Report Time	Date	The system time when the report executed.
Node Id	String	The numeric member identifier.
Unicast Address	String	The Unicast address for the member.
Member Name	String	The member name.
Process Name	String	The process name for the member.
Role Name	String	The role name for the member.
Machine Name	String	The computer name for the member.
Rack Name	String	The rack name for the member.
Site Name	String	The site name for the member.

**Table 6–8 (Cont.) Contents of the Node List Report**

Column	Data Type	Description
Refresh Time	Date/Time	The time which the information was refreshed from a remote member. If the time is different than the refresh time on other rows in the batch, the member did not respond in a timely matter. This is often caused by a member performing a garbage collection. Any information regarding a member with an "old" refresh date is questionable.

## Understanding the Proxy Report

The proxy file provides information about proxy servers and the information being transferred to clients. The Proxy file is a tab delimited file that is prefixed with the date and hour in YYYYMMDDHH format and post fixed with `-report-proxy.txt`. For example `2009013101-report-proxy.txt` would be created on January 31, 2009 at 1:00 AM. [Table 6–9](#) describes the content of the Proxy report.

**Table 6–9 Contents of the Proxy Report**

Column	Type	Description
Batch Counter	Long	A sequential counter to help integrate information between related files. This value does reset when the reporter restarts and is not consistent across members. However, it is helpful when trying to integrate files.
Report Time	Date	The system time when the report executed.
Node Id	String	The numeric member identifier.
Service Name	String	The name of the proxy service.
HostIp	String	The IP Address and Port of the proxy service.
Connection Count	Long	The current number of connections to the proxy service.
Outgoing Byte Backlog	Long	The number of bytes queued to be sent by the proxy service.
Outgoing Message Backlog	Long	The number of messages queued by the proxy service.
Bytes Sent	Long	The number of bytes sent by the proxy service since the last execution of the report.
Bytes Received	Long	The number of bytes received by the proxy service since the last execution of the report.
Messages Sent	Long	The number of messages sent by the proxy service since the last execution of the report.
Messages Received	Long	The number of messages received by the proxy service since the last execution of the report.

## Understanding the Ram Journal Report

The ram journal report displays statistics that are used to determine how well data is being stored to RAM memory. The report is a tab-delimited file that is prefixed with the date and hour in YYYYMMDDHH format and post fixed with `-ramjournal.txt`. For example, `2010013113-ramjournal.txt` would be created on January 31, 2010 1:00 pm. [Table 6–10](#) describes the content of the cache utilization report.

**Table 6–10 Contents of the Ram Journal Report**

Column	Type	Description
Batch Counter	Long	A sequential counter to help integrate information between related files. This value does reset when the reporter restarts and is not consistent across members. However, it is helpful when trying to integrate files.
Report Time	Date	The system time when the report executed.
Node ID	Long	The member for the flash journal statistics.
FileCount	Integer	The number of journal files that are currently in use.
BinaryStoreCount	Integer	The number of active <code>JournalBinaryStore</code> objects that are using this journal.
TotalDataSize	Long	The amount of data, in bytes, that are currently stored for this journal.
TotalFileSize	Long	The total size of all journal files for this journal.

## Understanding the Service Report

The service report provides information to the requests processed, request failures, and request backlog, tasks processed, task failures and task backlog. Request Count and Task Count are useful to determine performance and throughput of the service. RequestPendingCount and Task Backlog are useful in determining capacity issues or blocked processes. Task Hung Count, Task Timeout Count, Thread Abandoned Count, Request Timeout Count are the number of unsuccessful executions that have occurred in the system. [Table 6–11](#) describes the contents of the Service report.

**Table 6–11 Contents of the Service Report**

Column	Data Type	Description
Batch Counter	Long	A sequential counter to help integrate information between related files. This value does reset when the reporter restarts and is not consistent across members. However, it is helpful when trying to integrate files.
Report Time	Date	The system time when the report executed.
Service	String	The service name.
Node Id	String	The numeric member identifier.
Refresh Time	Date	The system time when the service information was updated from a remote member.
Request Count	Long	The number of requests since the last report execution.
RequestPendingCount	Long	The number of pending requests at the time of the report.
RequestPendingDuration	Long	The duration for the pending requests at the time of the report.
Request Timeout Count	Long	The number of request timeouts since the last report execution.
Task Count	Long	The number of tasks executed since the last report execution.
Task Backlog	Long	The task backlog at the time of the report execution.

**Table 6–11 (Cont.) Contents of the Service Report**

<b>Column</b>	<b>Data Type</b>	<b>Description</b>
Task Timeout Count	Long	The number of task timeouts since the last report execution.
Task Hung Count	Long	The number of tasks that hung since the last report execution.
Thread Abandoned Count	Long	The number of threads abandoned since the last report execution.

---

---

## Coherence MBeans Reference

This appendix provides a reference of the Coherence MBeans that are used to manage and monitor different parts of Coherence. All of the MBeans' attributes and operations are described in detail.

In addition to this reference, Coherence MBeans are described in the `Registration` interface, see *Oracle Coherence Java API Reference*. An MBean-capable agent (such as JConsole) can also be used to view MBean information. See [Chapter 2, "Using JMX to Manage Coherence,"](#) for more information on accessing Coherence MBeans.

The following sections are included in this Appendix:

- [MBean Index](#)
- [CacheMBean](#)
- [ClusterMBean](#)
- [ClusterNodeMBean](#)
- [ConnectionManagerMBean](#)
- [ConnectionMBean](#)
- [FlashJournalRM](#)
- [ManagementMBean](#)
- [PointToPointMBean](#)
- [RamJournalRM](#)
- [ReporterMBean](#)
- [ServiceMBean](#)
- [StorageManagerMBean](#)
- [TransactionManagerMBean](#)

### MBean Index

[Table A-1](#) lists the MBeans that are provided for managing Coherence.

**Table A-1** *Coherence MBeans*

<b>MBean</b>	<b>Description</b>
<a href="#">CacheMBean</a>	Represents a cache. A cluster member may have zero or more instances of this managed bean.
<a href="#">ClusterMBean</a>	Represents a cluster object. Each cluster member has a single instance of this managed bean.
<a href="#">ClusterNodeMBean</a>	Represents a cluster member. Each cluster member has a single instance of this managed bean.
<a href="#">ConnectionManagerMBean</a>	Represents a Coherence*Extend proxy. A cluster member may have zero or more instances of this managed bean.
<a href="#">ConnectionMBean</a>	Represents a remote client connection through Coherence*Extend. A cluster member may have zero or more instances of this managed bean.
<a href="#">FlashJournalRM</a>	Represents a flash journal resource manager. Each cluster member has a single instance of this managed bean.
<a href="#">ManagementMBean</a>	Represents the grid JMX infrastructure. Each cluster member has a single instance of this managed bean.
<a href="#">PointToPointMBean</a>	Represents the network status between two cluster members. Each cluster member has a single instance of this managed bean.
<a href="#">RamJournalRM</a>	Represents a RAM journal resource manager. Each cluster member has a single instance of this managed bean.
<a href="#">ReporterMBean</a>	Represents the Coherence JMX Reporter. Each cluster member has a single instance of this managed bean.
<a href="#">ServiceMBean</a>	Represents a clustered Service. A cluster member may have zero or more instances of this managed bean.
<a href="#">StorageManagerMBean</a>	Represents a storage instance for a storage-enabled distributed cache service. A cluster member may have zero or more instances of this managed bean.
<a href="#">TransactionManagerMBean</a>	Represents a transaction manager. A cluster member may have zero or more instances of this managed bean.

## CacheMBean

The CacheMBean MBean represents a cache. A cluster member may have zero or more instances of this managed bean depending on the number of caches that are configured for each data service type (distributed, replicated, and so on). This MBean provides operational and performance statistics for a cache. Some of the MBean's attributes are writable and allow the behavior of a cache to be changed in real time.

The MBean's object name is:

```
type=Cache,service=service name,name=cache name,nodeId=cluster node's id,tier=tier tag
```

---

**Note:** The CacheMBean MBean contains many attributes that are not applicable to transactional caches. A transactional cache returns a -1 value if the non-applicable attributes are invoked. See *Oracle Coherence Developer's Guide* for a list of the supported attributes and transaction-specific descriptions.

---

## Attributes

Table A-2 describes the attributes for CacheMBean.

**Table A-2** CacheMBean Attributes

Attribute	Type	Access	Description
AverageGetMillis	Double	RO	The average number of milliseconds per <code>get()</code> invocation since the statistics were last reset.
AverageHitMillis	Double	RO	The average number of milliseconds per <code>get()</code> invocation that is a hit since the statistics were last reset.
AverageMissMillis	Double	RO	The average number of milliseconds per <code>get()</code> invocation that is a miss since the statistics were last reset.
AveragePutMillis	Double	RO	The average number of milliseconds per <code>put()</code> invocation since the statistics were last reset.
BatchFactor	Double	RW	The BatchFactor attribute is used to calculate the soft-ripe time for write-behind queue entries. A queue entry is considered to be ripe for a write operation if it has been in the write-behind queue for no less than the <code>QueueDelay</code> interval. The soft-ripe time is the point in time before the actual ripe time after which an entry is included in a batch asynchronous write operation to the cache store (along with all other ripe and soft-ripe entries). This attribute is only applicable if asynchronous writes are enabled (that is, the value of the <code>QueueDelay</code> attribute is greater than zero) and the cache store implements the <code>storeAll()</code> method. The value of the element is expressed as a percentage of the <code>QueueDelay</code> interval. Valid values are doubles in the interval [0.0, 1.0].
CacheHits	Long	RO	The rough number of cache hits since the statistics were last reset. A cache hit is a read operation invocation (that is, <code>get()</code> ) for which an entry exists in this map.

**Table A-2 (Cont.) CacheMBean Attributes**

<b>Attribute</b>	<b>Type</b>	<b>Access</b>	<b>Description</b>
CacheHitsMillis	Long	RO	The total number of milliseconds (since the statistics were last reset) for the <code>get()</code> operations for which an entry existed in this map.
CacheMisses	Long	RO	The rough number of cache misses since the statistics were last reset.
CacheMissesMillis	Long	RO	The total number of milliseconds (since the statistics were last reset) for the <code>get()</code> operations for which no entry existed in this map.
CachePrunes	Long	RO	The number of prune operations since the statistics were last reset. A prune operation occurs every time the cache reaches its high watermark as specified by the <code>HighUnits</code> attribute.
CachePrunesMillis	Long	RO	The total number of milliseconds for the prune operations since the statistics were last reset.
Description	String	RO	The cache description.
ExpiryDelay	Integer	RW	The time-to-live for cache entries in milliseconds. A value of zero indicates that the automatic expiry is disabled. Changing this attribute does not affect existing entries that are scheduled to expire.
HighUnits	Integer	RW	The limit of the cache size measured in units. The cache prunes itself automatically after it reaches its maximum unit level. This is often referred to as the high water mark of the cache.
HitProbability	Double	RO	The rough probability ( $0 \leq p \leq 1$ ) that the next invocation is a hit, based on the statistics collected since the last time statistics were reset.
LowUnits	Integer	RW	The number of units to which the cache shrinks when it prunes. This is often referred to as a low water mark of the cache.
PersistenceType	String	RO	The persistence type for this cache. Possible values include: NONE, READ-ONLY, WRITE-THROUGH, WRITE-BEHIND.
QueueDelay	Integer	RW	The number of seconds that an entry added to a write-behind queue sits in the queue before being stored using a cache store. This attribute is only applicable if the persistence type is WRITE-BEHIND.
QueueSize	Integer	RO	The size of the write-behind queue size. This attribute is only applicable if the persistence type is WRITE-BEHIND.
RefreshFactor	Double	RW	This attribute is used to calculate the soft-expiration time for cache entries. Soft-expiration is the point in time before the actual expiration after which any access request for an entry schedules an asynchronous load request for the entry. This attribute is only applicable for a read write backing map which has an internal local cache with scheduled automatic expiration. The value of this element is expressed as a percentage of the internal local cache expiration interval. Valid values are doubles in the interval [0.0, 1.0]. If zero, refresh-ahead scheduling is disabled.
RefreshTime	Date	RO	The timestamp when this model was last retrieved from a corresponding member. For local servers it is the local time.

**Table A-2 (Cont.) CacheMBean Attributes**

<b>Attribute</b>	<b>Type</b>	<b>Access</b>	<b>Description</b>
RequeueThreshold	Integer	RW	The maximum size of the write-behind queue for which failed cache store write operations are requeued. If zero, the write-behind requeueing is disabled. This attribute is only applicable if the persistence type is WRITE-BEHIND.
Size	Integer	RO	The number of entries in the cache.
StoreAverageBatchSize	Long	RO	The average number of entries stored for each cache store write operation since the statistics were last reset. A call to the <code>store()</code> method is counted as a batch of one, whereas a call to the <code>storeAll()</code> method is counted as a batch of the passed Map size. The value is -1 if the persistence type is NONE.
StoreAverageReadMillis	Long	RO	The average time (in milliseconds) spent per read operation since the statistics were last reset. The value is -1 if the persistence type is NONE.
StoreAverageWriteMillis	Long	RO	The average time (in milliseconds) spent per write operation since the statistics were last reset. The value is -1 if the persistence type is NONE.
StoreFailures	Long	RO	The total number of cache store failures (load, store and erase operations) since the statistics were last reset. The value is -1 if the persistence type is NONE.
StoreReadMillis	Long	RO	The cumulative time (in milliseconds) spent on load operations since the statistics were last reset. The value is -1 if the persistence type is NONE.
StoreReads	Long	RO	The total number of load operations since the statistics were last reset. The value is -1 if the persistence type is NONE.
StoreWriteMillis	Long	RO	The cumulative time (in milliseconds) spent on store and erase operations since the statistics were last reset. The value is -1 if the persistence type is NONE or READ-ONLY.
StoreWrites	Long	RO	The total number of store and erase operations since the statistics were last reset. The value is -1 if the persistence type is NONE or READ-ONLY.
TotalGets	Long	RO	The total number of <code>get()</code> operations since the statistics were last reset.
TotalGetsMillis	Long	RO	The total number of milliseconds spent on <code>get()</code> operations since the statistics were last reset.
TotalPuts	Long	RO	The total number of <code>put()</code> operations since the statistics were last reset.
TotalPutsMillis	Long	RO	The total number of milliseconds spent on <code>put()</code> operations since the statistics were last reset.
UnitFactor	Integer	RO	The factor by which the <code>Units</code> , <code>LowUnits</code> and <code>HighUnits</code> properties are adjusted. Using a BINARY unit calculator, for example, the factor of 1048576 could be used to count megabytes instead of bytes.
Units	Integer	RO	The size of the cache measured in units. This value must be adjusted by the <code>UnitFactor</code> .

## Operations

The CacheMBean MBean includes a `resetStatistics` operation that resets all cache statistics.

## ClusterMBean

The `ClusterMBean` MBean represents a cluster. Each cluster member has a single instance of this managed bean. This MBean provides operational statistics about the cluster.

The MBean's object name is:

```
type=Cluster
```

### Attributes

[Table A-3](#) describes the attributes for `ClusterMBean`.

**Table A-3** *ClusterMBean Attributes*

Attribute	Type	Access	Description
<code>ClusterName</code>	<code>String</code>	RO	The name of the cluster.
<code>ClusterSize</code>	<code>Integer</code>	RO	The total number of cluster members.
<code>LicenseMode</code>	<code>String</code>	RO	The license mode that this cluster is using. Possible values are <code>Evaluation</code> , <code>Development</code> or <code>Production</code> .
<code>LocalMemberId</code>	<code>Integer</code>	RO	The member id for the cluster member that is collocated with the reporting MBean server. The value is <code>-1</code> if the cluster service is not running.
<code>MemberIds</code>	<code>Integer[]</code>	RO	An array of all existing cluster member ids.
<code>Members</code>	<code>String[]</code>	RO	An array of all existing cluster members.
<code>MembersDeparted</code>	<code>String[]</code>	RO	An array of strings containing the member information for recently departed cluster members. Members are removed from this array when the member id is recycled. This information is since the member has joined the cluster and is reset when the MBean server member leaves and rejoins the cluster. The <code>MembersDepartureCount</code> is the total count of departed members and not the size of this array.
<code>MembersDepartureCount</code>	<code>Long</code>	RO	The number of times this member has observed another member's departure from the cluster since this management member has joined the cluster or statistics have been reset.
<code>OldestMemberId</code>	<code>Integer</code>	RO	The senior cluster member id. The value is <code>-1</code> if the cluster service is not running.
<code>RefreshTime</code>	<code>Date</code>	RO	The timestamp when this model was last retrieved from a corresponding member. For local servers it is the local time.
<code>Running</code>	<code>Boolean</code>	RO	Specifies whether the cluster is running.
<code>Version</code>	<code>String</code>	RO	The Coherence version.

### Operations

[Table A-4](#) describes the operations for `ClusterMBean`.

**Table A-4 ClusterMBean Operations**

<b>Operation</b>	<b>Parameters</b>	<b>Return Type</b>	<b>Description</b>
ensureRunning	NA	Void	Ensures that the cluster service is running on this member.
shutdown	NA	Void	Shuts down the cluster service on this member.

## ClusterNodeMBean

The `ClusterNodeMBean` MBean represents a cluster member. Each cluster member has a single instance of this managed bean. This MBean provides many operational and performance statistics for a member of a cluster. Many of the attributes are writable and allow the behavior of the member to be changed in real time.

The MBean's object name is:

```
type=Node,nodeId=cluster node's id
```

### Attributes

Table A-5 describes the attributes for `ClusterNodeMBean`.

**Table A-5 ClusterNodeMBean Attributes**

Attribute	Type	Access	Description
<code>BufferPublishSize</code>	Integer	RW	The buffer size of the unicast datagram socket used by the Publisher, measured in the number of packets. Changing this value at run time is an inherently unsafe operation that pauses all network communications and may result in the termination of all cluster services.
<code>BufferReceiveSize</code>	Integer	RW	The buffer size of the unicast datagram socket used by the Receiver, measured in the number of packets. Changing this value at run time is an inherently unsafe operation that pauses all network communications and may result in the termination of all cluster services.
<code>CpuCount</code>	Integer	RO	Number of CPU cores for the computer this member is running on.
<code>FlowControlEnabled</code>	Boolean	RO	Indicates whether <code>FlowControl</code> is enabled. To enable flow control, see the <code>&lt;flow-control&gt;</code> element in <i>Oracle Coherence Developer's Guide</i> .
<code>Id</code>	Integer	RO	The short member id that uniquely identifies the member now and does not change for the life of this member.
<code>LoggingDestination</code>	String	RO	The output device used by the logging system. Valid values are <code>stdout</code> , <code>stderr</code> , <code>jdk</code> , <code>log4j</code> , or a file name.
<code>LoggingFormat</code>	String	RW	Specifies how messages are formatted before being passed to the log destination.
<code>LoggingLevel</code>	Integer	RW	Specifies which logged messages are output to the log destination. Valid values are nonnegative integers. A value of <code>-1</code> to disable all logger output.
<code>LoggingLimit</code>	Integer	RW	The maximum number of characters that the logger daemon processes from the message queue before discarding all remaining messages in the queue. Valid values are positive integers in the range 0 to <code>Integer.MAX_VALUE</code> (2147483647). Zero implies <code>Integer.MAX_VALUE</code> .
<code>MachineId</code>	Integer	RO	The member's machine Id.
<code>MachineName</code>	String	RO	A configured name that should be the same for all members that are on the same physical computer, and different for members that are on different physical computers.
<code>MemberName</code>	String	RO	A configured name that must be unique for every member.

**Table A-5 (Cont.) ClusterNodeMBean Attributes**

Attribute	Type	Access	Description
MemoryAvailableMB	Integer	RO	The total amount of memory in the JVM available for new objects in MB.
MemoryMaxMB	Integer	RO	The maximum amount of memory that the JVM attempts to use in MB.
MulticastAddress	String	RO	The IP address of the member's multicast socket for group communication.
MulticastEnabled	Boolean	RO	Specifies whether this member uses multicast for group communication. If <code>false</code> , this member uses the addresses listed in the <code>WellKnownAddresses</code> attribute to join the cluster and point-to-point unicast to communicate with other members of the cluster.
MulticastPort	Integer	RO	The port of the member's multicast socket for group communication.
MulticastTTL	Integer	RO	The time-to-live for multicast packets sent out on this member's multicast socket.
MulticastThreshold	Integer	RW	The percentage (0 to 100) of the servers in the cluster that a packet is sent to, above which the packet is sent using multicast and below which it is sent using unicast.
NackEnabled	Boolean	RO	Indicates whether the early packet loss detection protocol is enabled.
NackSent	Long	RO	The total number of NACK packets sent since the member statistics were last reset.
PacketDeliveryEfficiency	Float	RO	The efficiency of packet loss detection and retransmission. A low efficiency is an indication that there is a high rate of unnecessary packet retransmissions.
PacketsBundled	Long	RO	The total number of packets which were bundled before transmission. The total number of network transmissions equals <code>PacketsSent - PacketsBundled</code> .
PacketsReceived	Long	RO	The number of packets received since the member statistics were last reset.
PacketsRepeated	Long	RO	The number of duplicate packets received since the member statistics were last reset.
PacketsResent	Long	RO	The number of packets resent since the member statistics were last reset. A packet is resent when there is no ACK received within a timeout period.
PacketsResentEarly	Long	RO	The total number of packets resent too soon. A packet is resent too soon when there is a NACK indicating that the packet has not been received.
PacketsResentExcess	Long	RO	The total number of packet retransmissions which were later proven unnecessary.
PacketsSent	Long	RO	The number of packets sent since the member statistics were last reset.
Priority	Integer	RO	The priority, or weight, of the member; used to determine tie-breakers.
ProcessName	String	RO	A configured name that should be the same for members that are in the same process (JVM), and different for members that are in different processes. If not explicitly provided, the name is calculated internally as the <code>Name</code> attribute of the system <code>RuntimeMXBean</code> , which normally represents the process identifier (PID).

**Table A-5 (Cont.) ClusterNodeMBean Attributes**

Attribute	Type	Access	Description
ProductEdition	String	RO	The product edition this member is running. Possible values are: Standard Edition, Enterprise Edition, Grid Edition.
PublisherPacketUtilization	Float	RO	The publisher packet utilization for this cluster member since the member socket was last reopened. This value is a ratio of the number of bytes sent to the number that would have been sent had all packets been full. A low utilization indicates that data is not being sent in large enough chunks to make efficient use of the network.
PublisherSuccessRate	Float	RO	The publisher success rate for this cluster member since the member statistics were last reset. The publisher success rate is a ratio of the number of packets successfully delivered in a first attempt to the total number of sent packets. A failure count is incremented when there is no ACK received within a timeout period. It could be caused by either very high network latency or a high packet drop rate.
QuorumStatus	String	RO	The current state of the cluster quorum.
RackName	String	RO	A configured name that should be the same for members that are on the same physical "rack" (or frame or cage), and different for members that are on different physical "racks".
ReceiverPacketUtilization	Float	RO	The receiver packet utilization for this cluster member since the socket was last reopened. This value is a ratio of the number of bytes received to the number that would have been received had all packets been full. A low utilization indicates that data is not being sent in large enough chunks to make efficient use of the network.
ReceiverSuccessRate	Float	RO	The receiver success rate for this cluster member since the member statistics were last reset. The receiver success rate is a ratio of the number of packets successfully acknowledged in a first attempt to the total number of received packets. A failure count is incremented when a re-delivery of previously received packet is detected. It could be caused by either very high inbound network latency or lost ACK packets.
RefreshTime	Date	RO	The timestamp when this model was last retrieved from a corresponding member. For local servers it is the local time.
ResendDelay	Integer	RW	The minimum number of milliseconds that a packet remains queued in the Publisher's re-send queue before it is resent to the recipient(s) if the packet has not been acknowledged. Setting this value too low can overflow the network with unnecessary repetitions. Setting the value too high can increase the overall latency by delaying the re-sends of dropped packets. Additionally, change of this value may require a change in the SendAckDelay value.
RoleName	String	RO	A configured name that can indicate the role of a member to the application. While managed by Coherence, this property is only used by the application.
SendAckDelay	Integer	RW	The minimum number of milliseconds between the queuing and sending of an ACK packet. This value should not be more than a half the ResendDelay value.

**Table A-5 (Cont.) ClusterNodeMBean Attributes**

Attribute	Type	Access	Description
SendQueueSize	Integer	RO	The number of packets currently scheduled for delivery. This number includes both packets that are to be sent immediately and packets that have been sent and awaiting for acknowledgment. Packets that do not receive an acknowledgment within the <code>ResendDelay</code> interval are automatically resent.
SiteName	String	RO	A configured name that should be the same for members that are on the same physical site (for example, data center), and different for members that are on different physical sites.
SocketCount	Integer	RO	Number of CPU sockets for the computer this member is running on.
Statistics	String	RO	Statistics for this cluster member in a human readable format.
TcpRingFailures	Long	RO	The number of recovered TcpRing disconnects since the member statistics were last reset. A recoverable disconnect is an abnormal event that is registered when the TcpRing peer drops the TCP connection, but recovers after no more than the maximum configured number of attempts. A -1 value indicates that TcpRing is disabled.
Timestamp	Date	RO	The date/time value (in cluster time) that this member joined the cluster.
TrafficJamCount	Integer	RW	The maximum total number of packets in the send and resend queues that forces the publisher to pause client threads. Zero means no limit.
TrafficJamDelay	Integer	RW	The number of milliseconds to pause client threads when a traffic jam condition has been reached. Anything less than one (for example, zero) is treated as one millisecond.
UnicastAddress	String	RO	The IP address of the member's datagram socket for point-to-point communication.
UnicastPort	Integer	RO	The port of the member's datagram socket for point-to-point communication.
WeakestChannel	Integer	RO	The id of the cluster member to which this member is having the most difficulty communicating, or -1 if none is found. A channel is considered to be weak if either the point-to-point publisher or receiver success rates are below 1.0.
WellKnownAddresses	String[]	RO	An array of well-known socket addresses that this member uses to join the cluster.

## Operations

[Table A-6](#) describes the operations for `ClusterNodeMBean`.

**Table A-6 ClusterNodeMBean Operations**

Operation	Parameters	Return Type	Description
ensureService	String sCacheName	void	Ensure that a cache service for the specified cache runs at the cluster member represented by this MBean. This method uses the configurable cache factory to find out which cache service to start if necessary.
resetStatistics	NA	void	Reset the cluster member statistics.
shutdown	NA	void	Stop all the clustered services running at this member (controlled shutdown). The management of this member is not available until the member is restarted (manually or programmatically).

---

## ConnectionManagerMBean

The `ConnectionManagerMBean` MBean represents a Coherence\*Extend proxy. A cluster member may have zero or more instances of this managed bean depending on the number of configured proxies. The MBean contains statistics for throughput and connection information for proxy hosts.

The MBean's object name is:

`type=ConnectionManager,name=service name,nodeId=cluster node's id`

### Attributes

Table A-7 describes the attributes for `ConnectionManagerMBean`.

**Table A-7** *ConnectionManagerMBean Attributes*

Attribute	Type	Access	Description
<code>ConnectionCount</code>	Integer	RO	The number of client connections.
<code>HostIP</code>	String	RO	The IP address and port of the proxy host.
<code>IncomingBufferPoolCapacity</code>	Long	RO	The pool capacity (in bytes) of the incoming buffer.
<code>IncomingBufferPoolSize</code>	Integer	RO	The number of buffers in the incoming pool.
<code>OutgoingBufferPoolCapacity</code>	Long	RO	The pool capacity (in bytes) of the outgoing buffer.
<code>OutgoingBufferPoolSize</code>	Integer	RO	The number of buffers in the outgoing pool.
<code>OutgoingByteBacklog</code>	Long	RO	The backlog (in bytes) of the outgoing queue
<code>OutgoingMessageBacklog</code>	Long	RO	The backlog of the outgoing message queue.
<code>RefreshTime</code>	Date	RO	The timestamp when this model was last retrieved from a corresponding member. For local servers it is the local time.
<code>TotalBytesReceived</code>	Long	RO	The total number of bytes received by the proxy host since the statistics were last reset.
<code>TotalBytesSent</code>	Long	RO	The total number of bytes sent by the proxy host since the statistics were last reset.
<code>TotalMessagesReceived</code>	Long	RO	The total number of messages received by the proxy host since the statistics were last reset.
<code>TotalMessagesSent</code>	Long	RO	The total number of messages sent by the proxy host since the statistics were last reset.

### Operations

The `ConnectionManagerMBean` MBean has no operations.

## ConnectionMBean

The `ConnectionMBean` MBean represents a remote client connection through Coherence\*Extend. A cluster member may have zero or more instances of this managed bean depending on the number of active remote connections to the cluster. The MBean contains performance and usage statistics for the connection.

The MBean's object name is:

```
type=Connection,name=service name ,nodeId=cluster node's id,UUID=connection's id
```

### Attributes

Table A-8 describes the attributes for `ConnectionMBean`.

**Table A-8** *ConnectionMBean Attributes*

Attribute	Type	Access	Description
<code>ConnectionTimeMillis</code>	Long	RO	The time duration (in milliseconds) that the client has been connected.
<code>Member</code>	String	RO	The client's member identity information. The value can be customized using the <code>&lt;member-identity&gt;</code> element of the client's operational override file. See <i>Oracle Coherence Developer's Guide</i> for details on setting up member identity information.
<code>OutgoingByteBacklog</code>	Long	RO	The backlog (in bytes) of the outgoing queue
<code>OutgoingMessageBacklog</code>	Integer	RO	The backlog of the outgoing message queue.
<code>RefreshTime</code>	Date	RO	The timestamp when this model was last retrieved from a corresponding member. For local servers it is the local time.
<code>RemoteAddress</code>	String	RO	The IP address of the corresponding client.
<code>RemotePort</code>	Integer	RO	The port of the corresponding client.
<code>Timestamp</code>	Date	RO	The date/time value (in local time) that the corresponding client connected to the proxy.
<code>TotalBytesReceived</code>	Long	RO	The total number of bytes received since the last time the statistics were reset.
<code>TotalBytesSent</code>	Long	RO	The total number of bytes sent since the last time the statistics were reset.
<code>TotalMessagesReceived</code>	Long	RO	The total number of messages received since the last time the statistics were reset.
<code>TotalMessagesSent</code>	Long	RO	The total number of messages sent since the last time the statistics were reset.
<code>UUID</code>	String	RO	The unique identifier for this connection.

### Operations

Table A-9 describes the operations for `ConnectionMBean`.

**Table A-9 ConnectionMBean Operations**

<b>Operation</b>	<b>Parameters</b>	<b>Return Type</b>	<b>Description</b>
closeConnection	NA	void	Close the corresponding connection.
resetStatistics	NA	void	Reset the connection statistics.

## FlashJournalRM

The `JournalMBean` MBean is a common interface that is used by the flash journal resource manager (`FlashJournalRM`). The resource manager is responsible for storing data to flash memory. Each cluster member has a single instance of this managed resource, which provides operational statistics.

The managed resource's object name is

```
type=Journal,name=FlashJournalRM,nodeId=cluster node's id
```

### Attributes

Table A-15 describes the attributes for `FlashJournalRMMBean`.

**Table A-10** *FlashJournalRMMBean Attributes*

Attribute	Type	Access	Description
<code>BacklogCount</code>	Integer	RO	The number of serialized values that have yet to be persisted.
<code>BacklogSize</code>	Integer	RO	The maximum size, in bytes, of the backlog. The backlog is the amount of serialized values that has yet to be persisted. Client threads are blocked if this limit is exceeded and remain blocked until the backlog recedes below this limit.
<code>BinaryStoreCount</code>	Integer	RO	The number of active <code>JournalBinaryStore</code> objects that are using this journal.
<code>BufferSize</code>	Integer	RO	The size of the buffers that are used to write data to a journal file.
<code>CollectorLoadFactor</code>	Double	RO	The threshold after which files are eligible for garbage collection. The larger the value, the more aggressively files are collected.
<code>FileCount</code>	Integer	RO	The number of journal files that are currently in use.
<code>MaxBacklogSize</code>	Integer	RO	The maximum size, in bytes, of the backlog. Writes are delayed when the backlog exceeds this level and remain delayed until the backlog recedes below this level.
<code>MaxFileSize</code>	Long	RO	The maximum allowable size of an individual journal file.
<code>MaxJournalFilesNumber</code>	Integer	RO	The maximum number of journal files that can be used.
<code>MaxPoolSize</code>	Integer	RO	The maximum size, in bytes, of the buffer pool.
<code>MaxTotalRam</code>	Long	RO	This attribute does not pertain to a flash journal and returns -1.
<code>MaxValueSize</code>	Integer	RO	The maximum allowable size, in bytes, for serialized values.
<code>NioRam</code>	Boolean	RO	Whether the RAM journal uses on-heap byte buffers or off-heap NIO buffers (NIO RAM). Valid values are <code>true</code> and <code>false</code> . A <code>true</code> value indicates off-heap NIO buffers. A <code>false</code> value indicates on-heap byte buffers. The default value is <code>false</code> .

**Table A-10 (Cont.) FlashJournalRMMBean Attributes**

<b>Attribute</b>	<b>Type</b>	<b>Access</b>	<b>Description</b>
PoolSize	Integer	RO	The total size, in bytes, of all available buffers in the pool.
TotalDataSize	Long	RO	The amount of data, in bytes, that are currently stored for this journal.
TotalFileSize	Long	RO	The total size of all journal files for this journal.

**Operations**

There are no operations available for this managed resource.

## ManagementMBean

The `ManagementMBean` MBean represents the grid JMX infrastructure. Each cluster member has a single instance of this managed bean. The MBean contains management settings. Some of the attributes are writable and allow management behavior to be changed in real time.

The MBean's object name is:

```
type=Management
```

### Attributes

[Table A-11](#) describes the attributes for `ManagementMBean`.

**Table A-11** *ManagementMBean Attributes*

Attribute	Type	Access	Description
<code>ExpiryDelay</code>	Long	RW	The number of milliseconds that the MBean server keeps a remote model snapshot before refreshing.
<code>RefreshCount</code>	Long	RO	The total number of snapshots retrieved since the statistics were last reset.
<code>RefreshExcessCount</code>	Long	RO	The number of times the MBean server predictively refreshed information and the information was not accessed.
<code>RefreshOnQuery</code>	Boolean	RO	Specifies whether the refresh-on-query MBean server is configured. If this is <code>true</code> then the <code>RefreshPolicy</code> value should be <code>refresh-onquery</code> .
<code>RefreshPolicy</code>	String	RW	The policy used to determine the behavior when refreshing remote models. Valid values are: <code>refresh-ahead</code> , <code>refresh-behind</code> , <code>refresh-expired</code> , <code>refresh-onquery</code> . Invalid values are converted to <code>refresh-expired</code> .
<code>RefreshPredictionCount</code>	Long	RO	The number of times the MBean server used a predictive ( <code>refresh-behind</code> , <code>refresh-ahead</code> , <code>refresh-onquery</code> ) algorithm to refresh MBean information.
<code>RefreshTime</code>	Date	RO	The timestamp when this model was last retrieved from a corresponding member. For local servers it is the local time.
<code>RefreshTimeoutCount</code>	Long	RO	The number of times this management member has timed out while attempting to refresh remote MBean attributes.
<code>RemoteNotificationCount</code>	Long	RO	The total number of remote notifications received for all MBeans by this member since the last time the statistics were reset.

### Operations

The `ManagementMBean` MBean includes a `resetStatistics` operation that resets the `RefreshCount`, `RefreshExcessCount` and `RefreshPredictionCount` statistics.

## PointToPointMBean

The `PointToPointMBean` MBean represents the network status between two cluster members. Each cluster member has a single instance of this managed bean. The MBean provides network statistics from the perspective of the current viewing member to a specified viewed member. To specify the member, enter its ID using the `ViewedMemberId` attribute.

The MBean's object name is:

```
type=PointToPoint,nodeId=cluster node's id
```

### Attributes

Table A-12 describes the attributes for `PointToPointMBean`.

**Table A-12** *PointToPointMBean Attributes*

Attribute	Type	Access	Description
<code>DeferredPackets</code>	Integer	RO	The number of packets addressed to the viewed member that the viewing member is currently deferring to send. The viewing member delays sending these packets until the number of outstanding packets falls below the value of the <code>Threshold</code> attribute. The value of this attribute is only meaningful if the viewing member has <code>FlowControl</code> enabled. See the <code>&lt;flow-control&gt;</code> element in <i>Oracle Coherence Developer's Guide</i> .
<code>Deferring</code>	Boolean	RO	Indicates whether the viewing member is currently deferring packets to the viewed member. The value of this attribute is only meaningful if the viewing member has <code>FlowControl</code> enabled. See the <code>&lt;flow-control&gt;</code> element in <i>Oracle Coherence Developer's Guide</i> .
<code>LastIn</code>	Long	RO	The number of milliseconds that have elapsed since the viewing member last received an acknowledgment from the viewed member.
<code>LastOut</code>	Long	RO	The number of milliseconds that have elapsed since the viewing member last sent a packet to the viewed member.
<code>LastSlow</code>	Long	RO	The number of milliseconds that have elapsed since the viewing member declared the viewed member as slow, or -1 if the viewed member has never been declared slow.
<code>OutstandingPackets</code>	Integer	RO	The number of packets that the viewing member has sent to the viewed member which have yet to be acknowledged. The value of this attribute is only meaningful if the viewing member has <code>FlowControl</code> enabled. See the <code>&lt;flow-control&gt;</code> element in <i>Oracle Coherence Developer's Guide</i> .
<code>PauseRate</code>	Float	RO	The percentage of time since the last time statistics were reset in which the viewing member considered the viewed member to be unresponsive. Under normal conditions this value should be very close to 0.0. Values near 1.0 would indicate that the viewed member is nearly inoperable, likely due to extremely long GC pauses. The value of this attribute is only meaningful if the viewing member has <code>FlowControl</code> enabled. See the <code>&lt;flow-control&gt;</code> element in <i>Oracle Coherence Developer's Guide</i> .

**Table A–12 (Cont.) PointToPointMBean Attributes**

Attribute	Type	Access	Description
Paused	Boolean	RO	Indicates whether the viewing member currently considers the viewed member to be unresponsive. The value of this attribute is only meaningful if the viewing member has <code>FlowControl</code> enabled. See the <code>&lt;flow-control&gt;</code> element in <i>Oracle Coherence Developer's Guide</i> .
PublisherSuccessRate	Float	RO	The publisher success rate from the viewing member to the viewed member since the statistics were last reset.
ReceiverSuccessRate	Float	RO	The receiver success rate from the viewing member to the viewed member since the statistics were last reset.
RefreshTime	Date	RO	The timestamp when this model was last retrieved from a corresponding member. For local servers it is the local time.
Threshold	Integer	RO	The maximum number of outstanding packets for the viewed member that the viewing member is allowed to accumulate before initiating the deferral algorithm. The value of this attribute is only meaningful if the viewing member has <code>FlowControl</code> enabled. See the <code>&lt;flow-control&gt;</code> element in <i>Oracle Coherence Developer's Guide</i> .
ViewedMemberId	Integer	RW	The Id of the member being viewed.
ViewerStatistics	String[]	RO	Human readable summary of the point-to-point statistics from the viewing member for all other members.

## Operations

[Table A–13](#) describes the operations for `PointToPointMBean`.

**Table A–13 PointToPointMBean Operations**

Operation	Parameters	Return Type	Description
<code>resetStatistics</code>	NA	void	Reset the viewing member's point-to-point statistics for all other members.
<code>trackWeakest</code>	NA	void	Instruct the Point-to-Point MBean to track the weakest member. A viewed member is considered to be weak if either the corresponding publisher or receiver success rates are below 1.0.

## RamJournalRM

The `JournalMBean` MBean is a common interface that is used by the RAM journal resource manager (`RamJournalRM`). The resource manager is responsible for storing data to RAM memory. Each cluster member has a single instance of this managed resource, which provides operational statistics.

The managed resource's object name is:

```
type=Journal,name=RamJournalRM,nodeId=cluster node's id
```

### Attributes

[Table A-15](#) describes the attributes for `RamJournalRMMBean`.

**Table A-14** *RamJournalRMMBean Attributes*

Attribute	Type	Access	Description
<code>BacklogCount</code>	Integer	RO	This attribute does not pertain to a RAM journal and returns -1.
<code>BacklogSize</code>	Integer	RO	This attribute does not pertain to a RAM journal and returns -1.
<code>BinaryStoreCount</code>	Integer	RO	The number of active <code>JournalBinaryStore</code> objects that are using this journal.
<code>BufferSize</code>	Integer	RO	This attribute does not pertain to a RAM journal and returns -1.
<code>CollectorLoadFactor</code>	Double	RO	The threshold after which files are eligible for garbage collection. The larger the value, the more aggressively files are collected.
<code>FileCount</code>	Integer	RO	The number of journal files that are currently in use.
<code>MaxBacklogSize</code>	Integer	RO	This attribute does not pertain to a RAM journal and returns -1.
<code>MaxFileSize</code>	Long	RO	The maximum allowable size of an individual journal file.
<code>MaxJournalFilesNumber</code>	Integer	RO	The maximum number of journal files that can be used.
<code>MaxPoolSize</code>	Integer	RO	The maximum size, in bytes, of the buffer pool.
<code>MaxTotalRam</code>	Long	RO	The total amount of RAM used for this journal.
<code>MaxValueSize</code>	Integer	RO	The maximum allowable size, in bytes, for serialized values.
<code>NioRam</code>	Boolean	RO	Whether the RAM journal uses on-heap byte buffers or off-heap NIO buffers (NIO RAM). Valid values are <code>true</code> and <code>false</code> . A <code>true</code> value indicates off-heap NIO buffers. A <code>false</code> value indicates on-heap byte buffers. The default value is <code>false</code> .
<code>PoolSize</code>	Integer	RO	This attribute does not pertain to a RAM journal and returns -1.
<code>TotalDataSize</code>	Long	RO	The amount of data, in bytes, that are currently stored for this journal.
<code>TotalFileSize</code>	Long	RO	The total size of all journal files for this journal.

**Operations**

There are no operations available for this managed resource.

## ReporterMBean

The `ReporterMBean` MBean represents the Coherence JMX Reporter. Each cluster member has a single instance of this managed bean. The MBean contains settings and statistics for JMX reporting. Many of the attributes are writable and allow JMX reporting to be changed in real time. In addition, the MBean contains operations that start and stop reporting and run reports in real time.

The MBean's object name is:

```
type=Reporter
```

### Attributes

[Table A-15](#) describes the attributes for `ReporterMBean`.

**Table A-15** *ReporterMBean Attributes*

Attribute	Type	Access	Description
<code>AutoStart</code>	Boolean	RO	Specifies whether JMX reporting starts automatically with the member.
<code>ConfigFile</code>	String	RW	The report group configuration file to be used for JMX reporting.
<code>CurrentBatch</code>	Long	RW	The batch identifier for the Reporter.
<code>IntervalSeconds</code>	Long	RW	The interval between executions in seconds.
<code>LastExectionTime</code>	Date	RO	The last time a report batch was executed. For local servers it is the local time.
<code>LastReport</code>	String	RO	The last report to execute.
<code>OutputPath</code>	String	RW	The path where report output is located.
<code>RefreshTime</code>	Date	RO	The last time that the reporter statistics were reset. For local servers it is the local time.
<code>Reports</code>	String[]	RO	The list of reports executed.
<code>RunAverageMillis</code>	Double	RO	The average batch run in milliseconds since the statistics were last reset.
<code>RunLastMillis</code>	Long	RO	The last batch run in milliseconds since the statistics were last reset.
<code>RunMaxMillis</code>	Long	RO	The maximum batch run in milliseconds since the statistics were last reset.
<code>State</code>	String	RO	The state of JMX reporting. Valid values are: <code>Running</code> (reports are being executed); <code>Waiting</code> (the reporter is waiting for the interval to complete); <code>Starting</code> (the reporter is being started); <code>Stopping</code> (the reporter is attempting to stop execution and waiting for running reports to complete); <code>Stopped</code> (the reporter is stopped); <code>Sleeping</code> (the reporter is sleeping).

### Operations

[Table A-16](#) describes the operations for `ReporterMBean`.

**Table A-16 ReporterMBean Operations**

<b>Operation</b>	<b>Parameters</b>	<b>Return Type</b>	<b>Description</b>
Stop	NA	void	Stop JMX reporting.
Start	NA	void	Start JMX reporting.
RunReport	String sReportFile	void	Run a report group or single report one time using the specified report group or report file, respectively (for example (reports/report-group.xml or reports/report-cache-size.xml).
resetStatistics	NA	void	Reset the reporter statistics.

## ServiceMBean

The `ServiceMBean` MBean represents a clustered Service. A cluster member may have zero or more instances of this managed bean depending on the number of clustered services that are started. The MBean contains usage and performance statistics for a service. Some of the attributes are writable and allow the behavior of a service to be changed in real time. In addition, the MBean contains operations that are used to start and stop a service in real time.

The MBean's object name is:

```
type=Service,name=service name,nodeId=cluster node's id
```

### Terminology

The terms *task* and *request* have unique definitions within Coherence. These definitions should be understood before setting the task-related and request-related attributes for `ServiceMBean`.

- **Task** — A task is an invoked object that executes on one or more members. The objects include filters, invocation agents (entry processors and aggregators), or single-pass agents (Invocable objects).
- **Request** — A request is the round trip required to complete a task. A request begins the moment a task is sent for execution by a client and includes the following:
  - The time it takes to deliver the request to an executing member (server).
  - The interval between the time the task is received and placed into a service queue until the execution starts.
  - The task execution time.
  - The time it takes to deliver a result back to the client.

### Attributes

[Table A-17](#) describes the attributes for `ServiceMBean`.

**Table A-17** *ServiceMBean Attributes*

Attribute	Type	Access	Description
BackupCount	Integer	RO	The number of backups for every cache storage.
BackupCountAfterWritebehind	Integer	RO	The number of members of the partitioned (distributed) cache service that retain backup data that does not require write-behind. That is, data that is not vulnerable to being lost even if the entire cluster were shut down.
OwnedPartitionsBackup	Integer	RO	The number of partitions that this member backs up (responsible for the backup storage).
OwnedPartitionsPrimary	Integer	RO	The number of partitions that this member owns (responsible for the primary storage).
PartitionsAll	Integer	RO	The total number of partitions that every cache storage is divided into.
PartitionsEndangered	Integer	RO	The total number of partitions that are not currently backed up.

**Table A-17 (Cont.) ServiceMBean Attributes**

Attribute	Type	Access	Description
PartitionsUnbalanced	Integer	RO	The total number of primary and backup partitions which remain to be transferred until the partition distribution across the storage enabled service members is fully balanced.
PartitionsVulnerable	Integer	RO	The total number of partitions that are backed up on the same computer where the primary partition owner resides.
QuorumStatus	String	RO	The current state of the service quorum.
RefreshTime	Date	RO	The timestamp when this model was last retrieved from a corresponding member. For local servers it is the local time.
RequestAverageDuration	Float	RO	The average duration (in milliseconds) of an individual synchronous request issued by the service since the last time the statistics were reset.
RequestMaxDuration	Long	RO	The maximum duration (in milliseconds) of a synchronous request issued by the service since the last time the statistics were reset.
RequestPendingCount	Long	RO	The number of pending synchronous requests issued by the service.
RequestPendingDuration	Long	RO	The duration (in milliseconds) of the oldest pending synchronous request issued by the service.
RequestTimeoutCount	Long	RO	The total number of timed-out requests since the last time the statistics were reset.
RequestTimeoutMillis	Long	RW	The default timeout value in milliseconds for requests that can be timed-out (for example, implement the <code>com.tangosol.net.PriorityTask</code> interface) but do not explicitly specify the request timeout value.
RequestTotalCount	Long	RO	The total number of synchronous requests issued by the service since the last time the statistics were reset.
Running	Boolean	RO	Specifies whether the service is running.
SeniorMemberId	Integer	RO	The service senior member id. The value is -1 if the service is not running.
Statistics	String	RO	Statistics for this service in human readable format.
StatusHA	String	RO	The High Availability (HA) status for this service. The value of <code>MACHINE-SAFE</code> means that all the cluster members running on any given computer could be stopped without data loss. The value of <code>NODE-SAFE</code> means that any cluster member could be stopped without data loss. The value of <code>ENDANGERED</code> indicates that abnormal termination of any cluster member that runs this service may cause data loss.
StorageEnabled	Boolean	RO	Specifies whether the local storage is enabled for this cluster member.
StorageEnabledCount	Integer	RO	Specifies the total number of cluster members running this service for which local storage is enabled.
TaskAverageDuration	Float	RO	The average duration (in milliseconds) of an individual task execution.
TaskBacklog	Integer	RO	The size of the backlog queue that holds tasks scheduled to be executed by a service thread.

**Table A-17 (Cont.) ServiceMBean Attributes**

Attribute	Type	Access	Description
TaskCount	Long	RO	The total number of executed tasks since the last time the statistics were reset.
TaskHungCount	Integer	RO	The total number of currently executing hung tasks.
TaskHungDuration	Long	RO	The longest currently executing hung task duration in milliseconds.
TaskHungTaskId	String	RO	The id of the longest currently executing hung task.
TaskHungThresholdMillis	Long	RW	The amount of time in milliseconds that a task can execute before it is considered hung. Note that a posted task that has not yet started is never considered as hung.  This attribute is applied only if a thread pool is started (that is, the <code>ThreadCount</code> value is <code>&gt; 0</code> ).
TaskMaxBacklog	Integer	RO	The maximum size of the backlog queue since the last time the statistics were reset.
TaskTimeoutCount	Integer	RO	The total number of timed-out tasks since the last time the statistics were reset.
TaskTimeoutMillis	Long	RW	The default timeout value in milliseconds for tasks that can be timed-out (for example, implement the <code>com.tangosol.net.PriorityTask</code> interface) but do not explicitly specify the task execution timeout value.  This attribute is applied only if a thread pool is started (that is, the <code>ThreadCount</code> value is <code>&gt; 0</code> ).
ThreadAbandonedCount	Integer	RO	The number of abandoned threads from the service thread pool. A thread is abandoned and replaced with a new thread if it executes a task for a period longer than the execution timeout and all attempts to interrupt it fail.
ThreadAverageActiveCount	Float	RO	The average number of active (not idle) threads in the service thread pool since the last time the statistics were reset.
ThreadCount	Integer	RW	The number of threads in the service thread pool. This attribute can only be changed in real time if a service is configured to use a thread pool (that is, a thread count <code>&gt; 0</code> ). If the value is 0 (the default) then only the service thread is used and a thread pool is never started. To initially set this value, configure the <code>thread-count</code> parameter either for the service in an operational override file or for a cache in the cache configuration file.
ThreadIdleCount	Integer	RO	The number of currently idle threads in the service thread pool.
Type	String	RO	The type identifier of the service.

## Operations

[Table A-18](#) describes the operations for `ServiceMBean`.

**Table A-18 ServiceMBean Operations**

<b>Operation</b>	<b>Parameters</b>	<b>Return Type</b>	<b>Description</b>
reportOwnership	fVerbose	String	Reports the partitions owned by the service on this node. The fVerbose parameter is used to format the ownership summary. Valid values are true or false. Setting the parameter to true includes the detailed ownership catalog.
resetStatistics	NA	void	Reset the service statistics.
shutdown	NA	void	Stop the service. This is a controlled shut-down, and is preferred to the stop method.
start	NA	void	Start the service.
stop	NA	void	Hard-stop the service. Use the shutdown method for normal service termination.

## StorageManagerMBean

The `StorageManagerMBean` MBean represents a storage instance for a storage-enabled distributed cache service. A Storage instance manages all index, listener, and lock information for the portion of the distributed cache managed by the local member. A cluster member may have zero or more instances of this managed bean depending on the number of configured distributed caches. The MBean contains usage statistics for the storage-enabled cache and also includes statistics for queries.

The MBean's object name is:

`type=StorageManager,service=service name,cache=cache name,nodeId=cluster node's id`

### Attributes

Table A-19 describes the attributes for `StorageManagerMBean`.

**Table A-19** *StorageManagerMBean Attributes*

Attribute	Type	Access	Description
<code>EventsDispatched</code>	Long	RO	The total number of events dispatched by the Storage Manager since the last time the statistics were reset.
<code>EvictionCount</code>	Long	RO	The number of evictions from the backing map managed by this Storage Manager caused by entries expiry or insert operations that would make the underlying backing map to reach its configured size limit. The eviction count is used to audit the cache size in a static system:  <i>Cache Size = Insert Count - Remove Count - Eviction Count</i>  Therefore, the eviction count is not reset by the reset statistics method.
<code>IndexInfo</code>	String[]	RO	An array of information for each index applied to the portion of the partitioned cache managed by the Storage Manager. Each element is a string value that includes a <code>ValueExtractor</code> description, ordered flag ( <code>true</code> to indicate that the contents of the index are ordered; <code>false</code> otherwise), and cardinality (number of unique values indexed).
<code>InsertCount</code>	Long	RO	The number of inserts into the backing map managed by this Storage Manager. In addition to standard inserts caused by <code>put</code> and <code>invoke</code> operations or synthetic inserts caused by <code>get</code> operations with read-through backing map topology, this counter is incremented when distribution transfers move resources into the underlying backing map and is decremented when distribution transfers move data out.  The insert count is used to audit the cache size in a static system:  <i>Cache Size = Insert Count - Remove Count - Eviction Count</i>  Therefore, the insert count is not reset by the reset statistics method.
<code>ListenerFilterCount</code>	Integer	RO	The number of filter-based listeners currently registered with the Storage Manager.

**Table A-19 (Cont.) StorageManagerMBean Attributes**

Attribute	Type	Access	Description
ListenerKeyCount	Integer	RO	The number of key-based listeners currently registered with the Storage Manager.
ListenerRegistrations	Long	RO	The total number of listener registration requests processed by the Storage Manager since the last time the statistics were reset.
LocksGranted	Integer	RO	The number of locks currently granted for the portion of the partitioned cache managed by the Storage Manager.
LocksPending	Integer	RO	The number of pending lock requests for the portion of the partitioned cache managed by the Storage Manager.
MaxQueryDescription	String	RO	A description of the query with the longest duration that exceeds the <code>MaxQueryThresholdMillis</code> attribute since the statistics were last reset.
MaxQueryDurationMillis	Long	RO	The number of milliseconds of the longest running query since the statistics were last reset.
MaxQueryThresholdMillis	Long	RW	A threshold, in milliseconds, for recording queries. The longest query that executes longer than this threshold is reported by <code>MaxQueryDescription</code> attribute. The default value is 30 ms.
NonOptimizedQueryAverageMillis	Long	RO	The average duration, in milliseconds, for non-optimized query execution since the cache statistics were last reset.
NonOptimizedQueryCount	Long	RO	The total number of parallel queries that could not be resolved (or that were partially resolved) using indexes since the statistics were last reset.
NonOptimizedQueryTotalMillis	Long	RO	The total execution time, in milliseconds for queries that could not be resolved (or that were partially resolved) using indexes since the statistics were last reset
OptimizedQueryAverageMillis	Long	RO	The average duration, in milliseconds, per optimized query execution since the statistics were last reset.
OptimizedQueryCount	Long	RO	The total number of queries that were fully resolved using indexes since the statistics were last reset.
OptimizedQueryTotalMillis	Long	RO	The total execution time, in milliseconds, for queries that were fully resolved using indexes since the statistics were last reset.
RefreshTime	Date	RO	The timestamp when this model was last retrieved from a corresponding member. For local servers it is the local time.
RemoveCount	Long	RO	<p>The number of removes from the backing map managed by this Storage Manager caused by operations such as <code>clear</code>, <code>remove</code> or <code>invoke</code>.</p> <p>The remove count is used to audit the cache size in a static system:</p> $\text{Cache Size} = \text{Insert Count} - \text{Remove Count} - \text{Eviction Count}$ <p>Therefore, the remove count is not reset by the reset statistics method.</p>
TriggerInfo	String[]	RO	An array of information for each trigger applied to the portion of the partitioned cache managed by the Storage Manager. Each element is a string value that represents a human-readable description of the corresponding <code>MapTrigger</code> .

## Operations

The `StorageManagerMBean` MBean includes a `resetStatistics` operation that resets storage manager statistics. This operation does not reset the `EvictionCount`, `InsertCount` or `RemoveCount` attributes.

## TransactionManagerMBean

The `TransactionManagerMBean` MBean represents a transaction manager and is specific to the transactional framework. A cluster member may have zero or more instances of this managed bean depending on the number of configured transaction caches. The MBean provides global transaction manager statistics by aggregating service-level statistics from all transaction service instances. Each cluster member has an instance of the transaction manager managed bean per service.

The MBean's object name is:

```
type=TransactionManager,service=service name,nodeId=cluster node's id
```

---

**Note:** For certain transaction manager attributes, the count is maintained at the coordinator member for the transaction, even though multiple members may have participated in the transaction. For example, a transaction may include modifications to entries stored on multiple members but the `TotalCommitted` attribute is only incremented on the MBean on the member that coordinated the commit of that transaction.

---

### Attributes

Table A-20 describes the attributes for `TransactionManagerMBean`.

**Table A-20** *TransactionManagerMBean Attributes*

Attribute	Type	Access	Description
<code>CommitTotalMillis</code>	Long	RO	The cumulative time (in milliseconds) spent during the commit phase since the last time statistics were reset.
<code>RefreshTime</code>	Date	RO	The timestamp when this model was last retrieved from a corresponding member. For local servers it is the local time.
<code>TotalActive</code>	Long	RO	The total number of currently active transactions. An active transaction is counted as any transaction that contains at least one modified entry and has yet to be committed or rolled back. Note that the count is maintained at the coordinator member for the transaction, even though multiple members may have participated in the transaction.
<code>TotalCommitted</code>	Long	RO	The total number of transactions that have been committed by the Transaction Manager since the last time the statistics were reset. Note that the count is maintained at the coordinator member for the transaction being committed, even though multiple members may have participated in the transaction.
<code>TotalRecovered</code>	Long	RO	The total number of transactions that have been recovered by the Transaction Manager since the last time the statistics were reset. Note that the count is maintained at the coordinator member for the transaction being recovered, even though multiple members may have participated in the transaction.

**Table A-20 (Cont.) TransactionManagerMBean Attributes**

<b>Attribute</b>	<b>Type</b>	<b>Access</b>	<b>Description</b>
TotalRolledback	Long	RO	The total number of transactions that have been rolled back by the Transaction Manager since the last time the statistics were reset. Note that the count is maintained at the coordinator member for the transaction being rolled back, even though multiple members may have participated in the transaction.
TotalTransactionMillis	Long	RO	The cumulative time (in milliseconds) spent on active transactions.
TimeoutMillis	Long	RO	The transaction timeout value in milliseconds. Note that this value only applies to transactional connections obtained after the value is set. This attribute is currently not supported.
CommitTotalMillis	Long	RO	The cumulative time (in milliseconds) spent during the commit phase since the last time statistics were reset.
RefreshTime	Date	RO	The timestamp when this model was last retrieved from a corresponding member. For local servers it is the local time.
TotalActive	Long	RO	The total number of currently active transactions. An active transaction is counted as any transaction that contains at least one modified entry and has yet to be committed or rolled back. Note that the count is maintained at the coordinator member for the transaction, even though multiple members may have participated in the transaction.
TotalCommitted	Long	RO	The total number of transactions that have been committed by the Transaction Manager since the last time the statistics were reset. Note that the count is maintained at the coordinator member for the transaction being committed, even though multiple members may have participated in the transaction.
TotalRecovered	Long	RO	The total number of transactions that have been recovered by the Transaction Manager since the last time the statistics were reset. Note that the count is maintained at the coordinator member for the transaction being recovered, even though multiple members may have participated in the transaction.
TotalRolledback	Long	RO	The total number of transactions that have been rolled back by the Transaction Manager since the last time the statistics were reset. Note that the count is maintained at the coordinator member for the transaction being rolled back, even though multiple members may have participated in the transaction.

## Operations

The `TransactionManagerMBean` MBean includes a `resetStatistics` operation that resets all transaction manager statistics.

---

---

## Report File Configuration Reference

This appendix provides a detailed reference of the report file deployment descriptor elements and includes a brief overview of the descriptor. See [Appendix C, "Report Group Configuration Reference,"](#) for a detailed reference of the report group deployment descriptor elements.

The following sections are included in this appendix:

- [Report File Deployment Descriptor](#)
- [Element Reference](#)

### Report File Deployment Descriptor

The report file deployment descriptor specifies a report for displaying management information that is based on Coherence MBeans. Coherence includes many predefined report files for most of the MBeans. The default files are located in the `coherence.jar` file in the `/reports` directory. The reports can be modified as required or new reports can be created. See [Chapter 6, "Analyzing Reporter Content,"](#) for additional information about the default reports.

The report file deployment descriptor schema is defined in the `coherence-report-config.xsd` file which is located in the root of the `coherence.jar` library and at the following Web URL:

<http://xmlns.oracle.com/coherence/coherence-report-config/1.0/coherence-report-config.xsd>

The `<report-config>` element is the root element of the deployment descriptor and typically includes the XSD reference, the Coherence namespace reference, and the location of the `coherence-report-config.xsd` file. For example:

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

```
<report-config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.oracle.com/coherence/coherence-report-config"
  xsi:schemaLocation="http://xmlns.oracle.com/coherence/coherence-report-config
  coherence-report-config.xsd">
```

---

---

**Notes:**

- The schema located in the `coherence.jar` library is always used at run time even if the `xsi:schemaLocation` attribute references the Web URL.
  - The `xsi:schemaLocation` attribute can be omitted to disable schema validation.
  - When deploying Coherence into environments where the default character set is EBCDIC rather than ASCII, ensure that the deployment descriptor file is in ASCII format and is deployed into its run-time environment in the binary format.
- 
-

---

## Element Reference

Table B-1 lists all non-terminal report file deployment descriptor elements.

**Table B-1 Report File Deployment Descriptor Non-Terminal Elements**

<b>Element</b>	<b>Used in</b>
column	row
filter	filters
filters	report
params	filter, column
query	report, column
report	report-config
report-config	<i>root element</i>
row	report-config

---

## column

Used in: [row](#)

### Description

The `column` element contains information to generate a report column. The column element supports the use of an `id` attribute to uniquely identify the column. The ID is used within the `column-ref` element.

### Elements

[Table B-2](#) describes the subelements of the `column` element.

**Table B-2** *column Subelements*

Element	Required/Optional	Description
<code>type</code>	Optional	Specifies an <code>XmlColumn</code> implementation. The <code>type</code> element defines the type of the column in the report. Valid values are: <ul style="list-style-type: none"> <li>▪ <code>attribute</code> – (default) A column which contains the data from a MBean attribute.</li> <li>▪ <code>key</code> – A column which contains the value from a MBean key attribute.</li> <li>▪ <code>method</code> – A column which contains the result of an MBean method invocation.</li> <li>▪ <code>function</code> – A column which contains the result of a function or aggregation. See <code>function-name</code> below.</li> <li>▪ <code>global</code> – A column that contains a value not related to any specific MBean. Global column names are <code>{report-time}</code>, <code>{report-count}</code>, and <code>{node-id}</code>.</li> <li>▪ <code>constant</code> – A column which contains a constant string of numeric value.</li> <li>▪ <code>property</code> – A column which contains a value of a Java system property.</li> </ul>
<code>name</code>	Optional	Specifies an attribute or method name on the MBean. For composite data types, the <code>name</code> element can contain a slash (/) delimited name sequence.
<code>header</code>	Optional	Specifies the column header. If this item is omitted, the value of the <code>name</code> element is used instead.
<code>delim</code>	Optional	Specifies a character that is used to separate column or array values. Valid values are: <code>{tab}</code> , <code>{space}</code> , or any non-whitespace character.
<a href="#">query</a>	Optional	Specifies information necessary to construct a JMX query to find all MBeans contributing to the report or column.
<code>hidden</code>	Optional	Specifies whether the column value should be hidden in the report. Valid values are <code>true</code> and <code>false</code> .
<code>column-ref</code>	Optional	Specifies a reference to a column identifier. The reference is used to pass a column value as an argument to a filter or another column.

**Table B-2 (Cont.) column Subelements**

<b>Element</b>	<b>Required/ Optional</b>	<b>Description</b>
function-name	Optional	Specifies the name of the calculation to apply to the associated MBean attribute values. This element is only valid when the <code>type</code> element is <code>function</code> . Valid values are: <ul style="list-style-type: none"> <li>▪ <code>sum</code> – The sum of all retrieved attribute values.</li> <li>▪ <code>avg</code> – The average value for all retrieved attribute values.</li> <li>▪ <code>min</code> – The minimum numeric value for all retrieved attribute values.</li> <li>▪ <code>max</code> – The maximum numeric value for all retrieved attribute values.</li> <li>▪ <code>add</code> – The sum of values for two column references.</li> <li>▪ <code>subtract</code> – The difference between values for two column references.</li> <li>▪ <code>multiply</code> – The product of values for two column references.</li> <li>▪ <code>divide</code> – The ratio between values for two column references.</li> </ul>
<code>params</code>	Optional	Specifies an argument column or a filter reference.
data-type	Optional	Specifies the data type of a constant column. Valid values are <code>double</code> and <code>string</code> .
value	Optional	Specifies the value of a constant column.
group-by	Optional	specifies whether the column should be included in the <code>group by</code> clause of the query. Valid values are <code>true</code> and <code>false</code> . The default values is <code>false</code> .
subquery	Optional	Specifies whether the column is included as part of a subquery. Valid values are <code>true</code> and <code>false</code> . The default values is <code>false</code> .
return-neg	Optional	Specifies if a negative value should be returned to the query. Negative values are considered <code>error</code> or <code>not available</code> codes on Coherence MBeans. By default, these codes are returned as zero to not impact column calculations. Valid values are <code>true</code> and <code>false</code> . The default values is <code>false</code> .

## filter

Used in: [filters](#)

### Description

The `filter` element defines a set of filters that can be used in the report. The filter element supports the use of an `id` attribute to uniquely identify the filter. The ID is used within the `filter-ref` element.

### Elements

[Table B-3](#) describes the subelements of the `filter` element.

**Table B-3** *filter Subelements*

Element	Required/ Optional	Description
<code>type</code>	Optional	Specifies an <code>XmlFilter</code> implementation. The <code>type</code> element defines the type of the filter in the report. Valid values are: <ul style="list-style-type: none"> <li>▪ <code>equals</code> – A filter which compares the result of two or more column references for a value equality.</li> <li>▪ <code>greater</code> – A filter which compares the results of two column references for <code>greater than</code> condition. If any of the values is resolved to <code>null</code>, the evaluation yields <code>false</code>. (This approach is equivalent to the way the <code>NULL</code> values are handled by SQL.)</li> <li>▪ <code>less</code> – A filter which compares the results of two column references for <code>less than</code> condition. If any of the values is resolved to <code>null</code>, the evaluation yields <code>false</code>.</li> <li>▪ <code>not</code> – A filter which returns the logical <code>not</code> of a filter reference.</li> <li>▪ <code>and</code> – A filter which returns the logical <code>and</code> of two filter references.</li> <li>▪ <code>or</code> – A filter which returns the logical <code>or</code> of two filter references.</li> </ul>
<code>params</code>	Optional	Specifies an argument column or a filter reference.

## filters

Used in: [report](#)

### Description

The `filters` element contains a group of filter elements.

### Elements

[Table B-4](#) describes the subelements of the `filters` element.

**Table B-4** *filters Subelements*

Element	Required/ Optional	Description
<a href="#">filter</a>	Optional	Specifies a set of filters that can be used in the report.

## params

Used in: [column](#), [filter](#)

### Description

The `params` element identifies an argument column or a filter reference.

### Elements

[Table B-5](#) describes the subelements of the `params` element.

**Table B-5** *params* Subelements

Element	Required/ Optional	Description
<code>filter-ref</code>	Required	Specifies a reference to a filter id. The reference is used to pass a filter as an argument to a query or another filter.
<code>column-ref</code>	Required	Specifies a reference to a column identifier. The reference is used to pass a column value as an argument to a filter or another column.

## query

Used in: [column](#), [report](#)

### Description

The `query` element contains information necessary to construct a JMX query to find all MBeans contributing to the report or column. Queries are entered within a `<pattern>` element.

### Examples

An example that includes all Node MBeans in the report or column:

```
<query>
  <pattern>Coherence:type=Node, *</pattern>
</query>
```

An example that includes only the Cluster MBean in the report or column:

```
<query>
  <pattern>Coherence:type=Cluster</pattern>
</query>
```

The pattern string allows macro substitutions with run-time values that come from report columns. For example, to provide a cache name into the query pattern, define a system property called `reporter.cacheName` and use the following construct:

```
<report>
  ...
  <query>
    <pattern>Coherence:type=Cache, name={CacheName}, *</pattern>
  </query>
  <row>
    <column id="CacheName">
      <type>property</type>
      <name>reporter.cacheName</name>
    </column>
  </row>
</report>
```

Sometimes a query pattern is known to result in a list of MBean names that have a well known key attribute. That key attribute could be used to retrieve an attribute from a related (joined) MBean. For example, since the `Coherence:type=Service, *` pattern is known to result in MBeans in the `Coherence:type=Service, nodeId=NNN` format, the configuration below is used to print the `MemberName` attribute from a corresponding `NodeMBean` along with the `ServiceName` attribute for the `ServiceMBean`.

```
<report>
  ...
  <query>
    <pattern>Coherence:type=Service, *</pattern>
    <params>
      <column-ref>MemberName</column-ref>
      <column-ref>NodeId</column-ref>
      <column-ref>ServiceName</column-ref>
    </params>
  </query>
  <row>
```

```

    <column id="MemberName">
      <type>attribute</type>
      <name>MemberName</name>
      <query>
        <pattern>Coherence:type=Node,nodeId={NodeId}</pattern>
      </query>
    </column>
    <column id="NodeId">
      <type>key</type>
      <name>nodeId</name>
    </column>
    <column id="ServiceName">
      <type>key</type>
      <name>name</name>
    </column>
  </row>
</report>

```

## Elements

Table B-6 describes the subelements of the query element.

**Table B-6** query Subelements

Element	Required/ Optional	Description
pattern	Required	Specifies a JMX query or Object name to be included in the report.
filter-ref	Optional	Specifies a reference to a filter id. The reference is used to pass a filter as an argument to a query or another filter.
params	Optional	Specifies an argument column or a filter reference.

## report

Used in: [report-config](#)

### Description

The `report` element contains information necessary to generate a JMX based report. Any number of report elements can included; however, it is typical to have a single report definition for each report file.

### Elements

[Table B-7](#) describes the subelements of the `report` element.

**Table B-7** *report Subelements*

Element	Required/ Optional	Description
<code>description</code>	Optional	Specifies a descriptive heading for the report.
<code>file-name</code>	Required	<p>Specifies the file name for the generated report. The file name could be either absolute or relative to a directory specified in the corresponding report group deployment descriptor. If the specified file exists then the new report lines are appended to the file; otherwise, a new report file is created.</p> <p>The file name may contain any of the three special macros:</p> <ul style="list-style-type: none"> <li>▪ <code>{sequence}</code> – The macro is replaced with an execution counter (a sequential number), which is used to generate new files for each report execution.</li> <li>▪ <code>{node}</code> – This macro is replaced with the member id of the report executing member. Use this macro to create different files for reports executed on different cluster members.</li> <li>▪ <code>{date}</code> – This macro is replaced with the current date (YYYYMMDD).</li> </ul> <p><b>Note:</b> a process running the reporter requires read, write, and create access to the location where the output files are created.</p>
<code>delim</code>	Optional	Specifies a character that is used to separate column or array values. Valid values are: <code>{tab}</code> , <code>{space}</code> , or any non-whitespace character.
<code>hide-headers</code>	Optional	Specifies whether the description and the column headers should be included in the report. Valid values are <code>true</code> and <code>false</code> .
<code>filters</code>	Optional	Specifies a group of filters for the report.
<code>query</code>	Required	Specifies information necessary to construct a JMX query to find all MBeans contributing to the report or column.
<code>row</code>	Required	Specifies information to generate a report column.

## report-config

*Root Element*

### Description

The `report-config` element is the root element of the report configuration deployment descriptor and contains the report definition.

### Elements

[Table B-8](#) describes the subelements of the `report-config` element.

**Table B-8** *report-config Subelements*

Element	Required/ Optional	Description
<code>report</code>	required	Specifies the information necessary to generate a JMX based report.

## row

Used in: [report-config](#)

### Description

The `row` element contains a list of columns to be included in the report.

### Elements

[Table B-9](#) describes the subelements of the `row` element.

**Table B-9** *row Subelements*

Element	Required/ Optional	Description
<a href="#">column</a>	Required	Specifies information that the Reporter uses to generate a report column.



---

---

## Report Group Configuration Reference

This appendix provides a detailed reference of the report group deployment descriptor elements and includes a brief overview of the descriptor. See [Appendix B, "Report File Configuration Reference,"](#) for a detailed reference of the report file deployment descriptor elements that are used to create individual report files.

The following sections are included in this appendix:

- [Report Group Configuration Deployment Descriptor](#)
- [Element Reference](#)

### Report Group Configuration Deployment Descriptor

The report group deployment descriptor specifies any number of individual reports that are created when the Coherence JMX Reporter is run. Two predefined descriptors are located in the `coherence.jar` file within the `/reports` directory. The default descriptor is called `report-group.xml` and includes a subset of the predefined reports. The `report-all.xml` descriptor includes all the predefined reports. The predefined report group deployment descriptors can be modified as required or new report group deployment descriptors can be created. The name and location of which report group deployment descriptor to use at run time is configured in the operational deployment descriptor and can be overridden in an operational override file. See [Chapter 4, "Using JMX Reporting,"](#) for detailed information.

The report file deployment descriptor schema is defined in the `coherence-report-group-config.xsd` file which is located in the root of the `coherence.jar` library and at the following Web URL:

<http://xmlns.oracle.com/coherence/coherence-report-group/1.0/coherence-report-group.xsd>

The `<report-group>` element is the root element of the descriptor and typically includes the XSD reference, the Coherence namespace reference, and the location of the `coherence-report-group-config.xsd` file. For example:

```
<?xml version='1.0'?>
<report-group xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.oracle.com/coherence/coherence-report-group-config"
  xsi:schemaLocation="http://xmlns.oracle.com/coherence/
  coherence-report-group-config coherence-report-group-config.xsd">
```

---

---

**Notes:**

- The schema located in the `coherence.jar` library is always used at run time even if the `xsi:schemaLocation` attribute references the Web URL.
  - The `xsi:schemaLocation` attribute can be omitted to disable schema validation.
  - When deploying Coherence into environments where the default character set is EBCDIC rather than ASCII, ensure that the deployment descriptor file is in ASCII format and is deployed into its run-time environment in the binary format.
- 
-

---

## Element Reference

Table C-1 lists all non-terminal report group deployment descriptor elements.

**Table C-1 Report Group Deployment Descriptor None-Terminal Elements**

<b>Element</b>	<b>Used in</b>
<a href="#">init-param</a>	<a href="#">init-params</a>
<a href="#">init-params</a>	<a href="#">report-config</a>
<a href="#">report-config</a>	<a href="#">report-group</a>
<a href="#">report-group</a>	<i>root element</i>
<a href="#">report-list</a>	<a href="#">report-group</a>

## init-param

Used in: [init-params](#)

### Description

The `init-param` element contains an initialization parameter for a report. The parameter consists of either a parameter name or type, and its value.

### Elements

[Table C-2](#) describes the subelements of the `init-param` element.

**Table C-2** *init-param* Subelements

Element	Required/ Optional	Description
<code>param-name</code>	Optional	Specifies the name of the initialization parameter. This element cannot be used with the <code>param-type</code> element.
<code>param-type</code>	Optional	Specifies the Java type of the initialization parameter. Supported types are: <ul style="list-style-type: none"><li>▪ <code>string</code>—indicates that the value is a <code>java.lang.String</code></li><li>▪ <code>long</code>—indicates that the value is a <code>java.lang.Long</code></li><li>▪ <code>double</code>—indicates that the value is a <code>java.lang.Double</code></li></ul> This element cannot be used with the <code>para-name</code> element.
<code>param-value</code>	Required	Specifies the value of the initialization parameter. The value is in a format specific to the type of the parameter.

## init-params

Used in: [report-config](#)

### Description

The `init-params` element contains a list of initialization parameters.

### Elements

[Table C-2](#) describes the subelements of the `init-params` element.

**Table C-3** *init-params Subelements*

Element	Required/ Optional	Description
<a href="#">init-param</a>	Optional	Specifies an initialization parameter for a report.

## report-config

Used in: [report-group](#)

### Description

The `report-config` contains the configuration file name and the initialization parameters for the report.

### Elements

[Table C-4](#) describes the subelements of the `report-config` element.

**Table C-4** *report-config Subelements*

Element	Required/ Optional	Description
<code>location</code>	Required	Specifies a path to a report configuration descriptor that conforms to the <code>coherence-report-config.xsd</code> file. This path could be either a file or a URL.
<code>init-params</code>	Optional	Specifies a list of initialization parameters.

## report-group

Used in: *root element*

### Description

The `report-group` element describes the report list, the frequency, the report parameters, and the output directory for the batch.

### Elements

[Table C-5](#) describes the subelements of the `report-group` element.

**Table C-5** *report-group Subelements*

Element	Required/ Optional	Description
<code>frequency</code>	Required	A string containing the number of seconds, minutes between each execution of the report batch. 10s runs the report ever 10 seconds. 5m runs the report every 5 minutes. Selecting an appropriate frequency is critical. If the frequency is too short, the reporter can generate a large amount of data and consume significant disk space. If the frequency is too long, not enough information is generated. It is recommended that a process for purging and archiving historical information is in place before configuring the reporter.
<code>output-directory</code>	Optional	The directory path to prepend to the output file names from the report configuration files. The username which the member is executing <b>must</b> have read write access to this path.
<code>report-list</code>	Required	Specifies a list of report configurations.

## report-list

Used in: [report-group](#)

### Description

The `report-list` element contains the list of reports to include in the batch.

### Elements

[Table C-6](#) describes the subelements of the `report-list` element.

**Table C-6** *report-list Subelements*

Element	Required/ Optional	Description
<a href="#">report-config</a>	Required	The report configuration location and corresponding initialization parameters.