Oracle® Fusion Middleware

Managing Oracle Coherence

12*c* (12.1.3)

E47884-03

December 2014

Documentation for developers and system administrators that describes how to manage Oracle Coherence using Java Management Extensions (JMX) managed beans (MBeans) and Oracle Coherence reports

ORACLE®

Oracle Fusion Middleware Managing Oracle Coherence, 12*c* (12.1.3)

E47884-03

Primary Author: Joseph Ruzzi

Contents

6 Analyzing Report Contents

A Oracle Coherence MBeans Reference

B Report File Configuration Elements

C Report Group Configuration Elements

Preface

Managing Oracle Coherence describes how to manage Oracle Coherence using Java Management Extensions (JMX) Managed Beans (MBeans) and Oracle Coherence reports.

Audience

This guide is intended for the following audiences:

- Primary Audience – Application developers and administrators who want to learn how to manage an Oracle Coherence Environment
- Secondary Audience – System architects who want to understand the options and architecture for managing Oracle Coherence

The audience must be familiar with Java and JMX to use this guide effectively. 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 in the Oracle Coherence documentation set:

- *Administering HTTP Session Management with Oracle Coherence*Web*
- *Administering Oracle Coherence*
- *Developing Applications with Oracle Coherence*
- *Developing Remote Clients for Oracle Coherence*
- *Integrating Oracle Coherence*

- *Securing Oracle Coherence*
- *Tutorial for Oracle Coherence*
- *Java API Reference for Oracle Coherence*
- *C++ API Reference for Oracle Coherence*
- *.NET API Reference for Oracle Coherence*
- *Release Notes for Oracle Coherence*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

What's New in This Guide

The following topics introduce the new and changed features of Oracle Coherence and other significant changes that are described in this guide, and provides pointers to additional information. This document is the new edition of the formerly titled *Oracle Coherence Management Guid*e.

New and Changed Features for 12*c* (12.1.3)

Oracle Coherence 12*c* (12.1.3) includes the following new and changed features for this document.

- Java VisualVM plug-in for Coherence, which allows a Coherence cluster to be monitored for real-time analysis and troubleshooting. See "Using the Coherence-JVisualVM Plug-In" on page 2-13.
- Run tabular report operations, which allows reports and report groups to be generated as a tabular report. See "Administering Oracle Coherence Reporting Using the Reporter MBean" on page 4-4.

Other Significant Changes in This Document for 12*c* (12.1.3)

For 12*c* (12.1.3), this guide has been updated in several ways. Following are the sections that have been added or changed.

- Revised the default report group configuration file section. See "Switching the Default Report Group Configuration File" on page 4-3.
- Revised the MBean Reference appendix to include all new and updated attributes and operations. See Appendix A, "Oracle Coherence MBeans Reference."
- Revised the cache size report table to include a Service column. See "Understanding the Cache Size Report" on page 6-1

New and Changed Features for 12*c* (12.1.2)

Oracle Coherence 12*c* (12.1.2) includes the following new and changed features for this document.

- Report output system property, which allows the output location for a report group to be overridden at runtime. See "Overriding the Report Group Output Directory" on page 4-4 and "Specifying the Output Directory for Reports" on page 5-13.

- Log cluster state operation, which logs full thread dump and outstanding polls on cluster members that are running with a specified role. See Table A–4, " ClusterMBean Operations".
- Log node state operation, which Logs a full thread dump and outstanding polls for the services on a cluster member. See Table A–6, " ClusterNodeMBean Operations".
- Event Interceptor attribute, which reports statistics for live events. See Table A–17, " ServiceMBean Attributes".

Other Significant Changes in This Document for 12*c* (12.1.2)

For 12*c* (12.1.2), this guide has been updated in several ways. Following are the sections that have been added or changed.

- Revised the management overview documentation. See "Conceptual Overview of Oracle Coherence Management" on page 1-1.
- Revised the management reporting overview documentation. See "Management Reporting in Oracle Coherence" on page 1-3.
- Added a section that introduces the management features that are available if Coherence is deployed as part of a WebLogic Server domain. See "Managing Oracle Coherence with Oracle WebLogic Server" on page 1-4.
- Revised the instructions for programmatic registration of custom MBeans. See "Registering Custom MBeans Programmatically" on page 3-4.
- Revised the MBean Reference appendix to include all new and updated attributes and operations. See Appendix A, "Oracle Coherence MBeans Reference."

1

Introduction to Oracle Coherence Management

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

`http://docs.oracle.com/javase/7/docs/technotes/guides/jmx/index.html`

This chapter includes the following sections:

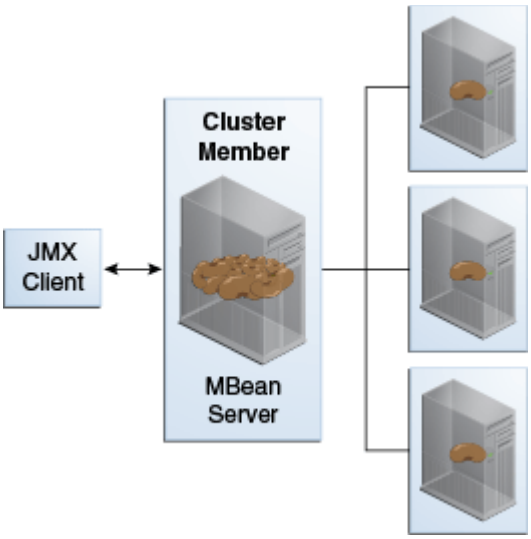
- Conceptual Overview of Oracle Coherence Management
- Management Reporting in Oracle Coherence
- Overview of Management Configuration
- Managing Oracle Coherence with Oracle WebLogic Server
- Managing Oracle Coherence with Oracle Enterprise Manager

Conceptual Overview of Oracle Coherence Management

Oracle 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. It must be explicitly enabled by configuring at least one cluster member to host an MBean server. See Chapter 2, "Using JMX to Manage Oracle Coherence," for more information about enabling JMX management.

Figure 1–1 shows a conceptual view of cluster members and their managed objects being managed through a remote MBean server located on a single cluster member.

Figure 1–1 Oracle Coherence JMX Management



Oracle Coherence MBeans

Oracle Coherence managed objects are registered to an MBean server using the `com.tangosol.net.management.Registry` interface. The interface is specific to managing 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 Oracle Coherence MBeans to be registered in an MBean server that is either collocated with or remote to the managed object.

Appendix A, "Oracle Coherence MBeans Reference," provides a list of all MBeans and describes each of the attributes and operations that are exposed for the managed resources. Some managed resources have only a single instance for each cluster member. However, some managed resources (such as the `CacheMBean` MBean) have multiple MBean instances for each cluster member. In addition, an MBean is registered only if at least one managed resource is operational. For the `CacheMBean` MBean, a cache must be started before the MBean is registered.

Custom MBeans

Custom MBeans can be managed and monitored 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. Registering custom MBeans allows application 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 about registering custom MBeans.

MBean Consoles

Any MBean-capable console can interact with Oracle Coherence MBeans. The Java VisualVM console and the Java Monitoring & Management Console (JConsole) that are distributed with the JDK are common choices. Support is also provided for the JMX HTML Adapter Web Application that is included as part of the JMX reference implementation. See "Accessing Oracle Coherence MBeans" on page 2-8 for more information about using these consoles to interact with MBeans.

Java VisualVM Coherence Plugin

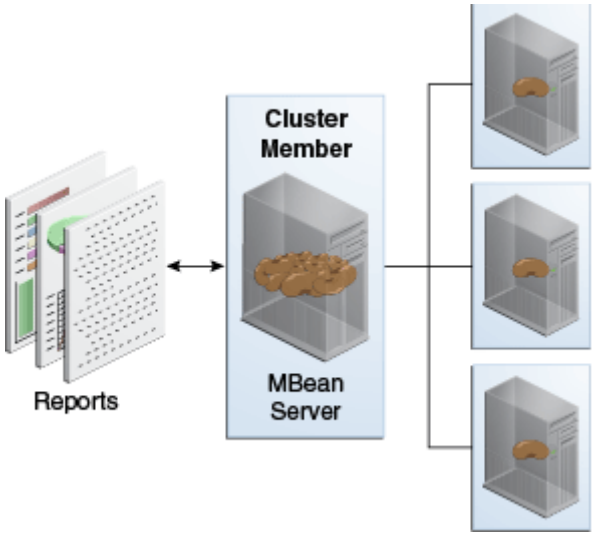
The Coherence-JVisualVM plug-in aggregates Coherence MBean data and shows a concise operational view of a Coherence cluster within the Java VisualVM console. Management information is presented over time, which allows real-time analysis and troubleshooting. See "Using the Coherence-JVisualVM Plug-In" on page 2-13 for more information about using the Coherence-JVisuakVM plug-in.

Management Reporting in Oracle Coherence

Oracle Coherence provides management reports that display management information over time. The reports consist of text files that are constructed from data that is obtained from Oracle Coherence MBeans. The reports update automatically at a configured time interval and provide a historical context that is not possible simply by monitoring the MBeans. The reports identify trends that are valuable for troubleshooting and planning.

Figure 1–2 shows a conceptual view of management reports that are generated on a single cluster member based on the management information of all cluster members.

Figure 1–2 Oracle Coherence Management Reports



Many predefined reports are provided. 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 Oracle Coherence Reporting," for details on enabling reporting. Also, see Chapter 6, "Analyzing Report Contents," for detailed information about the predefined reports.

Overview of Management Configuration

Management is configured using several configuration files. See *Developing Applications with Oracle Coherence* for detailed information about configuration. The files include:

- Operational override file – The `tangosol-coherence-override.xml` file is used to override the default operational settings that are contained in the operational deployment descriptor (`tangosol-coherence.xml`) that is located in the Coherence JAR file. The descriptor specifies the operational and run-time settings

for a cluster and includes management services. The management settings are defined within the `<management-config>` node. See *Developing Applications with Oracle Coherence* for a reference of all management settings.

- MBean configuration override File – The `custom-mbeans.xml` file is the default MBean configuration override file. It declaratively defines 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 report configuration file defines a report and 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` library file 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 Elements," for details about the report file configuration elements.
- Report group configuration file – A report group configuration file lists 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` library file. Additional report group configuration files are provided. In addition, custom report group files can be created as required. See Appendix C, "Report Group Configuration Elements," for details about the report group configuration elements.
- Management invocation service file – The `management-config.xml` file configures the management invocation service instance that is used by the Oracle Coherence JMX management framework. The file is located in the root of the `coherence.jar` library file and can be overridden by placing another `management-config.xml` file in the classpath before the `coherence.jar` library 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.

Managing Oracle Coherence with Oracle WebLogic Server

Oracle WebLogic Server includes an Oracle Coherence integration which standardizes how Oracle Coherence is managed within an Oracle WebLogic Server domain. The integration makes Oracle Coherence a subsystem of Oracle WebLogic Server and allows Oracle Coherence environments to be managed using Oracle WebLogic Server tools. The key management tasks include:

- Setup and configure Oracle Coherence clusters
- Add and remove Oracle Coherence cluster members
- Configure Oracle Coherence cluster member properties
- Start and stop Oracle Coherence cluster members
- Deploy Oracle Coherence applications as Grid ARchive (GAR) modules
- Start and stop Oracle Coherence applications
- Secure Oracle Coherence resources

For details about using Oracle Coherence with WebLogic server, see *Administering Oracle Coherence*.

Managing Oracle Coherence with Oracle Enterprise Manager

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

- Manage complexity by modeling the entire cluster as a single target
- Provide real-time and historical performance monitoring for caches and nodes for faster diagnostics and resolution times
- Monitor caches in the context of applications to analyze dependency
- Provide 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 Cloud Control, which includes the Management Pack for Oracle Coherence, can be downloaded from Oracle Technology Network (OTN):

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

For detailed instructions about configuring and using the management pack, see *Oracle Enterprise Manager Getting Started Guide for Oracle Coherence* in the Oracle Enterprise Manager Documentation Library.

2

Using JMX to Manage Oracle Coherence

This chapter provides instructions for enabling and configuring JMX management on cluster members and describes how to access Oracle Coherence MBeans using tools such as Java VisualVM and JConsole.

This chapter includes the following sections:

- Configuring JMX Management
- Accessing Oracle Coherence MBeans
- Using the Coherence-JVisualVM Plug-In

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 *Developing Applications with Oracle Coherence* for further description of the `<management-config>` element.

This section includes the following topics:

- 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
- Using an Existing MBean Server

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 practice because it avoids loading JMX software into every single cluster member while still providing fault tolerance if a single JMX member fails.

In smaller clusters, a common practice is to have dedicated JMX JVMs on two existing cluster members to ensure fault tolerance. In very large clusters, it is often practical to

have two computers that are dedicate solely for JMX; however, this is not always necessary.

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 manages only 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 also enables remote JMX management. For example:

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

Enabling Local JMX Management on a Cluster Member

Local JMX management constrains an MBean server to manage only the MBeans that are local (within the same JVM) to the cluster member. Accessing the MBean server on the cluster member shows only local management information. However, the member's MBeans can still be managed by a cluster 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 also enables local JMX management. 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. For example:

```
cache-server -jmx
```

The argument automatically sets the management system properties, which can be changed as required within the script. The default settings are the following:

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

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, set the <allow-remote-management> element 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 also disables remote management. 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. You must also set the <allow-remote-management> to false. 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 Oracle 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.tagosol.net.management.ObjectNameExcludeFilter`) excludes MBeans from being registered based on their JMX object name using standard regex patterns. For example, the pattern `.*type=Service,name=Management,.*` excludes MBeans with `type=Service` and `name=Management`. As configured out of the 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=ClassLoading,.*
              .*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 also filters MBeans. 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. The following example configures each of these settings, which are described in detail after the 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">300ms
      </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)

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

The tangosol.coherence.management.refresh.expiry system property also sets the expiry. For example:

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

Setting the Management Refresh Policy

The <refresh-policy> element specifies how to refresh remote management information. Each policy uses a different refresh algorithm that improves latency for a given MBean usage pattern. Table 2–1 describes each policy.

Table 2–1 Refresh Policies

Setting	Description
`refresh-ahead` (default)	Refreshes MBeans 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 or programmatic pattern.
`refresh-behind`	Refreshes each MBean after the data is accessed. This method ensures optimal response time. However, the information returned is offset by the last refresh time.
`refresh-expired`	Refreshes each MBean from the remote member when the MBean 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 also sets the policy. For example:

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

Setting the Management Refresh Timeout

The `<refresh-timeout>` element specifies 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)

A unit of milliseconds is assumed if the value does not contain a unit. The default value is `250ms`.

The `tangosol.coherence.management.refresh.timeout` system property also sets the timeout. For example:

```
-Dtangosol.coherence.management.refresh.timeout=300ms
```

Using an Existing MBean Server

The Oracle Coherence management framework allows the use of an existing MBean server to expose Oracle Coherence MBeans. The MBean server must be located in the same JVM process as the cluster member, and the cluster member must have JMX management enabled.

To use an existing MBean server, enter the MBean server's domain name using the `<default-domain-name>` element. MBean registration fails if an MBean server with the

provided domain name is not found. The following example specifies an existing MBean server with the `MyDomainName` domain name:

```
<?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>
      <default-domain-name>MyDomainName</default-domain-name>
   </management-config>
</coherence>
```

Implement the `com.tangosol.net.management.MBeanServerFinder` interface to customize how the Oracle Coherence management framework locates an MBean server. Add the fully qualified name of the implementation class within the `<server-factory>` element using the `<class-name>` element and include any initialization parameters using the `<init-params>` element. Use the `<default-domain-name>` element to specify an existing MBean server domain name to use when instantiating the class.

The following example specifies the `MyMBeanServerFinder` implementation class and passes the `MyDomainName` domain name to the implementation class.

```
<?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>
      <default-domain-name>MyDomainName</default-domain-name>
      <server-factory>
         <class-name
            system-property="tangosol.coherence.management.serverfactory">
               package.MyMBeanServerFinder</class-name>
         <init-params>
            <init-param>
               <param-type>int</param-type>
               <param-value>0</param-value>
            </init-param>
         </init-params>
     </server-factory>
   </management-config>
</coherence>
```

The `tangosol.coherence.management.serverfactory` system property also specifies the implementation class. For example:

```
-Dtangosol.coherence.management.serverfactory=package.MyMBeanServerFinder
```

Accessing Oracle Coherence MBeans

JMX management must be enabled on a cluster member before you can access Oracle Coherence MBeans; see "Configuring JMX Management" on page 2-1. The examples in this section use the `-jmx` argument of the Oracle Coherence startup script to enable JMX management.

This section includes the following topics:

Accessing MBeans Locally Using Java VisualVM

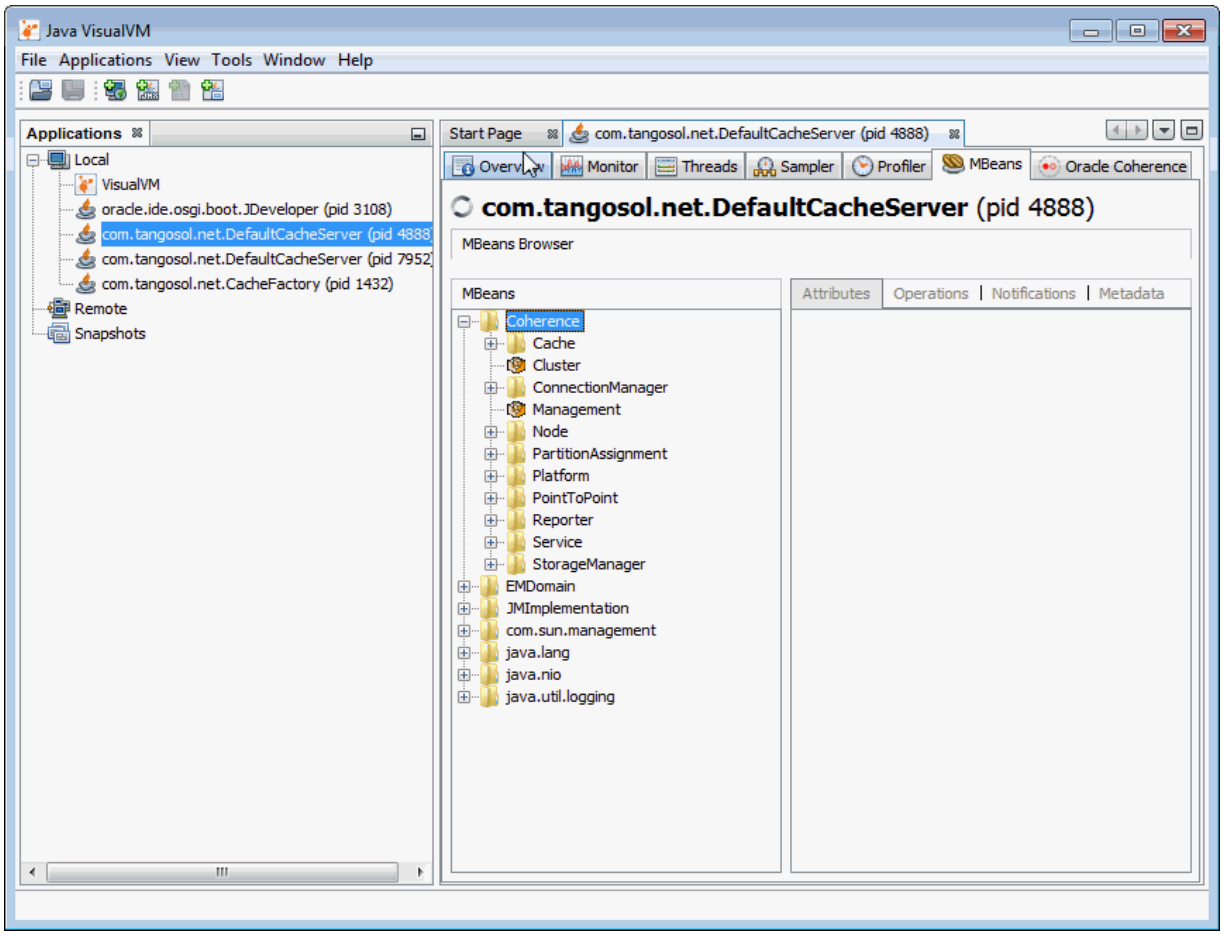
Java VisualVM is a management utility that is included with the JDK (*JDK_HOME*`\bin\jvisualvm`) and provides the ability to view and interact with MBeans. MBean functionality is provided as a plug-in to Java VisualVM. The plug-in must be installed from the Java VisualVM Plugins Center. From the tool, use the **Tools** -> **Plugins** option to install the VisualVM-MBeans plug-in.

To locally 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** and double-click the cluster member's process. The process information appears 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 MBeans.



Accessing MBeans Locally Using the JConsole Utility

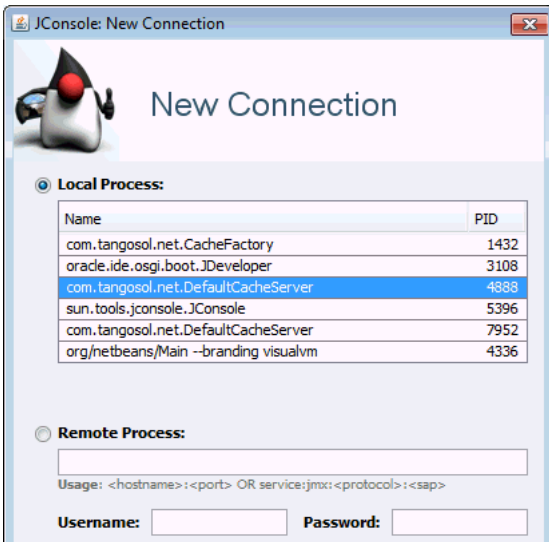
JConsole is a management utility that is included with the JDK (*JDK_HOME*\bin\jconsole) and provides the ability to view and interact with MBeans.

To locally access MBeans using the JConsole utility:

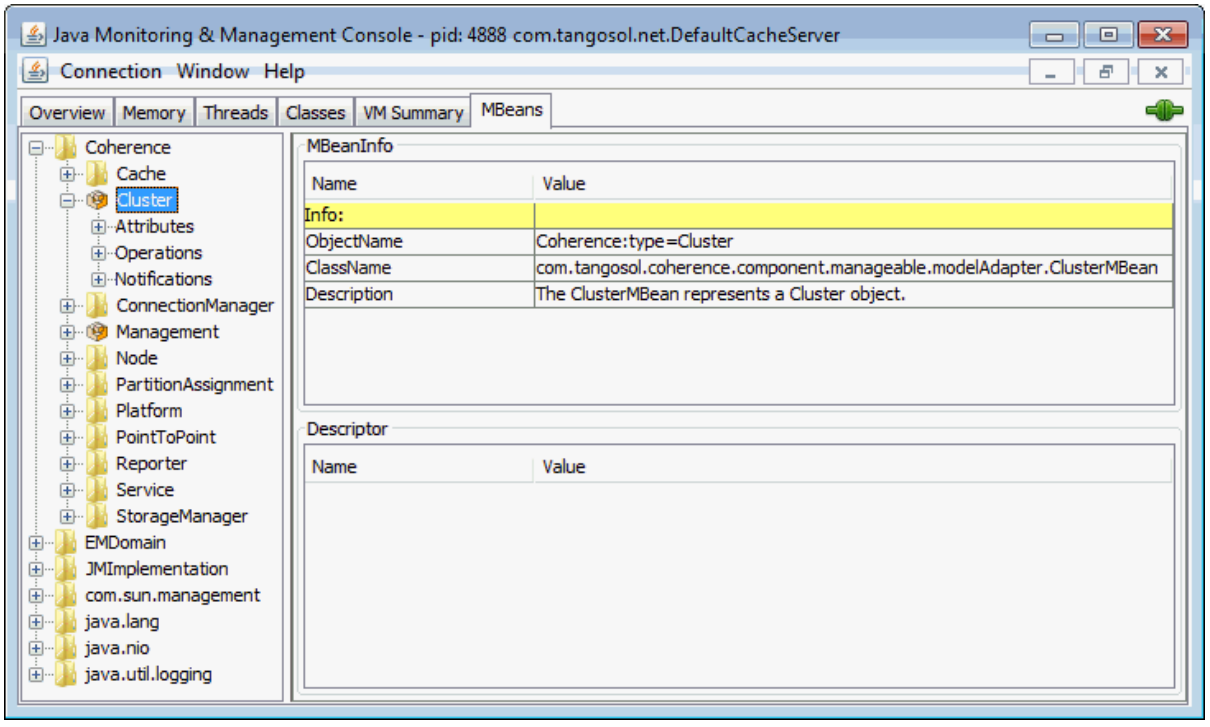
1. Start a cache server using the *COHERENCE_HOME*\bin\cache-server script and specify the `-jmx` argument to enable management on the member. For example:

   ```
   COHERENCE_HOME\bin\cache-server -jmx
   ```

2. Start *JDK_HOME*\bin\jconsole. The Java Monitoring & Management Console window appears and the JConsole: New Connection dialog box is displayed.
3. From the JConsole: New Connection dialog box, select the **Local Process:** option and then 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 MBeans.



Allowing Remote Access to Oracle Coherence MBeans

Remote JMX management allows JMX clients that are located on remote computers to access Oracle Coherence MBeans. System properties configure remote access and are often set when starting a JVM that hosts an MBean server: that is, the system properties are set when starting cluster members that host an MBean server for the cluster. The following instructions provide only the minimum configuration that is required. See the JMX documentation for complete details on configuring JMX management:

http://download.oracle.com/javase/7/docs/technotes/guides/management/agent.html

To allow remote access to Oracle Coherence MBeans without security enabled, set the following system properties when you start a cluster member. For ease of use, set the system properties in the Oracle Coherence startup scripts so that they are automatically set when you use the `-jmx` argument.

```
-Dcom.sun.management.jmxremote.port=port
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
```

Set the `com.sun.management.jmxremote.port` value to any available port on the cluster member. Specify the host and port in the JMX client (for example, JConsole) when you connect to the remote MBean server.

In some cases (such as virtualized environments or when using network address translation), the host name of the computer is incorrectly identified or the default value (127.0.0.1) is returned in the RMI stubs that are sent to the JMX client. As a result, the JMX client fails to connect to the MBean server. To ensure that the correct IP address is included in the RMI stubs, explicitly set the IP address of the host by using the `java.rmi.server.hostname` system property. For example:

```
-Djava.rmi.server.hostname=IP_Address
```

See the RMI documentation for complete details on RMI properties:

http://download.oracle.com/javase/7/docs/technotes/guides/rmi/javarmiproperties.html

Accessing MBeans Using the HTML Adapter Application

The HTML Adapter Web Application allows remote access to Oracle Coherence MBeans on cache clients and uses the HTML 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 is available from the following page:

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

To access 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:

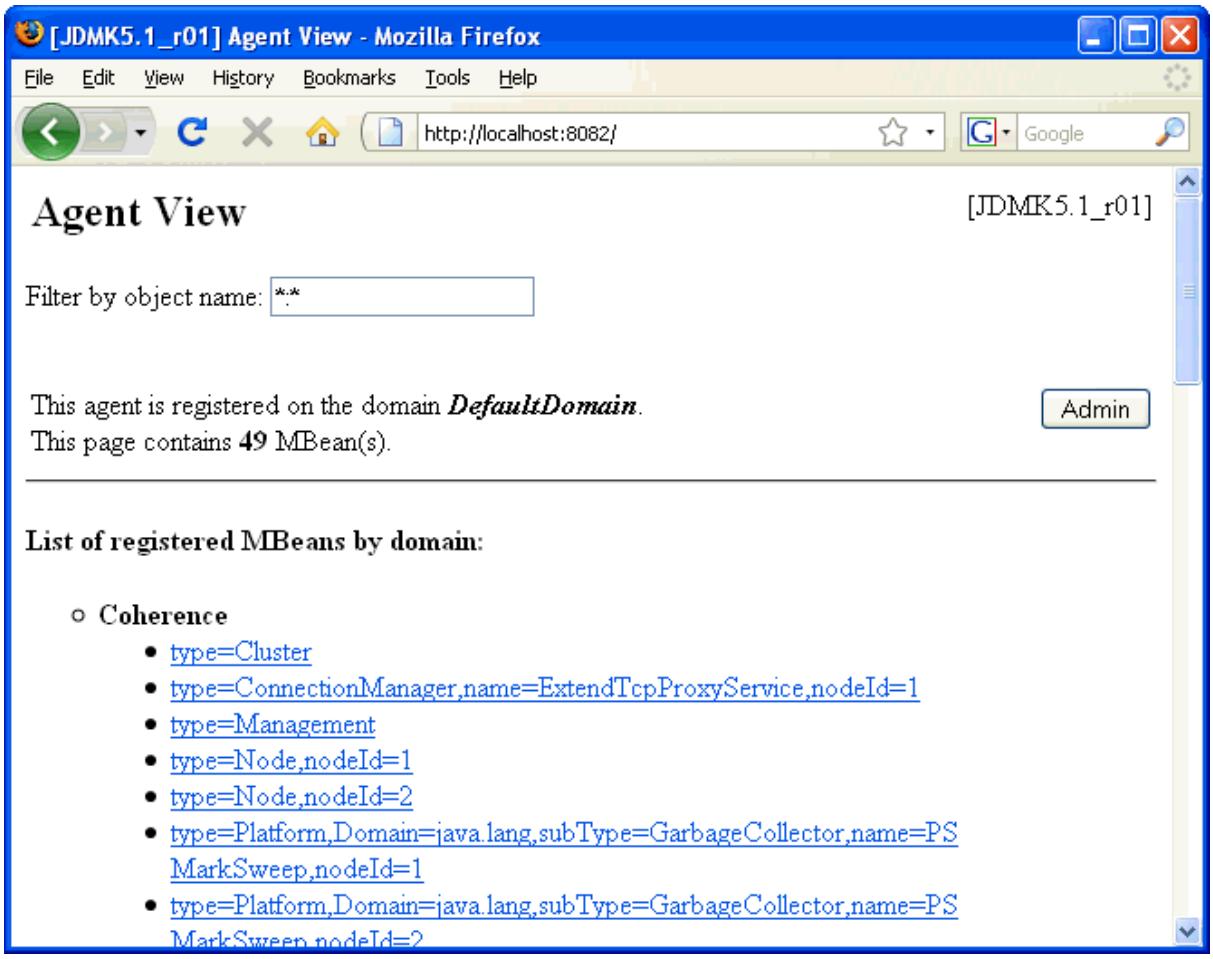
   ```
   -cp "jmxri-1.2.1.jar;jmxtools-1.2.1.jar;%coherence_home%\lib\coherence.jar"
   ```

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 by using the host name and port `8082` for the address as shown in the following screen:



Setting Up the Oracle Coherence MBean Connector

Oracle Coherence ships with a program to start a cluster member as a dedicated MBean server host. This program provides remote access to Oracle Coherence MBeans by using the JMX Remote API using RMI or the HTTP server provided by the JMX reference implementation. Configure the RMI and HTTP ports setting, if required, to allow access through a firewall. The server is started using the following command:

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

Table 2–2 describes optional properties for JMX RMI configuration.

Table 2–2 Optional Properties That Can Be Used for JMX RMI Configuration

Property	Description
`tangosol.coherence.management.remote.host`	The host to which the JMX server binds. The default value is `localhost`. On Linux, the value may have to be changed to the host name or IP address.

Table 2–2 (Cont.) Optional Properties That Can Be Used for JMX RMI Configuration

Property	Description
`tangosol.coherence.management.remote.registryport`	The port used for the JMX RMI registry. The default value is `9000`.
`tangosol.coherence.management.remote.connectionport`	The port used for the JMX RMI connection. The default value is `3000`.

Table 2–3 describes optional properties available for HTTP configuration.

Table 2–3 Optional Properties That Can Be Used for Http Configuration

Property	Description
`tangosol.coherence.management.remote.httpport`	The port used for the HTTP connection. The default value is `8888`.

To connect to the MBean server using JConsole with the default settings, use the following command:

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

To connect to the MBean server using HTTP with the default settings, use the following URL:

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

> **Note:** See the JMX Agent documentation to set up secure access using authentication and Secure Socket Layer (SSL):
>
> `http://download.oracle.com/javase/7/docs/technotes/guides/management/agent.html`

Using the Coherence-JVisualVM Plug-In

The Coherence-JVisualVM plug-in allows a Coherence cluster to be monitored using the Java VisualVM management utility that is included with the JDK (*JDK_HOME*`\bin\jvisualvm`). The plug-in aggregates Coherence MBean data and shows a concise operational view of a single Coherence cluster. Some management information is presented over time, which allows real-time analysis and troubleshooting.

> **Note:** The Coherence-JVisualVM plug-in is a design-time tool that is intended to monitor a single Coherence clusters during development and testing. Enterprise-level management products, such as Oracle Enterprise Manager, should be used for monitoring, managing, and alerting in production environments.

This section includes the following topics:

- Install the Coherence-JVisualVM Plug-In
- Monitor a Coherence Cluster Using the Coherence-JVisualVM Plug-in
- Monitoring Managed Coherence Servers with the Coherence-JVisualVM Plug-in
- Changing the Coherence-JVisualVM Plug-in Refresh Interval

Install the Coherence-JVisualVM Plug-In

The Coherence-JVisualVM plug-in file (`coherence-jvisualvm.nbm`) is included in the *`COHERENCE_HOME`*`\plugins\jvisualvm` directory. Use the Java VisualVM Plugins tool to install the plug-in. The Coherence-JVisualVM plug-in is supported on JDK 1.7 update 40 and above.

To install the Coherence-JVisualVM plug-in:

1. Run the *`JDK_HOME`*`\bin\jvisualvm` file. The Java VisualVM window displays.
2. From the Tools menu, select **Plugins**. The Plugins screen displays.
3. Select the Downloaded Tab and click **Add Plugins...**. Use the Add Plugins dialog box to find the *`COHERENCE_HOME`*`\plugins\jvisualvm\coherence-jvisualvm.nbm` file.
4. Click **Open**. The Coherence-JVisualVM plug-in is imported into the tool and displays on the list of downloaded plugins.
5. From the list of plugins, select **Coherence-JVisualVM** and click **Install**. The Plugin Installer displays. Follow the online instructions to install the plugin.
6. Select the Installed tab to verify that the plug-in is installed and activated.
7. Click **Close** to exit the Plugins screen.

Monitor a Coherence Cluster Using the Coherence-JVisualVM Plug-in

A Coherence cluster is monitored in the Java VisualVM tool by connecting to a Coherence cluster member JVM process and using the Oracle Coherence tab. Coherence JMX management must be enabled on a cluster member before you can access management data; see "Configuring JMX Management" on page 2-1.

To monitor a Coherence cluster using the Coherence-JVisualVM plug-in:

1. Start a Coherence cluster.
2. From the Java VisualVM Applications tree, click to expand **Local** and double-click a Coherence cluster member process that hosts an MBean server (JMX management-enabled member). The process information appears in a process tab on the right side of the Java VisualVM window.
3. From the selected process tab, click **Oracle Coherence** to monitor the Coherence cluster using the Coherence-JVisualVM plug-in as shown below.

Note: Enlarge the Java VisualVM window to at least 1280x800 in order to view all data.

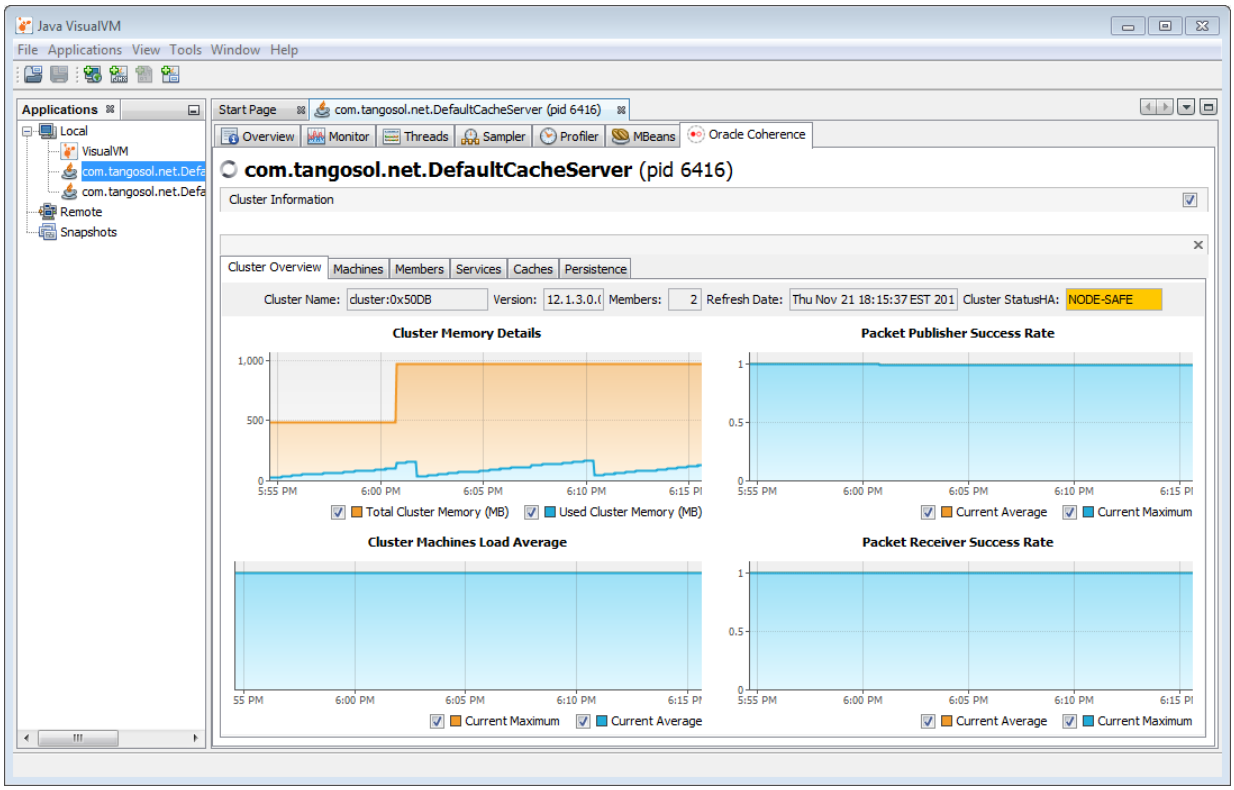


Table 2–4 provides a high-level description of the available tabs.

Table 2–4 Coherence-JVisualVM Plug-In Tab Descriptions

Tab	Description
Overview	Displays high-level information about the cluster including the cluster name, version, member count, and high availability status. Summary graphs are aggregated from the other tabs and show the total cluster memory available and used, load averages for cluster machines, and packet publisher and receiver success rates. Use the Overview tab to get a quick snapshot of how the cluster is operating and to identify potential performance and high availability issues.
Machines	Displays a list of the physical servers that are in the cluster and includes a memory profile for each server. A summary graph shows the average load on these servers. Use the Machine tab to monitor the load on each server and to identify any servers that are in danger of using all of their available memory.
Members	Displays a list of all cluster members, their address and role in the cluster, and includes both publisher and receiver success rates, heap memory usage, and send queue sizes. A summary graph shows the total cluster memory available and used. Use the Members tab to monitor cluster membership and to identify potential memory issues for both a member and the cluster as a whole.
Services	Displays information about cluster services including partition counts and statusHA values. Select a service to see detailed thread information for each node of the service as well as graphs of that information after the next data refresh. Raw data for service thread performance is included to the left of the graph. Use the Services tab to monitor services in the cluster and to identify potential performance issues with service threads.

Table 2–4 (Cont.) Coherence-JVisualVM Plug-In Tab Descriptions

Tab	Description
Caches	Displays information about cluster caches including size and memory usage. To get the correct information to be displayed for memory usage, the backing map scheme for a cache must be configured to use the binary unit-calculator, which is the default unit calculator if no unit-calculator is defined. Select a cache to see detailed information on the next data refresh. Use the Caches tab to get a detailed view of cache statistics and to identify potential performance issues with cache queries.
Proxy Servers	Displays information about proxy servers in the cluster including the number of connections across each proxy server and total messages that were sent and received. A summary graph shows the total number of proxy server connections. Use the Proxy Servers tab to monitor Coherence*Extend clients that are connecting to a proxy and to identify potential performance issues on the proxy. **Note:** This tab is only available if a proxy server is configured in the cluster. Also, you may need to reconnect to a Coherence JVM process if a proxy server was started after other Coherence JVM processes.
Coherence*Web	Displays information about HTTP session storage in the cluster including the number of applications deployed, the number of HTTP sessions being stored, as well as other information regarding session reaping. Use the Coherence*Web tab to monitor HTTP session storage and to identify potential performance issues with session storage. **Note:** This tab is only available if a Coherence cluster is being used for HTTP session storage.
JCache	Displays configuration information and performance statistic information for JCache caches. Use the JCache tab to troubleshoot configuration and performance issues. **Note:** This tab is only available if a Coherence cluster is being used store JCache caches.

Monitoring Managed Coherence Servers with the Coherence-JVisualVM Plug-in

The Coherence-JVisualVM plug-in can be used to monitor Coherence clusters that are managed in a WebLogic Server domain. One managed Coherence server must be configured as the management proxy and all manager Coherence servers must be restarted. The WebLogic Server domain runtime MBean server collects JMX information from the management proxy. For details about configuring a management proxy, see *Administering Clusters for Oracle WebLogic Server*.

To connect to the domain runtime MBean server using Java VisualVM:

1. Source your WebLogic Server environment by executing the `wlserver\server\bin\setWLSEnv` script.
2. Start Java VisualVM and include the following WebLogic Server libraries in the classpath. In addition, include the following options to ensure that MBean checking is disabled.

```
jvisualvm --cp WLS_HOME\server\lib\wljmxclient.jar;
  WLS_HOME\server\lib\weblogic.jar -J-Djmx.remote.protocol.provider.pkgs=
  weblogic.management.remote
  -J-Dcom.oracle.coherence.jvisualvm.disable.mbean.check=true
```

3. From the Java VisualVM Applications tree, right-click **Local** and select **Add JMX Connection**. The Add JMX Connection dialog box displays.

4. Enter the following connection URL and replace *hostname* and *port* with your WebLogic Server host name and port:

   ```
   service:jmx:iiop://hostname:port/jndi/weblogic.management.mbeanservers.
   domainruntime
   ```

5. Click **Use security credentials** and enter the WebLogic Server username and password.
6. Click **OK**.
7. Double-click the new connection.
8. From the selected process tab, click the **Oracle Coherence** tab to monitor the Coherence cluster.

Changing the Coherence-JVisualVM Plug-in Refresh Interval

The Coherence-JVisualVM plug-in is set to refresh data from the Coherence MBean server by default every 30 seconds. To change the refresh setting, use the following system property when starting Java VisualVM:

```
jvisualvm -J-Dcom.oracle.coherence.jvisualvm.refreshtime=10
```

If refreshing management data is adversely affecting performance (especially on large clusters), then consider leaving the default value for the plug-in refresh interval and changing the default Coherence management refresh expiry time of 1 second to a higher value. The value can be specified within an operational override file or by using a system property. For details on changing the refresh expiry time, see "Configuring Management Refresh" on page 2-5.

3

Registering Custom MBeans

This chapter provides instructions for registering custom MBeans to the Oracle Coherence management framework to integrate management of both custom MBeans and Oracle Coherence MBeans.

This chapter includes the following sections:

Overview of Registering Custom MBeans

Oracle 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 an Oracle Coherence MBean server either declaratively in an XML file or programmatically.

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

Note: A dead lock might occur when constructors of global custom MBeans access Coherence distributed services. Constructors of global custom mbeans are not allowed to access Coherence Distributed services.

Registering Custom MBeans Declaratively

Custom MBeans are declaratively registered within an MBean configuration override file named `custom-mbeans.xml`. At run time, the first instance of `custom-mbeans.xml` that is found in the classpath is used. See *Developing Applications with Oracle Coherence* for detailed information about MBean configuration. To register MBeans, use either an MBean's class name, an MBean factory, or run 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 cluster member, including the JMX management-enabled member.

This section includes the following topics:

Creating an MBean Configuration File

The `custom-mbeans.xml` file overrides the `<mbeans>` element of the operational deployment descriptor. Therefore, the root element must be the `<mbeans>` element. See the *Developing Applications with Oracle Coherence* for a complete reference of 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. Ensure that the location of the custom MBean configuration override file is 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 registers and instantiates 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>
```

The preceding configuration registers the MBean and allows remote management. Therefore, the MBean's name automatically includes the cluster member's node ID and domain namespace as part of the object name. For example:

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

See "Registering Custom MBeans Programmatically" on page 3-4 for changing the default registration behavior.

Registering MBeans from an MBean Factory

The <mbean-factory> and <mbean-accessor> elements 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>
```

The preceding configuration registers the MBean and allows remote management. Therefore, the MBean's name automatically includes the cluster member's node ID and domain namespace as part of the object name. For example:

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

See "Registering Custom MBeans Programmatically" on page 3-4 for changing the default registration behavior.

Registering MBeans Using a Query

The <mbean-query> element queries 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 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 instead of the default custom-mbeans.xml override file. For example:

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

Ensure that the classpath includes the location of the file, or 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 you specify the location of an MBean configuration override file.

Restricting MBeans to the Local MBean Server

Custom MBeans are visible to any cluster member that enables JMX management. To register MBeans to only the local MBean server and restrict the MBeans from being visible to remote cluster members, set the `<local-only>` element to `true`. 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 registers custom MBeans programmatically. The following example registers a standard MBean using the `register` method. See *Java API Reference for Oracle Coherence* 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);
```

The preceding configuration registers the MBean and allows remote management. Therefore, the MBean's name automatically includes the cluster member's node ID and domain namespace as part of the object name. For example:

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

Using a Custom Domain Namespace

The default namespace behavior ensures unique object names on the centralized MBean server when multiple clusters run within the same JVM. That is, a different domain namespace is automatically assigned (`Coherence@1:`, and so on) to ensure that MBeans across clusters do not use the same object name. When using a custom domain namespace, applications must ensure unique object names.

To use a custom domain namespace, explicitly add the namespace to the definition. For example:

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

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

To perform JMX operations on the custom MBeans, use the object name as returned by `Registry.ensureGlobalName()` API.

Using Static MBean Names

The `ensureGlobalName` method adds the `,nodeId=...` portion to the end of the MBean's `ObjectName` to ensure unique names on the centralized MBean server. If an application requires static MBean names, use an MBean query to add MBeans from a local MBean server to the management framework. For example, create MBeans on the managed member with static names and then the `,nodeId=...` portion is added to the name of the MBean during MBean aggregation 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` method before joining the cluster. See the `MBeanServer` interface reference for information about using these methods:

   `http://docs.oracle.com/javase/7/docs/api/javax/management/MBeanServer.html`

2. Use the `MBeanHelper.findMBeanServer()` method to obtain the same MBean server that Oracle Coherence is using.

3. Configure the `custom-mbeans.xml` file to query the MBean server for the MBeans. See "Registering MBeans Using a Query" on page 3-3.

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

`http://download.oracle.com/javase/7/docs/technotes/guides/management/agent.html`

4

Using Oracle Coherence Reporting

This chapter provides an overview of Oracle Coherence reporting and describes how to enable and configure reporting in a cluster.

This chapter includes the following sections:

Overview of Oracle Coherence Reporting

Oracle Coherence reports show key management information over time. The reports often identify trends that are valuable for troubleshooting and planning. Reporting is disabled by default and must be explicitly enabled in an operational override file or by using system properties.

Viewing Reports

The default reporting behavior creates ten reports each hour and refreshes them every minute. The reports are saved to the directory from which the cluster member starts. Each file includes a prefix with a report timestamp in a *YYYYMMDDHH* format. The format allows for easy location and purging of unwanted information. Table 4–1 summarizes the reports. See Chapter 6, "Analyzing Report Contents," for complete report details.

Table 4–1 Default Oracle Coherence Reports

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

Table 4–1 (Cont.) Default Oracle Coherence 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 cluster
YYYYMMDDHH-nodes.txt	Contains a list of members of the cluster
YYYYMMDDHH-report-proxy.txt	Contains utilization information about each proxy server in the cluster
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 Reporting Configuration

A report configuration file defines a single report. For example, the `report-node.xml` configuration file creates the *YYYYMMDDHH*`-nodes.txt` report. The report configuration files use Oracle Coherence MBeans to gather the data that is used in the report. The `coherence.jar/reports` directory includes the predefined report configuration files. See Appendix B, "Report File Configuration Elements," for details about the report file configuration elements.

Report group configuration files control which reports to generate, how often to refresh the reports, the directory in which to save the reports, and any parameters for a report. The `coherence.jar/reports` directory includes three predefined report group configuration files: `report-group.xml`, `report-web.xml`, and `report-all.xml`. See Appendix C, "Report Group Configuration Elements," for details about the report group configuration elements.

Customizing Reports

You can customize reporting configuration files or create new configuration files as required. A report can include data from any MBeans (including custom MBeans). See Chapter 5, "Creating Custom Reports," for detailed instructions.

Enabling Oracle Coherence Reporting on a Cluster Member

Reporting is disabled by default and must be explicitly enabled. Ensure that JMX management is enabled on the cluster member before you enable reporting. 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 saved in the directory where the cluster member starts.

Note: Decide on a plan for archiving reports before you enable reporting.

To enable reporting, edit the operational override file and within the `<reporter>` element, add an `<autostart>` 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 also enables reporting. For example:

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

Switching the Default Report Group Configuration File

Report group configuration files define which reports are generated. Three predefined report group configuration files are included in the `coherence.jar/reports` directory:

- `report-group.xml` – Configures a set of core reports to generate. This is the report group configuration file for default configuration.
- `report-web-group.xml` – Configures a set of Oracle Coherence*Web reports. For reports specific to Oracle Coherence*Web, see *Administering HTTP Session Management with Oracle Coherence*Web*.
- `report-all.xml` – Configures all the predefined reports to generate

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

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

Overriding the Report Group Output Directory

The predefined report groups are configured by default to save reports to the directory from which the cluster member starts. The output directory can be overridden using the `tangosol.coherence.reporter.output.directory` system property. For example:

```
-Dtangosol.coherence.reporter.output.directory=/mydirectory
```

The path can be absolute or relative to the directory where the cluster member starts (`./`). The user name that the member is executing must have read/write access to the path.

Changing the Report Timestamp Format

Report timestamps display the local time zone and a default time and date format (`EEE MMM dd HH:mm:ss zzz yyyy`). Change the time zone and time and date format as required. Enter the time zone using either a zone ID (for example, US/Eastern) or a custom ID (for example, GMT-05:00). The time and date format follow the patterns defined by the `java.text.SimpleDateFormat` class.

To change the report timestamp format, edit the operational override file and within the `<reporter>` element, add the `<timezone>` and `<timeformat>` elements 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 also change the report timestamp format. For example:

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

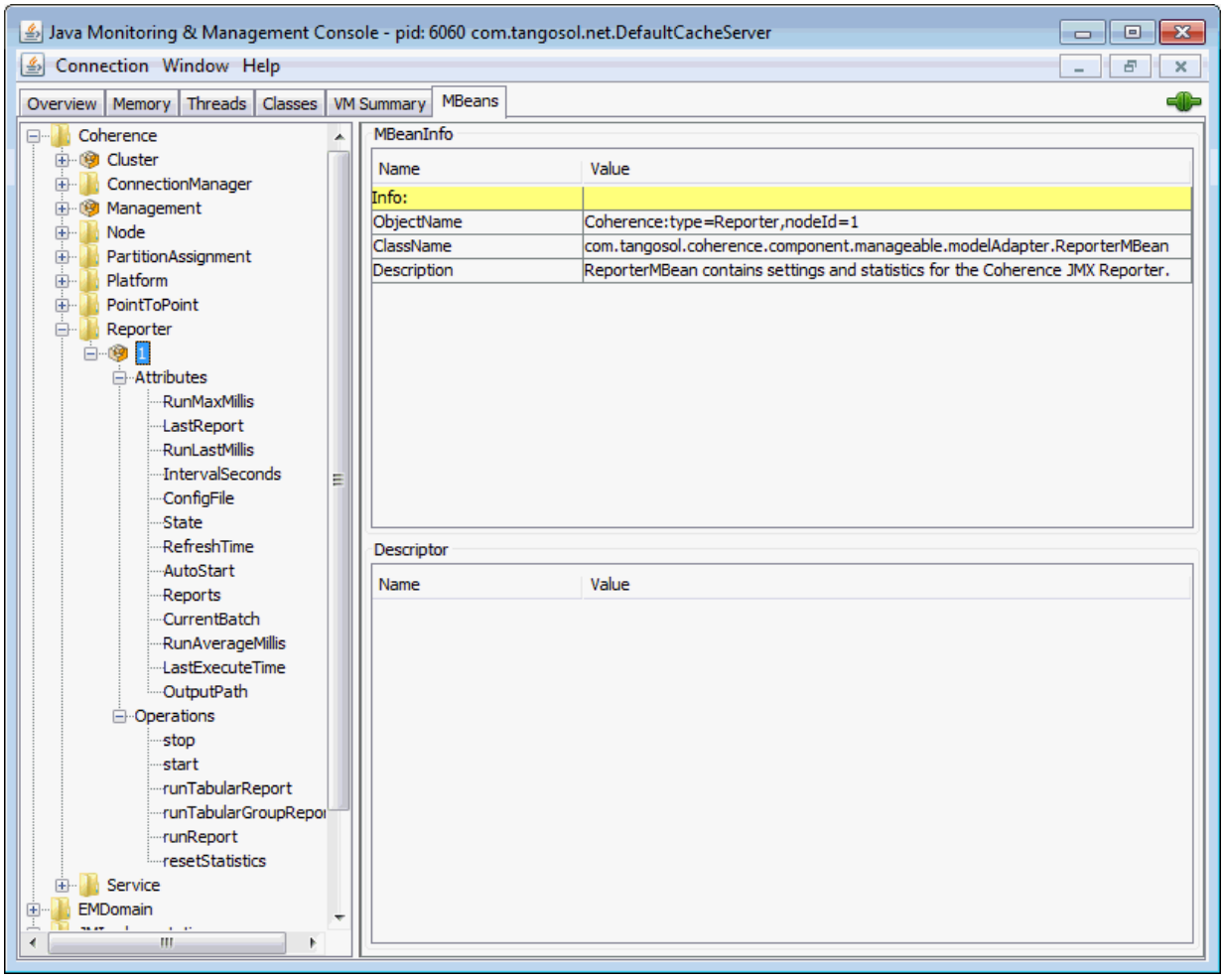
Administering Oracle Coherence Reporting Using the Reporter MBean

The `ReporterMBean` MBean administers Oracle Coherence reporting. The MBean contains attributes and operations to view JMX statistics and administer reporting in real time. See "Accessing Oracle Coherence MBeans" on page 2-8 for instructions on how to access Oracle Coherence MBeans.

Note: Changes that are made with the `ReporterMBean` MBean are not persistent. Settings revert to their original values when the cluster member restarts.

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

Figure 4–1 ReporterMBean MBean in JConsole



The `ReporterMBean` MBean performs the following tasks. See "ReporterMBean" on page A-24 for a complete reference of the `ReporterMBean` MBean's attributes and operations.

- Starting reporting – Use the `start` operation to start reporting if it is currently stopped.
- Stopping reporting – Use the `stop` operation to stop reporting. Reporting restarts when the cluster member restarts if the cluster member is configured to automatically start reporting.
- Running on-demand reports – Use the `runReport` operation to run either a single report or a report group one time. Specify the full or relative path of a report configuration file or a report group configuration file that is located on the classpath. Ensure that reporting is enabled before you run the report. For example,

enter `reports/report-memory-summary.xml` to run the memory summary report that is located in the `coherence.jar` file.

- Running on-demand tabular reports – Use the `runTabularReport` operation to run either a single report or a report group one time. The results are returned in a table format and display in a separate window. Ensure that reporting is enabled before you run the report.

  Specify the full or relative path of a report configuration file or a report group configuration file that is located on the classpath. For example, enter `reports/report-memory-summary.xml` to run the memory summary report that is located in the `coherence.jar` file.

  The `runTabularReport` operation also supports entering the XML syntax of a report or report group configuration file. Entering the XML syntax allows you to run a reports that are not on the classpath, which saves the steps of having to deploy the report or report group to all cache servers and performing a rolling restart. If you enter the XML of a report group, the individual reports must still be found on the classpath.

- Running on-demand tabular group reports – Use the `runTabularGroupReport` operation to run a report group one time and pass in the reports' XML syntax. The results are returned in a table format and display in a separate window. Ensure that reporting is enabled before you run the report.
- 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 increments from the new batch number.

Running Oracle Coherence Reporting in a Distributed Configuration

A distributed configuration is for situations where cluster stability has been observed. In this configuration, the distributed reporters run independently and the execution times do not align. Therefore, cluster-level analysis is extremely difficult, but member-level analysis during periods when members may be leaving or joining the cluster is still available.

When running reporting 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
```

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
```

5

Creating Custom Reports

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

This chapter includes the following sections:

Overview of Creating Custom Reports

Custom reports can use any MBeans that are registered in the Oracle Coherence MBean server domain (including custom MBeans). The use of different MBeans allows different management data to be combined in meaningful ways that are specific to a cluster implementation. Create custom reports during testing to correlate data and monitor trends when troubleshooting and planning for production.

Extract the predefined reporting configuration files from the `coherence.jar/reports` directory before creating custom reports. Use the files as a starting point when you create custom reports.

Report Configuration Files

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

- Report configuration file – A report configuration file constructs a report at run time. The file includes the name of the report, the MBean data to extract, and the organization of the data. 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 configures which reports to generate at run time. The file includes the name and location of each report configuration file, the output directory where to save the reports, and how often to refresh the reports. 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 Report Configuration Files

The topics in this section detail how to construct report configuration files based on the `coherence-report-config.xsd` file. See Appendix B, "Report File Configuration Elements," for details about the report file configuration elements.

This section includes the following topics:

- Specifying General Report Characteristics
- Querying MBeans in Reports
- Specifying Data Columns
- Using Filters in Reports
- Using Functions in Reports
- Using Aggregates in Reports
- Constructing Delta Functions

Specifying General Report Characteristics

Table 5–1 describes the elements that configure general report characteristics.

Table 5–1 General Report Elements

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

file-name Macros

Table 5–2 describes the macros predefined for use with the `file-name` element. These macros add a member name, a batch number, or a date to the file name.

Table 5–2 Macros for file-name Element

Macro	Description
`batch`	Adds a sequence number to the file name of the report
`date`	Adds the date (in the `YYYYMMDDHH` format) to the file name of the report. Use a date when the report is kept for a short time and then discarded.
`node`	Adds the member ID to the file name. 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 named `200901012_network_status.txt` on January 1, 2009, during hour 12. The file name 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 named `00012_network_status.txt` when the report is executed on member 12. Due to the volatile nature of the node ID, only use this macro during short-term analysis.

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

The following example creates a file named `0000000021_network_status.txt` on the 21st execution of the report. Due to the volatile nature of the batch, only use this macro during short-term analysis.

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

Querying MBeans in Reports

A query is the foundation of a report's information. The query pattern is a JMX `ObjectName` query string. The string can return one or more MBeans that are used to construct the report's data. Queries can be defined for the whole report or within specific columns in a report. A wild card (`*`) is often used to match multiple MBeans. Example 5–1 returns all registered `Service` MBeans. For details on the `<query>` element, see "query" on page B-9.

Example 5–1 Simple MBean Query String

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

Queries can use a filter definition and can also use column and filter references. For details on using filters, see "Using Filters in Reports" on page 5-7. Example 5–2 demonstrates how to reference a filter in a query and illustrates how to list all the node IDs and role names in the cluster where `RoleName` equals `CoherenceServer`.

Example 5–2 Including a List of the Cluster's Node IDs and Role Names 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>
```

Specifying Data Columns

Data columns can use JMX attributes, `ObjectName` key part, JMX composite attributes, JMX joined attributes, report macros, and report constants.

How to Include an Attribute

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

Example 5–3 illustrates how to include the `RoleName` attribute from the query pattern `Coherence:type=Node,*`.

Example 5–3 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 `ObjectName` that is returned from the query-pattern. This value can subsequently be included in the report.

Example 5–4 illustrates how to include the `nodeId` key part from the query pattern `Coherence:type=Node,*`.

Example 5–4 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

Use JMX composite values to include part of a composite data attribute (such as a `Map` attribute) in a report.

Example 5–5 illustrates how to include the `startTime` of the `LastGCInfo` attribute from the `GarbageCollector` MBean.

Example 5–5 Including Information from a Composite Attribute in a Report

```
<query>
   <pattern>Coherence:type=Platform,Domain=java.lang,subType=GarbageCollector,
   name=ParNew,*</pattern>
</query>

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

How to Include Information from Multiple MBeans

Use a JMX join attribute when a report requires information from multiple MBeans. The major consideration when creating a join is to determine the primary query, the join query, and the foreign key. The primary query is 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 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 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 braces (`{}`). The reporter does not check for cyclical references and fails during execution if a cycle is configured.

Including Multiple MBean Information Example

A report can use information from multiple MBeans. 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 use a `join` attribute.

Because more MBeans come from the `Cache` MBean than other MBeans, `Coherence:type=Cache,*` is the primary query and `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. Example 5–6 shows the configuration for this scenario.

Example 5–6 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

A report includes three report macros:

- Report Time (`report-time`) – The time and date for the report. This information is useful for time series analysis.
- Report Batch/Count (`report-count`) – A long identifier to correlate information from different reports executed at the same time.
- Reporting Member (`report-node`) – A member identifier to integrate information from the same report executed on different members or to exclude the executing member information from the report.

Example 5–7 illustrates how to include the execution time into the report.

Example 5–7 Including Execution Time in a Report

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

Example 5–8 illustrates how to include the Report Batch/Count.

Example 5–8 Including the Report Batch/Count in a Report

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

Example 5–9 illustrates how to include the execution member.

Example 5–9 Including the Execution Member

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

How to Include Constant Values

Static values or report parameters can use report constants. These constants are either double or string values. Often, filters use constant values to limit the results to a particular data set or in calculations.

Example 5–10 illustrates how to include a constant double of 1.0 in a report.

Example 5–10 Including a Constant Numeric Value in a Report

```
<column id ="One">
  <type>constant</type>
```

```
  <header>Constant1</header>
  <hidden>false</hidden>
  <data-type>double</data-type>
  <value>1.0</value>
</column>
```

Example 5–11 illustrates how to include the constant string `dist-Employee` in a report.

Example 5–11 Including a Constant String in a Report

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

Using Filters in Reports

Filters limit the data in reports. Filters are either comparison filters or composite filters. Comparison filters evaluate the results of two columns. Comparison filters are `equals`, `greater`, and `less`.

Composite filters evaluate the Boolean results from one or two filters. Composite filter types are `and`, `or`, and `not`. Each composite filter evaluates the filter parameters first to last and applies standard Boolean logic. Composite filter evaluation uses standard short-circuit logic. The use of cyclic references creates a run-time error.

Example 5–12 illustrates how to define an `equals` filter where `RoleRef` and `StringRef` are defined columns.

Example 5–12 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 5–13 illustrates how to define a filter in which the value of `PacketsResent` is greater than the value of `PacketsSent` (assuming that `PacketsResent` and `PacketsSent` are valid column references).

Example 5–13 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 5–14 illustrates how to define a filter in which the value of `PacketsResent` is less than the value of `PacketsSent` (assuming that `PacketsResent` and `PacketsSent` are valid column references).

Example 5–14 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 5–15 illustrates how to define an `and` filter (assuming that all `column-ref` values are valid).

Example 5–15 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 5–16 illustrates how to define an `or` filter (assuming that all `column-ref` values are valid).

Example 5–16 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 5–17 illustrates how to define a `not equals` filter, where `RoleRef` and `StringRef` are defined columns.

Example 5–17 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 in Reports

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 include (as parameters) other function columns.

Function Examples

Example 5–18 illustrates how to add two column values (`Attribute1` and `Attribute2`) and place the results into a third column (`Addition`).

Example 5–18 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 5–19 illustrates how to subtract one column value (`Attribute2`) from another column value (`Attribute1`) and place the results into a third column (`Subtraction`).

Example 5–19 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 5–20 illustrates how to multiply two column values (`Attribute1` and `Attribute2`) and place the results into a third column (`Multiplication`).

Example 5–20 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 5–21 illustrates how to divide one column value (`Attribute1`) by another (`Attribute2`) and place the results into a third column (`Division`). The result of all division is a `Double` data type.

Example 5–21 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 in Reports

Reporter aggregates combine multiple rows 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.

Example 5–22 illustrates how to sum the values in the `size` column.

Example 5–22 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>
```

Example 5–23 illustrates how to average the values in the `size` column.

Example 5–23 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>
```

Example 5–24 illustrates how to find the maximum value in the `size` column.

Example 5–24 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>
```

Example 5–25 illustrates how to find the minimum value in the `size` column.

Example 5–25 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 a 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.

A map on the reporter client stores the prior values for a report. This map is keyed by the delta key. By default, the delta key is the MBean name for the attribute. However, when a 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: Delta functions are only correct when the report is running as part of a report batch.

Delta Function Examples

Example 5–26 illustrates how to include a delta calculation of an attribute. (Assume that `PacketsSent` is a defined column.)

Example 5–26 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 5–27 illustrates how to include a delta calculation of an attribute with an alternate `delta` key. (Assume that `PacketsSent`, `NodeID`, and `TimeStamp` are defined columns.)

Example 5–27 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 Custom Report Group Configuration Files

The topics in this section detail how to create report group configuration files based on the `coherence-report-group-config.xsd` file. This configuration file is used at run time to determine what reports to generate, how often to refresh the reports, and where to save the reports. The report group configuration file also configures report parameters if required. See Appendix C, "Report Group Configuration Elements," for details about the report group configuration elements.

This section includes the following topics:

- 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 to refresh reports. Selecting an appropriate frequency is important: if the frequency is too short, the report contains too much data and consumes significant disk space; if the frequency is too long, the report does not contain enough information. Enter the value 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 user name that the member is executing must have read/write access to this path.

The path can be absolute or relative to the directory where the cluster member starts (`./`). 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 system-property=
      "tangosol.coherence.reporter.output.directory">/output</output-directory>
   ...
```

The `<output-directory>` element supports the use of a `system-property` attribute. The `system-property` attribute value is used at runtime to override the configured output directory. If a system property is not used to override the output directory, then the default output directory (`./`) is used. Any user-defined name can be used as the attribute value. The following example shows the default system property definition used by the predefined report group files:

```
<?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>1m</frequency>
   <output-directory system-property=
      "tangosol.coherence.reporter.output.directory">./</output-directory>
   ...
```

At runtime, specify the system property and include a path for the value. For example:

```
-Dtangosol.coherence.reporter.output.directory=/mydirectory
```

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, reporting must be enabled and the reporter must be configured to use a custom report group configuration file. See "Enabling Oracle Coherence Reporting on a Cluster Member" on page 4-2 for details on enabling reporting.

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

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

6

Analyzing Report Contents

This chapter describes predefined reports that are included with Oracle Coherence. Developers and system administrators use the reports to monitor and analyze operational statistics and troubleshoot potential problems. See *Administering HTTP Session Management with Oracle Coherence*Web* for reports specific to Oracle Coherence*Web.

This chapter includes the following sections:

Understanding the Cache Size Report

The cache size report indicates the size of a cache based on the number and size of the objects in the cache. The size does not include backup copies, indexes, or overhead. The size is reported for caches that set the `<unit-calculator>` subelement of `<local-scheme>` to `BINARY`. The name of the cache size report is *`timestamp`*`-cache-size.txt` where the timestamp is in `YYYYMMDDHH` format. For example, a file named `2009013101-cache-size.txt` represents a cache size report for January 31, 2009 at 1:00 a.m. Table 6–1 describes the contents of a 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 resets when the reporter restarts and is not consistent across members. However, it is helpful when trying to integrate files.
Service	String	The name of the cache service
Cache Name	String	The name of the cache
CacheSize	Double	The number of objects in the cache
MemoryBytes	Double	The number of bytes consumed by the objects in the cache
MemoryMB	Double	The number of Megabytes (MB) consumed by the objects in the cache
Avg Object Size	Double	The average amount of 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 name of the cache usage report is *timestamp*-cache-usage.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2010013113-cache-usage.txt represents a cache usage report for January 31, 2010 at 1:00 p.m. Table 6–2 describes the contents of the cache usage 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	A timestamp for each report refresh
Service	String	The name of the cache service
Cache Name	String	The name of the cache
Tier	String	Whether the cache resides in the front tier (local cache) or back tier (remote cache). The value is either front or back.
Total Puts	Double	The total number of puts for the cache across the cluster since the last report refresh
Total Puts Millis	Double	The total number of milliseconds spent per put() invocation across the cluster since the last report refresh
Total Gets	Double	The total number of gets for the cache across the cluster since the last report refresh
Total Gets Millis	Double	The total number of milliseconds spent per get() invocation across the cluster since the last report refresh
Total Hits	Double	The total number of visits for the cache across the cluster since the last report refresh
Total Hits Millis	Double	The total number of milliseconds spent per get() invocation that is a hit across the cluster since the last report refresh

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

Column	Data Type	Description
Total Misses	Double	The total number of misses for the cache across the cluster since the last report refresh
Total Misses Millis	Double	The total number of milliseconds spent per get() invocation that is a miss across the cluster since the last report refresh
Total Writes	Double	The total number of storage writes for the cache across the cluster since the last report refresh
Total Write Millis	Double	The total number of milliseconds spent in storage write operations across the cluster since the last report refresh
Total Reads	Double	The total number of reads from a cache store for the cache across the cluster since the last report refresh
Total Read Millis	Double	The total number of milliseconds spent on cache store reads for the cache across the cluster since the last time the report executed
Total Failures	Long	The total number of failures for the cache across the cluster since the last report refresh
Total Queue	Long	The sum of the queue link sizes across the cluster
Evictions	Long	The total number of evictions for the cache across the cluster since the last report refresh
Cache Prunes	Long	The total number of prunes for the cache across the cluster since the last report refresh
Cache Prunes Millis	Long	The total number of milliseconds spent in the prune operation across the cluster since the last report refresh

Understanding the Flash Journal Report

The flash journal report displays statistics to help determine how well data is being stored to flash memory. The name of the flash journal report is *timestamp*-flashjournal.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2010013113-flashjournal.txt represents a flash journal report for January 31, 2010 at 1:00 p.m. Table 6–3 describes the contents of the flash journal report.

Table 6–3 Contents of the Flash Journal 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	A timestamp for each report refresh
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 JournalBinaryStore objects that are using this journal
TotalDataSize	Long	The amount of data, in bytes, that is currently stored for this journal
TotalFileSize	Long	The total size of all journal files for this journal

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

Column	Data Type	Description
BacklogCount	Integer	The number of serialized values that have yet to be stored in the journal
BacklogSize	Integer	The maximum size, in bytes, of the backlog. The backlog is the amount of serialized values that have yet to be stored in the journal. 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 JCache Configuration Report

The JCache configuration report shows what configuration options have been set on a JCache cache. JCache caches are configured programmatically using the JCache API when the cache is created. The name of the report is *timestamp*-jcache-configuration.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013113-jcache-configuration.txt represents a management report for January 31, 2009 at 1:00 p.m. Table 6–4 describes the contents of the JCache configuration report.

Table 6–4 Contents of the JCache Configuration 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.
ReportTime	Date	A timestamp for each report refresh
CacheManager	String	The JCache CacheManager instance that created the cache. Coherence cache managers are scoped and uniquely identified by a cache configuration file that defines a JCacheNamespace handler class.
Cache	String	The name of the cache
KeyType	String	The required key type for the cache.
ValueType	String	The required value type for the cache.
ManagementEnabled	Boolean	Specifies whether management is enabled for the cache
StatisticsEnabled	Boolean	Specifies whether performance statistics are being collected for the cache
ReadThrough	Boolean	Specifies whether the cache operates in read-through mode
WriteThrough	Boolean	Specifies whether the cache operates in write-through mode
StoreByValue	Boolean	Specifies whether the cache uses store-by-value or store by-reference semantics. A value of true indicates that keys and values are stored by value. A value of false indicates that keys and values are stored by reference.

Understanding the JCache Statistics Report

The JCache statistic report contains information that is used to evaluate how well a JCache cache is performing. The name of the report is *timestamp*-jcache-statistics.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013113-jcache-statistics.txt represents a management report for January 31, 2009 at 1:00 p.m. Table 6–5 describes the contents of the JCache statistics report.

Table 6–5 Contents of the JCache Statistics 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.
ReportTime	Date	A timestamp for each report refresh
CacheManager	String	The JCache CacheManager instance that created the cache. Coherence cache managers are scoped and uniquely identified by a cache configuration file that defines a JCacheNamespace handler class.
Cache	String	The name of the cache
CacheGets	Long	The total number of get operations. The value is equal to the sum of hits and misses and does not include operations that check for the existence of a key.
CachePuts	Long	The total number of put operations including operations that replace and existing entry
CacheRemovals	Long	The total number of remove operations. The value does not include evictions initiated by the cache to free up space.
CacheHits	Long	The number of successful get operations
CacheMisses	Long	The number of unsuccessful get operations
CacheEvictions	Long	The total number of evictions from the cache. An eviction is initiated by the cache to free up space. An eviction is not considered a remove operation. **Note:** This attribute is not implemented by the Coherence JCache provider.
AverageGetTime	Float	The average time to perform get operations. For read-through caches, the time does not include the time that is required to load entries because of a cache miss.
AveragePutTime	Float	The average time to perform put operations
AverageRemoveTime	Float	The average time to perform remove operations
CacheHitPercentage	Float	The percentage of cache requests that return an entry. The percentage is reported as a decimal value and is calculated using the value of cache hits divided by cache get operations.
CacheMissPercentage	Float	The percentage of cache requests that do not return an entry. The percentage is reported as a decimal value and is calculated using the value of cache misses divided by cache get operations.

Understanding the Management Report

The management report contains refresh statistics to help determine if the management framework is providing a timely view of management data for all MBeans. The name of the management report is *timestamp*-management.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013113-Management.txt represents a management report for January 31, 2009 at 1:00 p.m. Table 6–6 describes the contents of the management report.

Table 6–6 Contents of the Management 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.
ReportTime	Date	A timestamp for each report refresh
RefreshPolicy	String	The refresh policy that is currently set. The policy determines how to refresh data for remote models.
RefreshTime	Date	The timestamp when this model was last retrieved from a corresponding member. For local servers it is the local time.
RefreshExcessCount	Long	The number of times that the MBean server predictively refreshed information and the information was not accessed
RefreshCount	Long	The total number of snapshots retrieved since the statistics were last reset
RefreshPredictionCount	Long	The number of times that the MBean server used a predictive algorithm to refresh MBean information
RefreshTimeoutCount	Long	The number of times that this management member has timed out while attempting to refresh remote MBean attributes

Understanding the Memory Status Report

The memory status report contains statistics to help understand memory consumption on each member and across the grid. A memory status report must be run as part of a report group. The memory status report relies on platform MBean information. See "Filtering MBeans" on page 2-4. The name of the memory status report is *timestamp*-memory-status.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013115-memory-status.txt represents a memory status report for January 31, 2009 at 3:00 p.m. Table 6–7 describes the contents of the memory status report.

Table 6–7 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 resets when the reporter restarts and is not consistent across members. However, it is helpful when trying to integrate files.
Report Time	Date	A timestamp for each report refresh
JVM Uptime	Long	The amount of time since the last JVM start
Node Id	Long	The member for the memory statistics

Table 6–7 (Cont.) Contents of the Memory Status Report

Column	Data Type	Description
Gc Name	String	The name of the garbage collector
CollectionCount	Long	The number of garbage collections since the last JVM start
Delta Collection Count	Long	The number of garbage collections since the last report refresh
CollectTime	Long	The number of milliseconds that the JVM has spent on garbage collection since the start of the JVM
Delta Collect Time	Long	The number of milliseconds that the JVM has spent on garbage collection since the last report refresh
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 the report ran
Heap Init	Long	The number of heap bytes initialized at the time the report ran
Heap Max	Long	The maximum number of bytes used by the JVM since its start
Heap Used	Long	The bytes used by the JVM at the time the report ran

Understanding the Network Health Detail Report

The network health detail report contains member-level details to help determine the health of network communications. The name of the network health detail report is *timestamp*-network-health-detail.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013114-network-health-detail.txt represents a network health detail report for January 31, 2009 at 2:00 p.m. Table 6–8 describes the contents of the network health detail report.

Table 6–8 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 resets when the reporter restarts and is not consistent across members. However, it is helpful when trying to integrate files.
Report Time	Date	A timestamp for each report refresh
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.

Table 6–8 (Cont.) Contents of the Network Health Detail Report

Column	Data Type	Description
Tx Success	Double	The publisher success rate for the member. If this value is within 2% to 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 (see Table 6–9), the corresponding member may be having difficulty communicating with the cluster. A poor success rate can occur due to a constrained CPU, constrained network bandwidth, or high network latency.
RX Success	Double	The receiver success rate for the member. If this value is within 2% to 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 (see Table 6–9), the corresponding member may be having difficulty communicating with the cluster. A poor success rate can occur due to a constrained CPU, constrained network bandwidth, or high network latency.
PacketsSent	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 last report refresh
PacketsResent	Long	The total number of network packets re-sent by the member. Packets are re-sent when the receiver of the packet receives an 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 re-sent by the member since the last report refresh
PacketsRepeated	Long	The total number of packets received multiple times
Current Repeated	Long	The number of packets received multiple times since the last report refresh
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 report refresh

Understanding the Network Health Report

The network health report contains the primary aggregates to help determine the health of the network communications. The name of the network health report is *timestamp*-network-health.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013113-network-health.txt represents a network health report for January 31, 2009 at 1:00 p.m. Table 6–9 describes the contents of the network health report.

Table 6–9 Contents of the Network Health 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.

Table 6–9 (Cont.) Contents of the Network Health Report

Column	Data Type	Description
Report Time	Date	A timestamp for each report refresh
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, analyze the network health detail report.
Grid Rx Success	Double	The receiver success rate for the grid as a whole. If this value is below 90%, analyze the network health detail report.
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, analyze the network health detail report.
Grid TX Success	Double	The publisher success rate for the grid as a whole. If this value is below 90%, analyze the network health detail report.

Understanding the Node List Report

The node list report provides information to help identify a cluster member. 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 the <member-identity> element in the *Developing Applications with Oracle Coherence*. The name of the nodes list report is *timestamp*-nodes.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013101-nodes.txt represents a node list report for January 31, 2009 at 1:00 a.m. Table 6–10 describes the contents of the node list report.

Table 6–10 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 resets when the reporter restarts and is not consistent across members. However, it is helpful when trying to integrate files.
Report Time	Date	A timestamp for each report refresh
RefreshTime	Date/Time	The time at 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.
Node Id	String	The numeric member identifier
UnicastAddress	String	The Unicast address for the member
MemberName	String	The member name
ProcessName	String	The process name for the member
RoleName	String	The role name for the member
MachineName	String	The computer name for the member
RackName	String	The rack name for the member
SiteName	String	The site name for the member

Understanding the Proxy Report

The proxy report provides information about proxy servers and the information being transferred to clients. The name of the proxy report is *timestamp*-network-report-proxy.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013101-report-proxy.txt represents a proxy report for January 31, 2009 at 1:00 a.m. Table 6–11 describes the contents of the proxy report.

Table 6–11 Contents of the Proxy 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	A timestamp for each report refresh
RefreshTime	Date/Time	The timestamp when this model was last retrieved from a corresponding member. For local servers it is the local time.
Service Name	String	The name of the proxy service
HostIp	String	The IP Address and Port of the proxy service
NodeId	String	The numeric member identifier
ConnectionCount	Long	The current number of connections to the proxy service
OutgoingByteBacklog	Long	The number of bytes queued to be sent by the proxy service
OutgoingMessageBacklog	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 report refresh
Bytes Received	Long	The number of bytes received by the proxy service since the last report refresh
Messages Sent	Long	The number of messages sent by the proxy service since the last report refresh
Messages Received	Long	The number of messages received by the proxy service since the last report refresh

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 name of the ram journal report is *timestamp*-ramjournal.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2010013113-ramjournal.txt represents a ram journal report for January 31, 2010 at 1:00 p.m. Table 6–12 describes the contents of the ram journal report.

Table 6–12 Contents of the Ram Journal 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.

Table 6–12 (Cont.) Contents of the Ram Journal Report

Column	Data Type	Description
Report Time	Date	A timestamp for each report refresh
Node Id	Long	The member for the RAM journal statistics
FileCount	Integer	The number of journal files that are currently in use
BinaryStoreCount	Integer	The number of active JournalBinaryStore objects that are using this journal
TotalDataSize	Long	The amount of data, in bytes, that is 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 for monitoring the health and performance of a service. The Request Count and Task Count values help to determine the performance and throughput of the service. The RequestPendingCount and Task Backlog values help to identify capacity issues or blocked processes. The Task Hung Count, Task Timeout Count, Thread Abandoned Count, and Request Timeout Count values represent the number of unsuccessful executions that have occurred in the system. The name of the service report is *timestamp*-service.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2010013113-service.txt represents a service report for January 31, 2010 at 1:00 p.m. Table 6–13 describes the contents of the service report.

Table 6–13 Contents of the Service 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	A timestamp for each report refresh
Service	String	The service name
Node Id	String	The numeric member identifier
RefreshTime	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 refresh 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 refresh
Task Count	Long	The number of tasks executed since the last report refresh
TaskBacklog	Long	The task backlog at the time of the report
Task Timeout Count	Long	The number of task timeouts since the last report refresh

Table 6–13 (Cont.) Contents of the Service Report

Column	Data Type	Description
`Task Hung Count`	`Long`	The number of tasks that hung since the last report refresh
`Thread Abandoned Count`	`Long`	The number of threads abandoned since the last report refresh

A

Oracle Coherence MBeans Reference

This appendix provides a reference of the MBeans for managing and monitoring different parts of Oracle Coherence. The reference provides a detailed description of the attributes and operations of the MBeans.

The `Registration` interface also includes descriptions of the MBeans. See the `Registration` interface in the *Java API Reference for Oracle Coherence*. An MBean-capable agent (such as JConsole) also shows MBean information. See Chapter 2, "Using JMX to Manage Oracle Coherence," for more information about accessing Oracle Coherence MBeans.

Table A–1 lists the MBeans for managing and monitoring Oracle Coherence.

Table A–1 Oracle Coherence MBeans

MBean	Description
`CacheMBean`	Represents a cache. A cluster member includes zero or more instances of this managed bean.
`ClusterMBean`	Represents a cluster. Each cluster member includes a single instance of this managed bean.
`ClusterNodeMBean`	Represents a cluster member. Each cluster member includes a single instance of this managed bean.
`ConnectionManagerMBean`	Represents an Oracle Coherence*Extend proxy. A cluster member includes zero or more instances of this managed bean.
`ConnectionMBean`	Represents a remote client connection through Oracle Coherence*Extend. A cluster member includes zero or more instances of this managed bean.
`FlashJournalRM`	Represents a flash journal resource manager. The managed bean is an instance of the `JournalMBean` interface. Each cluster member includes a single instance of this managed bean.
`ManagementMBean`	Represents the grid JMX infrastructure. Each cluster member includes a single instance of this managed bean.
`PointToPointMBean`	Represents the network status between two cluster members. Each cluster member includes a single instance of this managed bean.
`RamJournalRM`	Represents a RAM journal resource manager. The managed bean is an instance of the `JournalMBean` interface. Each cluster member includes a single instance of this managed bean.
`ReporterMBean`	Represents the Oracle Coherence reporter. Each cluster member includes a single instance of this managed bean.
`ServiceMBean`	Represents a clustered service. A cluster member includes zero or more instances of this managed bean.

Table A–1 (Cont.) Oracle Coherence MBeans

MBean	Description
StorageManagerMBean	Represents a storage instance for a storage-enabled distributed cache service. A cluster member includes zero or more instances of this managed bean.
TransactionManagerMBean	Represents a transaction manager. A cluster member includes zero or more instances of this managed bean.

CacheMBean

The CacheMBean MBean represents a cache. A cluster member includes 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 MBean attributes are writable and allow the behavior of a cache to be changed in real time.

The object name of the MBean is:

type=Cache,service=*service name*,name=*cache name*,nodeId=*cluster node 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 attributes are invoked that do not apply. See *Developing Applications with Oracle Coherence* 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	read-only	The average number of milliseconds per get() invocation since the statistics were last reset
AverageHitMillis	Double	read-only	The average number of milliseconds per get() invocation that is a hit since the statistics were last reset
AverageMissMillis	Double	read-only	The average number of milliseconds per get() invocation that is a miss since the statistics were last reset
AveragePutMillis	Double	read-only	The average number of milliseconds per put() invocation since the statistics were last reset
BatchFactor	Double	read/write	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 QueueDelay 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 QueueDelay attribute is greater than zero) and the cache store implements the storeAll() method. The value of the element is expressed as a percentage of the QueueDelay interval. Valid values are doubles in the interval [0.0, 1.0].
CacheHits	Long	read-only	The rough number of cache visits since the statistics were last reset. A cache hit is a read operation invocation (that is, get()) for which an entry exists in this map.

Table A–2 (Cont.) CacheMBean Attributes

Attribute	Type	Access	Description
CacheHitsMillis	Long	read-only	The total number of milliseconds (since the statistics were last reset) for the get() operations for which an entry existed in this map
CacheMisses	Long	read-only	The rough number of cache misses since the statistics were last reset
CacheMissesMillis	Long	read-only	The total number of milliseconds (since the statistics were last reset) for the get() operations for which no entry existed in this map
CachePrunes	Long	read-only	The number of prune operations since the statistics were last reset. A prune operation occurs every time the cache reaches its high-water mark as specified by the HighUnits attribute.
CachePrunesMillis	Long	read-only	The total number of milliseconds for the prune operations since the statistics were last reset
Description	String	read-only	The cache description
ExpiryDelay	Integer	read/write	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	read/write	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	read-only	The rough probability (*0 <= p <= 1*) that the next invocation is a hit since the last time statistics were reset
LowUnits	Integer	read/write	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	read-only	The persistence type for this cache. Possible values include: NONE, READ-ONLY, WRITE-THROUGH, WRITE-BEHIND.
QueueDelay	Integer	read/write	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	read-only	The size of the write-behind queue. This attribute is only applicable if the persistence type is WRITE-BEHIND.
RefreshFactor	Double	read/write	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 that 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 the factor is zero, refresh-ahead scheduling is disabled.
RefreshTime	Date	read-only	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

Attribute	Type	Access	Description
RequeueThreshold	Integer	read/write	The size of the write-behind queue at which additional actions could be taken. If the size is zero, write-behind requeuing is disabled. Otherwise, this value controls the frequency of the corresponding log messages. For example, a value of 100 produces a log message every time the size of the write queue is a multiple of 100.
Size	Integer	read-only	The number of entries in the cache
StoreAverageBatchSize	Long	read-only	The average number of entries stored for each cache store write operation since the statistics were last reset. A call to the store() method is counted as a batch of one; whereas a call to the storeAll() method is counted as a batch of the passed Map size. The value is -1 if the persistence type is NONE.
StoreAverageReadMillis	Long	read-only	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	read-only	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	read-only	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	read-only	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	read-only	The total number of load operations since the statistics were last reset. The value is -1 if the persistence type is NONE.
StoreWriteMillis	Long	read-only	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	read-only	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	read-only	The total number of get() operations since the statistics were last reset
TotalGetsMillis	Long	read-only	The total number of milliseconds spent on get() operations since the statistics were last reset
TotalPuts	Long	read-only	The total number of put() operations since the statistics were last reset
TotalPutsMillis	Long	read-only	The total number of milliseconds spent on put() operations since the statistics were last reset
UnitFactor	Integer	read-only	The factor by which the Units, LowUnits, and HighUnits 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	read-only	The size of the cache measured in units. This value must be adjusted by the UnitFactor.

Operations

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

ClusterMBean

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

The object name of the MBean is:

```
type=Cluster
```

Attributes

Table A–3 describes the attributes for `ClusterMBean`.

Table A–3 ClusterMBean Attributes

Attribute	Type	Access	Description
`ClusterName`	`String`	read-only	The name of the cluster
`ClusterSize`	`Integer`	read-only	The total number of cluster members
`LicenseMode`	`String`	read-only	The license mode that this cluster is using. Possible values are `Evaluation`, `Development`, or `Production`.
`LocalMemberId`	`Integer`	read-only	The member ID for the cluster member that is collocated with the reporting MBean server. The value is `-1` if the cluster service is not running.
`MemberIds`	`Integer[]`	read-only	An array of all existing cluster member IDs
`Members`	`String[]`	read-only	An array of all existing cluster members
`MembersDeparted`	`String[]`	read-only	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 `MembersDepartureCount` is the total count of departed members and not the size of this array.
`MembersDepartureCount`	`Long`	read-only	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.
`OldestMemberId`	`Integer`	read-only	The senior cluster member ID. The value is `-1` if the cluster service is not running.
`RefreshTime`	`Date`	read-only	The timestamp when this model was last retrieved from a corresponding member. For local servers, it is the local time.
`Running`	`Boolean`	read-only	Specifies whether the cluster is running
`Version`	`String`	read-only	The Oracle Coherence version

Operations

Table A–4 describes the operations for `ClusterMBean`.

Table A–4 ClusterMBean Operations

Operation	Parameters	Return Type	Description
`ensureRunning`	Not applicable	`Void`	Ensures that the cluster service is running on this member
`logClusterState`	`String sRole`	`Void`	Logs the state (full thread dump and outstanding polls) on cluster members that are running with the specified role. The role of a cluster member is defined within the `<member-identity>` element of an operational override file. See *Developing Applications with Oracle Coherence* for details on setting up member identity information. If the `sRole` parameter is not specified (no value in the field), then state is logged on all cluster members.
`shutdown`	Not applicable	`Void`	Shuts down the cluster service on this member

ClusterNodeMBean

The `ClusterNodeMBean` MBean represents a cluster member. Each cluster member includes a single instance of this managed bean. This MBean provides 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 object name of the MBean is:

`type=Node,nodeId=`*`cluster node id`*

Attributes

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

Table A–5 ClusterNodeMBean Attributes

Attribute	Type	Access	Description
`BufferPublishSize`	`Integer`	read/write	The buffer size of the unicast datagram socket that is 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.
`BufferReceiveSize`	`Integer`	read/write	The buffer size of the unicast datagram socket that is 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.
`CpuCount`	`Integer`	read-only	Number of CPU cores for the computer that this member is running on
`FlowControlEnabled`	`Boolean`	read-only	Indicates whether `FlowControl` is enabled. To enable flow control, see the `<flow-control>` element in *Developing Applications with Oracle Coherence*.
`Id`	`Integer`	read-only	The short member ID that uniquely identifies the member now and does not change for the life of this member
`LoggingDestination`	`String`	read-only	The output device used by the logging system. Valid values are `stdout`, `stderr`, `jdk`, `log4j`, or a file name.
`LoggingFormat`	`String`	read/write	Specifies how messages are formatted before being passed to the log destination
`LoggingLevel`	`Integer`	read/write	Specifies which logged messages are output to the log destination. Valid values are nonnegative integers. A value of `-1` disables all logger output.
`LoggingLimit`	`Integer`	read/write	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 `Integer.MAX_VALUE` (2147483647). A value of zero implies `Integer.MAX_VALUE`.
`MachineId`	`Integer`	read-only	The computer ID of this member

Table A–5 (Cont.) ClusterNodeMBean Attributes

Attribute	Type	Access	Description
MachineName	String	read-only	A unique name for the computer. The name should be the same for all members that are on the same computer and different for members that are on different computers.
MemberName	String	read-only	A unique name for a member. The name must be unique for every member.
MemoryAvailableMB	Integer	read-only	The total amount of memory (in MB) that is available in the JVM for new objects
MemoryMaxMB	Integer	read-only	The maximum amount of memory (in MB) that the JVM attempts to use
MulticastAddress	String	read-only	The multicast socket IP address that is used by this member for group communication
MulticastEnabled	Boolean	read-only	Specifies whether this member uses multicast for group communication. If `false`, this member uses the addresses listed in the `WellKnownAddresses` attribute to join the cluster and point-to-point unicast to communicate with other members of the cluster.
MulticastPort	Integer	read-only	The multicast socket port that is used by this member for group communication
MulticastThreshold	Integer	read/write	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
MulticastTTL	Integer	read-only	The time-to-live for multicast packets that were sent out on this member`s multicast socket
NackEnabled	Boolean	read-only	Indicates whether the early packet loss detection protocol is enabled
NackSent	Long	read-only	The total number of NACK packets that were sent since the member statistics were last reset
PacketDeliveryEfficiency	Float	read-only	The efficiency of packet loss detection and retransmission. A low efficiency indicates a high rate of unnecessary packet retransmissions.
PacketsBundled	Long	read-only	The total number of packets that were bundled before transmission. The total number of network transmissions equals *PacketsSent - PacketsBundled*.
PacketsReceived	Long	read-only	The number of packets that were received since the member statistics were last reset
PacketsRepeated	Long	read-only	The number of duplicate packets that were received since the member statistics were last reset
PacketsResent	Long	read-only	The number of packets that were re-sent since the member statistics were last reset. A packet is re-sent when there is no ACK received within a timeout period.
PacketsResentEarly	Long	read-only	The total number of packets that were re-sent too soon. A packet is re-sent too soon when there is a NACK indicating that the packet has not been received.
PacketsResentExcess	Long	read-only	The total number of packet retransmissions that were later proven unnecessary
PacketsSent	Long	read-only	The number of packets that were sent since the member statistics were last reset

Table A–5 (Cont.) ClusterNodeMBean Attributes

Attribute	Type	Access	Description
Priority	Integer	read-only	The priority, or weight, of the member. This value is used to determine tie-breakers
ProcessName	String	read-only	A 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 Name attribute of the system RuntimeMXBean, which normally represents the process identifier (PID).
ProductEdition	String	read-only	The product edition this member is running. Possible values are Standard Edition, Enterprise Edition, and Grid Edition.
PublisherPacketUtilization	Float	read-only	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	read-only	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	read-only	The current state of the cluster quorum
RackName	String	read-only	A name for the rack on which the member is located. This name 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	read-only	The receiver packet utilization for this cluster member since the socket was last reopened. This value is a ratio of the number of bytes that were 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	read-only	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 that were successfully acknowledged in a first attempt to the total number of received packets. A failure count increments when a redelivery of a previously received packet is detected. Very high inbound network latency or lost ACK packets can cause a low success rate.
RefreshTime	Date	read-only	The timestamp when this model was last retrieved from a corresponding member. For local servers, it is the local time.

Table A–5 (Cont.) ClusterNodeMBean Attributes

Attribute	Type	Access	Description
ResendDelay	Integer	read/write	The minimum number of milliseconds that a packet remains queued in the publisher`s resend queue before it is re-sent to any recipients 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-sending of dropped packets. Change of this value may require a change in the SendAckDelay value.
RoleName	String	read-only	A name that indicates the role of a cluster member to the application. Use a name that is meaningful to the application.
SendAckDelay	Integer	read/write	The minimum number of milliseconds between the queuing and sending of an ACK packet. This value should not be more then half of the ResendDelay value.
SendQueueSize	Integer	read-only	The number of packets that are currently scheduled for delivery. This number includes both packets that are to be sent immediately and packets that have been sent and are awaiting for an acknowledgment. Packets that do not receive an acknowledgment within the ResendDelay interval are automatically re-sent.
SiteName	String	read-only	A name for the site where the member is located. The name 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	read-only	The number of CPU sockets for the computer that this member is running on
Statistics	String	read-only	The statistics for this cluster member in a human readable format
TcpRingFailures	Long	read-only	The number of TcpRing disconnects that were recovered 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 value of -1 indicates that TcpRing is disabled.
Timestamp	Date	read-only	The date/time value (in cluster time) that this member joined the cluster
TrafficJamCount	Integer	read/write	The maximum total number of packets in the send and resend queues that forces the publisher to pause client threads. A value of zero means no limit.
TrafficJamDelay	Integer	read/write	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	read-only	The datagram socket IP address that is used by this member for point-to-point communication

Table A–5 (Cont.) ClusterNodeMBean Attributes

Attribute	Type	Access	Description
UnicastPort	Integer	read-only	The datagram socket port that is used by this member for point-to-point communication
WeakestChannel	Integer	read-only	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[]	read-only	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	Not applicable	void	Reset the cluster member statistics
logNodeSate	Not applicable	Void	Logs a full thread dump and outstanding polls for the services on this cluster member
shutdown	Not applicable	void	Stop all the clustered services running at this member (controlled shutdown). The management of this member is not available until the member restarts (manually or programmatically).

ConnectionManagerMBean

The `ConnectionManagerMBean` MBean represents an Oracle Coherence*Extend proxy. A cluster member includes 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 object name of the MBean is:

`type=ConnectionManager,name=`*`service name`*`,nodeId=`*`cluster node id`*

Attributes

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

Table A–7 ConnectionManagerMBean Attributes

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

Operations

The `ConnectionManagerMBean` MBean has no operations.

ConnectionMBean

The `ConnectionMBean` MBean represents a remote client connection through Oracle Coherence*Extend. A cluster member includes 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 object name of the MBean is:

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

Attributes

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

Table A–8 ConnectionMBean Attributes

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

Operations

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

Table A–9 ConnectionMBean Operations

Operation	Parameters	Return Type	Description
`closeConnection`	Not applicable	`void`	Close the corresponding connection.
`resetStatistics`	Not applicable	`void`	Reset the connection statistics.

FlashJournalRM

The `JournalMBean` MBean provides an interface for the flash journal resource manager (`FlashJournalRM`). The resource manager stores data to flash memory. Each cluster member includes a single instance of this managed bean, which provides operational statistics.

The object name of the MBean is:

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

Attributes

Table A–10 describes the attributes for `JournalMBean` that pertain to the flash journal resource manager.

Table A–10 JournalMBean Attributes for FlashJournalRM

Attribute	Type	Access	Description
`BacklogCount`	`Integer`	read-only	The number of serialized values that have yet to be stored in the journal
`BacklogSize`	`Integer`	read-only	The amount of data that has been enqueued for the journal to write to disk, but which has not yet been written to disk. The maximum amount of memory used by the backlog is at least twice the configured amount, because the data enqueued is held in binary form and rendered to the write-behind buffers.
`BinaryStoreCount`	`Integer`	read-only	The number of active `JournalBinaryStore` objects that are using this journal
`BufferSize`	`Integer`	read-only	The size of the buffers that are used to write data to a journal file
`CollectorLoadFactor`	`Double`	read-only	The threshold after which files are eligible for garbage collection. The larger the value, the more aggressively files are collected.
`CurrentCollectorLoadFactor`	`Double`	read-only	The current load factor threshold at which files are being garbage collected
`FileCount`	`Integer`	read-only	The number of journal files that are currently in use
`HighFileCount`	`Integer`	read-only	The high file count after which compulsory compaction (garbage collection) occurs for the Journal
`HighestLoadFactor`	`Double`	read-only	The approximate high-water mark of the Journal capacity utilization
`MaxBacklogSize`	`Integer`	read-only	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.
`MaxFileSize`	`Long`	read-only	The maximum allowable size of an individual journal file
`MaxJournalFilesNumber`	`Integer`	read-only	The maximum number of journal files that can be used
`MaxPoolSize`	`Integer`	read-only	The maximum size, in bytes, of the buffer pool
`MaxTotalRam`	`Long`	read-only	This attribute does not pertain to a flash journal and returns `-1`.
`MaxValueSize`	`Integer`	read-only	The maximum allowable size, in bytes, for serialized values

Table A–10 (Cont.) JournalMBean Attributes for FlashJournalRM

Attribute	Type	Access	Description
`NioRam`	`Boolean`	read-only	Whether the RAM journal uses on-heap byte buffers or off-heap NIO buffers (NIO RAM). Valid values are `true` and `false`. A `true` value indicates off-heap NIO buffers. A `false` value indicates on-heap byte buffers. The default value is `false`.
`PoolSize`	`Integer`	read-only	The total size, in bytes, of all available buffers in the pool
`TotalCompactionCount`	`Integer`	read-only	The total number of times compaction (garbage collection) has been done for the journal
`TotalDataSize`	`Long`	read-only	The amount of data, in bytes, that is currently stored for this journal
`TotalFileSize`	`Long`	read-only	The total size of all journal files for this journal

Operations

The `JournalMBean` MBean has no operations.

ManagementMBean

The ManagementMBean MBean represents the grid JMX infrastructure. Each cluster member includes 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 object name of the MBean is:

```
type=Management
```

Attributes

Table A–11 describes the attributes for ManagementMBean.

Table A–11 ManagementMBean Attributes

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

`type=PointToPoint,nodeId=`*`cluster node id`*

Attributes

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

Table A–12 PointToPointMBean Attributes

Attribute	Type	Access	Description
`DeferredPackets`	`Integer`	read-only	The number of packets that were 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 `Threshold` attribute. The value of this attribute is only meaningful if the viewing member has `FlowControl` enabled. See the `<flow-control>` element in *Developing Applications with Oracle Coherence*.
`Deferring`	`Boolean`	read-only	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 `FlowControl` enabled. See the `<flow-control>` element in *Developing Applications with Oracle Coherence*.
`LastIn`	`Long`	read-only	The number of milliseconds that have elapsed since the viewing member last received an acknowledgment from the viewed member
`LastOut`	`Long`	read-only	The number of milliseconds that have elapsed since the viewing member last sent a packet to the viewed member
`LastSlow`	`Long`	read-only	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
`OutstandingPackets`	`Integer`	read-only	The number of packets that the viewing member has sent to the viewed member that have yet to be acknowledged. The value of this attribute is only meaningful if the viewing member has `FlowControl` enabled. See the `<flow-control>` element in *Developing Applications with Oracle Coherence*.
`PauseRate`	`Float`	read-only	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 garbage collection. The value of this attribute is only meaningful if the viewing member has `FlowControl` enabled. See the `<flow-control>` element in *Developing Applications with Oracle Coherence*.

Table A–12 (Cont.) PointToPointMBean Attributes

Attribute	Type	Access	Description
Paused	Boolean	read-only	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 FlowControl enabled. See the <flow-control> element in *Developing Applications with Oracle Coherence*.
PublisherSuccessRate	Float	read-only	The publisher success rate from the viewing member to the viewed member since the statistics were last reset
ReceiverSuccessRate	Float	read-only	The receiver success rate from the viewing member to the viewed member since the statistics were last reset
RefreshTime	Date	read-only	The timestamp when this model was last retrieved from a corresponding member. For local servers, it is the local time
Threshold	Integer	read-only	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 FlowControl enabled. See the <flow-control> element in *Developing Applications with Oracle Coherence*.
ViewedMemberId	Integer	read/write	The ID of the member being viewed
ViewerStatistics	String[]	read-only	A 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
resetStatistics	Not applicable	void	Reset the viewing member`s point-to-point statistics for all other members.
trackWeakest	Not applicable	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 rate is below 1.0.

RamJournalRM

The JournalMBean MBean provides an interface for the RAM journal resource manager (RamJournalRM). The resource manager is responsible for storing data to RAM memory. Each cluster member includes a single instance of this managed bean, which provides operational statistics.

The object name of the MBean is:

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

Attributes

Table A–14 describes the attributes for JournalMBean that pertain to the RAM journal resource manager.

Table A–14 JournalMBean Attributes for RamJournalRM

Attribute	Type	Access	Description
BacklogCount	Integer	read-only	This attribute does not pertain to a RAM journal and returns -1.
BacklogSize	Integer	read-only	This attribute does not pertain to a RAM journal and returns -1.
BinaryStoreCount	Integer	read-only	The number of active JournalBinaryStore objects that are using this journal
BufferSize	Integer	read-only	This attribute does not pertain to a RAM journal and returns -1.
CollectorLoadFactor	Double	read-only	The threshold after which files are eligible for garbage collection. The larger the value, the more aggressively files are collected.
CurrentCollectorLoadFactor	Double	read-only	The current load factor threshold at which files are being garbage collected
FileCount	Integer	read-only	The number of journal files that are currently in use
HighFileCount	Integer	read-only	This attribute does not pertain to a RAM journal and returns -1.
HighestLoadFactor	Double	read-only	This attribute does not pertain to a RAM journal and returns 0.0.
MaxBacklogSize	Integer	read-only	This attribute does not pertain to a RAM journal and returns -1.
MaxFileSize	Long	read-only	The maximum allowable size of an individual journal file
MaxJournalFilesNumber	Integer	read-only	The maximum number of journal files that can be used
MaxPoolSize	Integer	read-only	The maximum size, in bytes, of the buffer pool
MaxTotalRam	Long	read-only	The total amount of RAM used for this journal
MaxValueSize	Integer	read-only	The maximum allowable size, in bytes, for serialized values
NioRam	Boolean	read-only	Whether the RAM journal uses on-heap byte buffers or off-heap NIO buffers (NIO RAM). Valid values are true and false. A true value indicates off-heap NIO buffers. A false value indicates on-heap byte buffers. The default value is false.

Table A–14 (Cont.) JournalMBean Attributes for RamJournalRM

Attribute	Type	Access	Description
PoolSize	Integer	read-only	This attribute does not pertain to a RAM journal and returns -1.
TotalCompactionCount	Integer	read-only	The total number of times compaction (garbage collection) has been done for the journal
TotalDataSize	Long	read-only	The amount of data, in bytes, that is currently stored for this journal
TotalFileSize	Long	read-only	The total size of all journal files for this journal

Operations

The JournalMBean MBean has no operations.

ReporterMBean

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

The object name of the MBean is:

```
type=Reporter
```

Attributes

Table A–15 describes the attributes for `ReporterMBean`.

Table A–15 ReporterMBean Attributes

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

Operations

Table A–16 describes the operations for `ReporterMBean`.

Table A–16 ReporterMBean Operations

Operation	Parameters	Return Type	Description
`resetStatistics`	Not applicable	`void`	Reset the reporter statistics.
`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`).
`runTabularReport`	`String sReportFile`	`TabularData`	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`). The results are returned in a table format in a separate window. You can also directly enter the XML syntax of a report group or report file. If you enter the XML of a report group, the individual reports must be found on the classpath.

`runTabularGroupReport`	`String sReportName`
	
	`Map mapXmlReports`
		`TabularData`	Run a report group one time and pass in the individual reports' XML content. The results are returned in a table format in a separate window.

`start`	Not applicable	`void`	Start reporting
`stop`	Not applicable	`void`	Stop reporting

ServiceMBean

The `ServiceMBean` MBean represents a clustered service. A cluster member includes 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 change the behavior of a service in real time. In addition, the MBean contains operations to start and stop a service in real time.

The object name of the MBean is:

`type=Service,name=`*`service name`*`,nodeId=`*`cluster node id`*

Terminology

The terms *task* and *request* have unique definitions within Oracle Coherence. Understand the terms 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`	read-only	The number of backups for every cache storage
`BackupCountAfterWritebehind`	`Integer`	read-only	The number of members of the partitioned (distributed) cache service that retain backup data, which does not require write-behind. The data is not vulnerable to being lost even if the entire cluster is shut down.
`EventInterceptorInfo`	`String[]`	read-only	An array of statistics for live events processed by event interceptors. The statistics include: ■ `Interceptors` – a list of registered interceptors ■ `ExceptionCount` – the number of exceptions thrown from the interceptors since the last time the statistics were reset ■ `LastException` – a stack trace of the last exception thrown from the interceptors

Table A–17 (Cont.) ServiceMBean Attributes

Attribute	Type	Access	Description
JoinTime	Date	read-only	The date and time (in cluster time) that this member joined the service
OwnedPartitionsBackup	Integer	read-only	The number of partitions that this member backs up (responsible for the backup storage)
OwnedPartitionsPrimary	Integer	read-only	The number of partitions that this member owns (responsible for the primary storage)
PartitionsAll	Integer	read-only	The total number of partitions that every cache storage is divided into
PartitionsEndangered	Integer	read-only	The total number of partitions that are not currently backed up
PartitionsUnbalanced	Integer	read-only	The total number of primary and backup partitions that remain to be transferred until the partition distribution across the storage enabled service members is fully balanced
PartitionsVulnerable	Integer	read-only	The total number of partitions that are backed up on the same computer where the primary partition owner resides
QuorumStatus	String	read-only	The current state of the service quorum
RefreshTime	Date	read-only	The timestamp when this model was last retrieved from a corresponding member. For local servers, it is the local time.
RequestAverageDuration	Float	read-only	The average duration (in milliseconds) of an individual request that was issued by the service since the last time the statistics were reset
RequestMaxDuration	Long	read-only	The maximum duration (in milliseconds) of a request that was issued by the service since the last time the statistics were reset
RequestPendingCount	Long	read-only	The number of pending requests that were issued by the service
RequestPendingDuration	Long	read-only	The duration (in milliseconds) of the oldest pending request that was issued by the service
RequestTimeoutCount	Long	read-only	The total number of timed-out requests since the last time the statistics were reset
RequestTimeoutMillis	Long	read/write	The default timeout value in milliseconds for requests that can be timed-out (for example, implement the com.tangosol.net.PriorityTask interface) but do not explicitly specify the request timeout value
RequestTotalCount	Long	read-only	The total number of synchronous requests that were issued by the service since the last time the statistics were reset
Running	Boolean	read-only	Specifies whether the service is running
SeniorMemberId	Integer	read-only	The service senior member ID. The value is -1 if the service is not running.
Statistics	String	read-only	The statistics for this service in a human readable format

Table A–17 (Cont.) ServiceMBean Attributes

Attribute	Type	Access	Description
StatusHA	String	read-only	The High Availability (HA) status for this service. A value of MACHINE-SAFE indicates that all the cluster members running on any given computer could be stopped without data loss. A value of NODE-SAFE indicates that a cluster member could be stopped without data loss. A value of ENDANGERED indicates that abnormal termination of any cluster member that runs this service may cause data loss. A value of N/A indicates that the service has no high availability impact.
StorageEnabled	Boolean	read-only	Specifies whether the local storage is enabled for this cluster member
StorageEnabledCount	Integer	read-only	Specifies the total number of cluster members running this service for which local storage is enabled
TaskAverageDuration	Float	read-only	The average duration (in milliseconds) of an individual task execution
TaskBacklog	Integer	read-only	The size of the backlog queue that holds tasks scheduled to be executed by a service thread
TaskCount	Long	read-only	The total number of executed tasks since the last time the statistics were reset
TaskHungCount	Integer	read-only	The total number of currently executing hung tasks
TaskHungDuration	Long	read-only	The longest currently executing hung task duration in milliseconds
TaskHungTaskId	String	read-only	The ID of the longest currently executing hung task
TaskHungThresholdMillis	Long	read/write	The amount of time in milliseconds that a task can execute before it is considered hung. 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 ThreadCount value is > 0).
TaskMaxBacklog	Integer	read-only	The maximum size of the backlog queue since the last time the statistics were reset
TaskTimeoutCount	Integer	read-only	The total number of timed-out tasks since the last time the statistics were reset
TaskTimeoutMillis	Long	read/write	The default timeout value in milliseconds for tasks that can be timed-out (for example, implement the com.tangosol.net.PriorityTask 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 ThreadCount value is > 0).
ThreadAbandonedCount	Integer	read-only	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	read-only	The average number of active (not idle) threads in the service thread pool since the last time the statistics were reset

Table A–17 (Cont.) ServiceMBean Attributes

Attribute	Type	Access	Description
ThreadCount	Integer	read/write	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 > 0). To initially set this value, configure the thread-count for a service either in an operational override file or a cache configuration file. If the value is 0 (the default), then only the service thread is used and a thread pool is never started. **Note:** For a proxy service, setting a thread count value disables dynamic thread pool sizing. A dynamic thread pool automatically adjusts the number of threads based on proxy service load.
ThreadCountMax	Integer	read/write	The maximum number of threads that are allowed for a proxy service when dynamic thread pool sizing is enabled. A value of -1 indicates that dynamic thread pool sizing is disabled.
ThreadCountMin	Integer	read/write	The minimum number of threads that are allowed for a proxy service when dynamic thread pool sizing is enabled. A value of -1 indicates that dynamic thread pool sizing is disabled.
ThreadCountUpdateTime	Date	read-only	The last time an update was made to the ThreadCount attribute. This attribute is only valid when the ThreadPoolSizingEnabled attribute is set to true.
ThreadIdleCount	Integer	read-only	The number of currently idle threads in the service thread pool
ThreadPoolSizingEnabled	Boolean	read-only	Specifies whether dynamic thread pool sizing is enabled for this service. Only proxy services support dynamic thread pools. To enable dynamic thread pools, no value must be specified for the thread-count attribute.
TransportAddress	String	read-only	The service-dedicated transport address. If an address is shown (indicating that a reliable transport has been enabled on the service), then the service instance communicates with other service members using the dedicated transport address rather then using the shared cluster transport.
TransportBackloggedConnectionList	String[]	read-only	A list of backlogged connections on the service-dedicated transport
TransportBackloggedConnections	Integer	read-only	The number of backlogged connections on the service-dedicated transport. Any new requests that require the connection are blocked until the backlog is cleared.
TransportConnections	Integer	read-only	The number of maintained connections on the service-dedicated transport. This count may be lower than the member count if some members have not been configured to use the dedicated transport, or it has been identified that there is no advantage in using the dedicated transport for communication with certain members.
TransportReceivedBytes	Long	read-only	The number of bytes that were received by the service-dedicated transport since the last time the statistics were reset

Table A–17 (Cont.) ServiceMBean Attributes

Attribute	Type	Access	Description
TransportReceivedMessages	Long	read-only	The number of messages that were received by the service-dedicated transport since the last time the statistics were reset
TransportRetainedBytes	Long	read-only	The number of bytes that were retained by the service-dedicated transport and that are awaiting delivery acknowledgment. This memory is allocated outside of the Java garbage collection heap space.
TransportSentBytes	Long	read-only	The number of bytes that were sent by the service-dedicated transport since the last time the statistics were reset
TransportSentMessages	Long	read-only	The number of messages that were sent by the service-dedicated transport since the last time the statistics were reset
Type	String	read-only	The type identifier of the service

Operations

Table A–18 describes the operations for ServiceMBean.

Table A–18 ServiceMBean Operations

Operation	Parameters	Return Type	Description
reportOwnership	fVerbose	String	Reports the partitions that are owned by the service on this node. The fVerbose parameter formats the ownership summary. Valid values are true or false. Setting the parameter to true includes the detailed ownership catalog.
resetStatistics	Not applicable	void	Reset the service statistics
shutdown	Not applicable	void	Stop the service. This is a controlled shutdown, and is preferred to the stop operation.
start	Not applicable	void	Start the service
stop	Not applicable	void	Force the service to stop. Use the shutdown operation 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 includes 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 object name of the MBean is:

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

Attributes

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

Table A–19 StorageManagerMBean Attributes

Attribute	Type	Access	Description
`EventInterceptorInfo`	`String[]`	read-only	An array of statistics for live events processed by event interceptors. The statistics include: ■ `Interceptors` – a list of registered interceptors ■ `ExceptionCount` – the number of exceptions thrown from the interceptors since the last time the statistics were reset ■ `LastException` – a stack trace of the last exception thrown from the interceptors
`EventsDispatched`	`Long`	read-only	The total number of events that were dispatched by the storage manager since the last time the statistics were reset
`EvictionCount`	`Long`	read-only	The number of evictions, from the backing map that is managed by this storage manager, that were caused by entry expiry or insert operations that would make the underlying backing map reach its configured size limit. The eviction count is used to audit the cache size in a static system: *`Cache Size = Insert Count - Remove Count - Eviction Count`* Therefore, the eviction count is not reset by the reset statistics method.
`IndexInfo`	`String[]`	read-only	An array of information for each index that is applied to the portion of the partitioned cache managed by the storage manager. Each element is a string value that includes a `ValueExtractor` description, ordered flag (`true` to indicate that the contents of the index are ordered; `false` otherwise), and cardinality (number of unique values indexed).

Table A–19 (Cont.) StorageManagerMBean Attributes

Attribute	Type	Access	Description
InsertCount	Long	read-only	The number of inserts into the backing map. In addition to standard inserts that are caused by put and invoke operations or synthetic inserts that are caused by get operations with read-through backing map topology, this counter increments when distribution transfers move resources into the underlying backing map and decrements when distribution transfers move data out. The insert count is used to audit the cache size in a static system: *Cache Size = Insert Count - Remove Count - Eviction Count* Therefore, the insert count is not reset by the reset statistics method.
ListenerFilterCount	Integer	read-only	The number of filter-based listeners that is currently registered with the storage manager
ListenerKeyCount	Integer	read-only	The number of key-based listeners that is currently registered with the storage manager
ListenerRegistrations	Long	read-only	The total number of listener registration requests that were processed by the storage manager since the last time the statistics were reset
LocksGranted	Integer	read-only	The number of locks that is currently granted for the portion of the partitioned cache managed by the storage manager.
LocksPending	Integer	read-only	The number of pending lock requests for the portion of the partitioned cache managed by the storage manager
MaxQueryDescription	String	read-only	A description of the query with the longest duration that exceeds the MaxQueryThresholdMillis attribute since the statistics were last reset
MaxQueryDurationMillis	Long	read-only	The number of milliseconds of the longest running query since the statistics were last reset
MaxQueryThresholdMillis	Long	read/write	A threshold, in milliseconds, for recording queries. The longest query that executes longer than this threshold is reported by the MaxQueryDescription attribute. The default value is 30 ms.
NonOptimizedQueryAverageMillis	Long	read-only	The average duration, in milliseconds, for non-optimized query execution since the cache statistics were last reset
NonOptimizedQueryCount	Long	read-only	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	read-only	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	read-only	The average duration, in milliseconds, per optimized query execution since the statistics were last reset
OptimizedQueryCount	Long	read-only	The total number of queries that were fully resolved using indexes since the statistics were last reset

Table A–19 (Cont.) StorageManagerMBean Attributes

Attribute	Type	Access	Description
OptimizedQueryTotalMillis	Long	read-only	The total execution time, in milliseconds, for queries that were fully resolved using indexes since the statistics were last reset
RefreshTime	Date	read-only	The timestamp when this model was last retrieved from a corresponding member. For local servers, it is the local time.
RemoveCount	Long	read-only	The number of removes from the backing map managed by this storage manager caused by operations such as clear, remove, or invoke. The remove count is used to audit the cache size in a static system: *Cache Size = Insert Count - Remove Count - Eviction Count* Therefore, the remove count is not reset by the reset statistics method.
TriggerInfo	String[]	read-only	An array of information for each trigger that is 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 MapTrigger implementation.

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 includes zero or more instances of this managed bean depending on the number of configured transactional caches. The MBean provides global transaction manager statics by aggregating service-level statistics from all transaction service instances. Each cluster member has one instance of the transaction manager MBean for each service.

The object name of the MBean is:

`type=TransactionManager,service=`*`service name`*`,nodeId=`*`cluster node id`*

Note: For certain transaction manager attributes, the coordinator member for the transaction maintains the count even though multiple members participate in the transaction. For example, a transaction may include modifications to entries stored on multiple members, but the `TotalCommitted` attribute only increments 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
`CommitTotalMillis`	`Long`	read-only	The cumulative time (in milliseconds) that was spent during the commit phase since the last time statistics were reset
`RefreshTime`	`Date`	read-only	The timestamp when this model was last retrieved from a corresponding member. For local servers, it is the local time.
`TimeoutMillis`	`Long`	read-only	The transaction timeout value in milliseconds. This value only applies to transactional connections obtained after the value is set. This attribute is currently not supported.
`TotalActive`	`Long`	read-only	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. The coordinator member for this transaction maintains the count even though multiple members may have participated in the transaction.
`TotalCommitted`	`Long`	read-only	The total number of transactions that have been committed by the transaction manager since the last time the statistics were reset. The coordinator member for this transaction maintains the count even though multiple members may have participated in the transaction.

Table A–20 (Cont.) TransactionManagerMBean Attributes

Attribute	Type	Access	Description
TotalRecovered	Long	read-only	The total number of transactions that have been recovered by the transaction manager since the last time the statistics were reset. The coordinator member for this transaction maintains the count even though multiple members may have participated in the transaction.
TotalRolledback	Long	read-only	The total number of transactions that have been rolled back by the transaction manager since the last time the statistics were reset. The coordinator member for this transaction maintains the count even though multiple members may have participated in the transaction.
TotalTransactionMillis	Long	read-only	The cumulative time (in milliseconds) that was spent on active transactions

Operations

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

B

Report File Configuration Elements

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 Elements," for a detailed reference of the report group deployment descriptor elements.

This appendix includes the following sections:

- 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 MBeans. The `coherence.jar/reports` directory contains many predefined report files. Modify the reports or create new reports as required. See Chapter 6, "Analyzing Report Contents," for additional information about the predefined 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 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 that is located in the `coherence.jar` library is always used at run time even if the `xsi:schemaLocation` attribute references the Web URL.
- Omit the `xsi:schemaLocation` attribute to disable schema validation.
- When deploying Oracle 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 nonterminal report file deployment descriptor elements.

Table B–1 Report File Deployment Descriptor Nonterminal Elements

Element	Used in
column	row
filter	filters
filters	report
params	filter, column
query	report, column
report	report-config
report-config	*root element*
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` subelement.

Elements

Table B–2 describes the subelements of the `column` element.

Table B–2 column Subelements

Element	Required/ Optional	Description
`type`	Optional	Specifies an `XmlColumn` implementation. The `type` element defines the type of the column in the report. The following values are valid: ■ `attribute` – (default) A column that contains the data from an MBean attribute. ■ `key` – A column that contains the value from an MBean key attribute. ■ `method` – A column that contains the result of an MBean method invocation. ■ `function` – A column that contains the result of a function or aggregation. See the `function-name` element in this table. ■ `global` – A column that contains a value not related to any specific MBean. Global column names are `{report-time}`, `{report-count}`, and `{node-id}`. ■ `constant` – A column that contains a constant string of numeric value. ■ `property` – A column that contains a value of a Java system property.
`name`	Optional	Specifies an attribute or method name on the MBean. For composite data types, the `name` element can contain a slash (`/`) delimited name sequence.
`header`	Optional	Specifies the column header. The value of the `name` element is used if this item is omitted.
`delim`	Optional	Specifies a character that separates column or array values. Valid values are `{tab}`, `{space}`, or any non-whitespace character.
`query`	Optional	Specifies information necessary to construct a JMX query to find all MBeans contributing to the report or column.
`hidden`	Optional	Specifies whether the column value is hidden in the report. Valid values are `true` and `false`.
`column-ref`	Optional	Specifies a reference to a column identifier. The reference passes a column value as an argument to a filter or another column.

Table B–2 (Cont.) column Subelements

Element	Required/ Optional	Description
function-name	Optional	Specifies the name of the calculation to apply to the associated MBean attribute values. This element is only valid when the type element is function. The following values are valid: ■ sum – The sum of all retrieved attribute values ■ avg – The average value for all retrieved attribute values ■ min – The minimum numeric value for all retrieved attribute values ■ max – The maximum numeric value for all retrieved attribute values ■ add – The sum of values for two column references ■ subtract – The difference between values for two column references ■ multiply – The product of values for two column references ■ divide – The ratio between values for two column references
params	Optional	Specifies an argument column or a filter reference
data-type	Optional	Specifies the data type of a constant column. Valid values are double and string.
value	Optional	Specifies the value of a constant column
group-by	Optional	specifies whether the column is included in the group by clause of the query. Valid values are true and false. The default values is false.
subquery	Optional	Specifies whether the column is included as part of a subquery. Valid values are true and false. The default values is false.
return-neg	Optional	Specifies if a negative value is returned to the query. Negative values are considered error or not available codes on Oracle Coherence MBeans. By default, these codes return zero to not affect column calculations. Valid values are true and false. The default values is false.

filter

Used in: filters

Description

The `filter` element defines a filter to use in the report. The filter element supports the use of an `id` attribute to uniquely identify the filter. Use the ID when referring to a filter with the `filter-ref` element. Filters can be referenced from within a `params` element and a `query` element.

Elements

Table B–3 describes the subelements of the `filter` element.

Table B–3 filter Subelements

Element	Required/ Optional	Description
`type`	Optional	Specifies an `XmlFilter` implementation. The `type` element defines the type of the filter in the report. The following values are valid: ■ `equals` – A filter that compares the result of two or more column references for a value equality. ■ `greater` – A filter that compares the results of two column references for the `greater than` condition. If any of the values is resolved to `null`, the evaluation yields `false`. (This approach is equivalent to the way the `NULL` values are handled by SQL.) ■ `less` – A filter that compares the results of two column references for the `less than` condition. If any of the values is resolved to `null`, the evaluation yields `false`. ■ `not` – A filter that returns the logical `not` of a filter reference. ■ `and` – A filter that returns the logical `and` of two filter references. ■ `or` – A filter that returns the logical `or` of two filter references.
`params`	Optional	Specifies an argument column or a filter reference

filters

Used in: report

Description

The `filters` element contains any number of `filter` elements.

Elements

Table B–4 describes the subelements of the `filters` element.

Table B–4 filters Subelements

Element	Required/ Optional	Description
`filter`	Optional	Defines a filter to use in the report

params

Used in: column, filter, query

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
`filter-ref`	Required	Specifies a reference to a filter ID. The reference passes a filter as an argument to a query or another filter.
`column-ref`	Required	Specifies a reference to a column identifier. The reference passes 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. Enter queries within a `<pattern>` element.

Examples

The following example includes all node MBeans in the report or column:

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

The following 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 in the query pattern, define a `reporter.cacheName` system property 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. Use the `key` attribute to retrieve an attribute from a related (joined) MBean. For example, because the `Coherence:type=Service,*` pattern is known to result in MBeans in the `Coherence:type=Service,nodeId=NNN` format, the following configuration below prints 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 include in the report
`filter-ref`	Optional	Specifies a reference to a filter ID. The reference passes 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. A report can include any number of `report` elements; however, a report file typically contains a single report definition.

Elements

Table B–7 describes the subelements of the `report` element.

Table B–7 report Subelements

Element	Required/ Optional	Description
`description`	Optional	Specifies a descriptive heading for the associated report or column.
`file-name`	Required	Specifies the file name for the generated report. The file name is either absolute or relative to a directory that is 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. The file name may contain three macros: ▪ `{batch}` – This macro is replaced with a counter (a sequential number). ▪ `{node}` – This macro is replaced with the cluster member ID. Use this macro to differentiate reports on different cluster members. ▪ `{date}` – This macro is replaced with the current date (`YYYYMMDD`). **Note**: A process running the reporter requires read, write, and create access to the report output directory.
`delim`	Optional	Specifies a character that separates column or array values. Valid values are `{tab}`, `{space}`, or any non-whitespace character.
`hide-headers`	Optional	Specifies whether the report includes description and column headers. Valid values are `true` and `false`.
`filters`	Optional	Specifies a group of filters for the report
`query`	Required	Specifies information necessary to construct a JMX query to find all MBeans contributing to the report or column
`row`	Required	Specifies information to generate a report row

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
`report`	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 include in the report.

Elements

Table B–9 describes the subelements of the `row` element.

Table B–9 row Subelements

Element	Required/ Optional	Description
`column`	Required	Specifies the information that generates a report column

C

Report Group Configuration Elements

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 Elements," for a detailed reference of the report file deployment descriptor elements that create individual report files.

This appendix includes the following sections:

- "Report Group Configuration Deployment Descriptor"
- "Element Reference"

Report Group Configuration Deployment Descriptor

The report group deployment descriptor specifies any number of individual reports to create when reporting is enabled. The `coherence.jar/reports` directory contains two predefined descriptors. The `report-group.xml` descriptor is the default descriptor and includes a subset of the predefined reports. The `report-all.xml` descriptor includes all the predefined reports. Modify the predefined report group deployment descriptors or create new report group deployment descriptors as required. The name and location of which report group deployment descriptor to use at run time is configured in the operational deployment descriptor and is overridden in an operational override file. See Chapter 4, "Using Oracle Coherence Reporting," for detailed information.

The report group 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-config/1.0/coherence-report-group-config.xsd`

The `<report-group>` element is the root element of the descriptor and typically includes the XSD reference, the 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 that is located in the `coherence.jar` library is always used at run time even if the `xsi:schemaLocation` attribute references the Web URL.
- Omit the `xsi:schemaLocation` attribute to disable schema validation.
- When deploying Oracle 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 nonterminal report group deployment descriptor elements.

Table C–1 Report Group Deployment Descriptor Nonterminal Elements

Element	Used in
`init-param`	`init-params`
`init-params`	`report-config`
`report-config`	`report-group`
`report-group`	*root element*
`report-list`	`report-group`

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
`param-name`	Optional	Specifies the name of the initialization parameter. This element cannot be used with the `param-type` element.
`param-type`	Optional	Specifies the Java type of the initialization parameter. The following types are supported: ■ `string` – Indicates that the value is a `java.lang.String` ■ `long` – Indicates that the value is a `java.lang.Long` ■ `double` – Indicates that the value is a `java.lang.Double` This element cannot be used with the `para-name` element.
`param-value`	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–3 describes the subelements of the `init-params` element.

Table C–3 init-params Subelements

Element	Required/ Optional	Description
`init-param`	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
`location`	Required	Specifies a path to a report configuration descriptor that conforms to the `coherence-report-config.xsd` file. This path is either a file or a URL.
`init-params`	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
`frequency`	Required	Specifies how often a report batch refreshes. Enter the value in either seconds seconds (`s`) or minutes (`m`). For example, a value of `10s` refreshes the reports every 10 seconds; a value of `5m` refreshes the reports every 5 minutes. Selecting an appropriate frequency is important: if the frequency is too short, the report contains too much data and consumes significant disk space; if the frequency is too long, the report does not contain enough information. In addition, decide on a process for purging and archiving historical information before you enable reporting.
`output-directory`	Optional	The directory path to prepend to the output file names from the report configuration files. The cluster member must have read/write access to this path. The `<output-directory>` element supports an optional `system-property` attribute. The attribute value is a user-defined name that can be used at runtime to override the configured output location. The preconfigured system property override that is used in the predefined report group configuration files is `tangosol.coherence.reporter.output.directory`.
`report-list`	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
`report-config`	Required	Specifies the location of the report configuration file and corresponding initialization parameters