Oracle® Coherence

Managing Oracle Coherence

14.1.1.2206
F44662-15
October 2024

ORACLE

Oracle Coherence Managing Oracle Coherence, 14.1.1.2206

F44662-15

Primary Author: Oracle Corporation

Contents

5 Using the Health Check API

6 Using Oracle Coherence Reporting

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.

This preface includes the following sections:

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 `https://www.oracle.com/corporate/accessibility/`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `https://support.oracle.com/portal/` or visit `Oracle Accessibility Learning and Support` if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve.

Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

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*
- *Installing Oracle Coherence*
- *Integrating Oracle Coherence*
- *Securing 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.

1

Introduction to Oracle Coherence Management

Oracle Coherence management is critical to the success of developing and deploying Coherence solutions. Understanding how Coherence management is implemented and what capabilities it offers is an important first step when learning how to manage a Coherence solution.
Coherence management is implemented using Java Management Extensions (JMX). JMX is a Java standard for managing and monitoring Java applications and services. If you are new to JMX, see Introduction to JMX Technology in *Java Management Extensions (JMX) Technology Overview*.

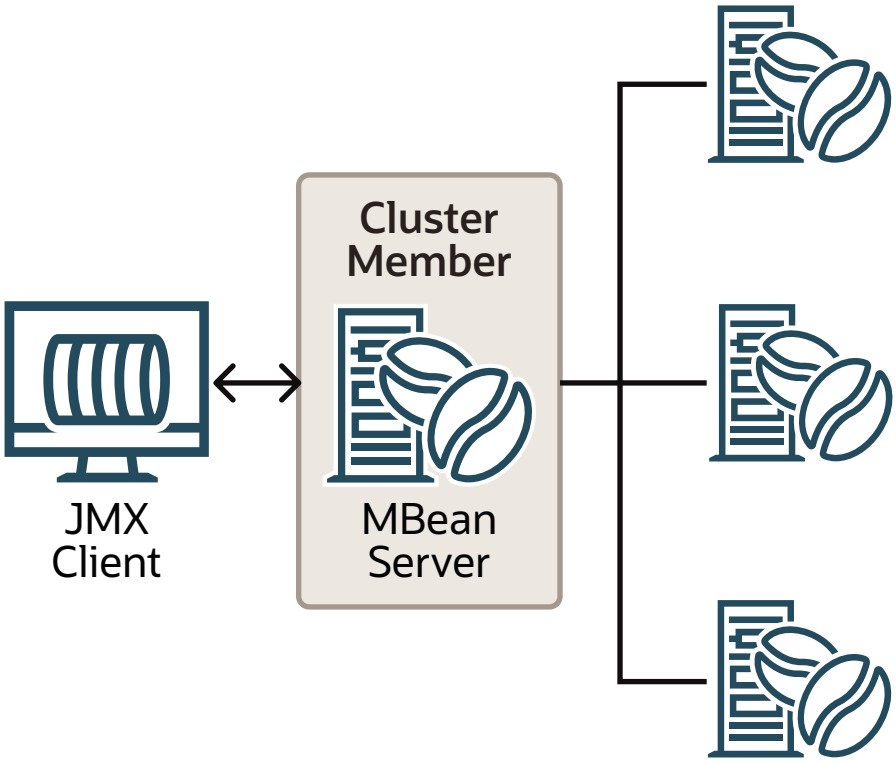
This chapter includes the following sections:

- Conceptual Overview of Oracle Coherence Management
- Management Reporting in Oracle Coherence
- Overview of Management Configuration
- Managing Oracle Coherence with Oracle WebLogic Server
  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.
- Managing Oracle Coherence with Fusion Middleware Control
- 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.
- Managing Oracle Coherence with Oracle Command-Line Interface
  The Oracle Coherence command-line interface (CLI), `cohctl`, is a lightweight tool using that can be scripted or used interactively to manage and monitor Coherence clusters.

Conceptual Overview of Oracle Coherence Management

Coherence management uses JMX and includes managed objects for many Coherence components. You can use a JMX compliant client to interact with the managed objects.Coherence includes a JMX framework that aggregates the managed objects of all cluster members to one or more MBean servers. The management framework allows access to management information from any cluster member and is fault tolerant should a JMX member fail. The management framework is disabled by default and must be explicitly enabled by configuring at least one cluster member to host an MBean server. See Using JMX to Manage Oracle Coherence .
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



This section includes the following topics:

- Oracle Coherence MBeans
- Custom MBeans
- MBean Consoles
- Coherence VisualVM Plug-In

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.

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 Registering Custom MBeans.

MBean Consoles

Any MBean-capable console can interact with Oracle Coherence MBeans. The 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.

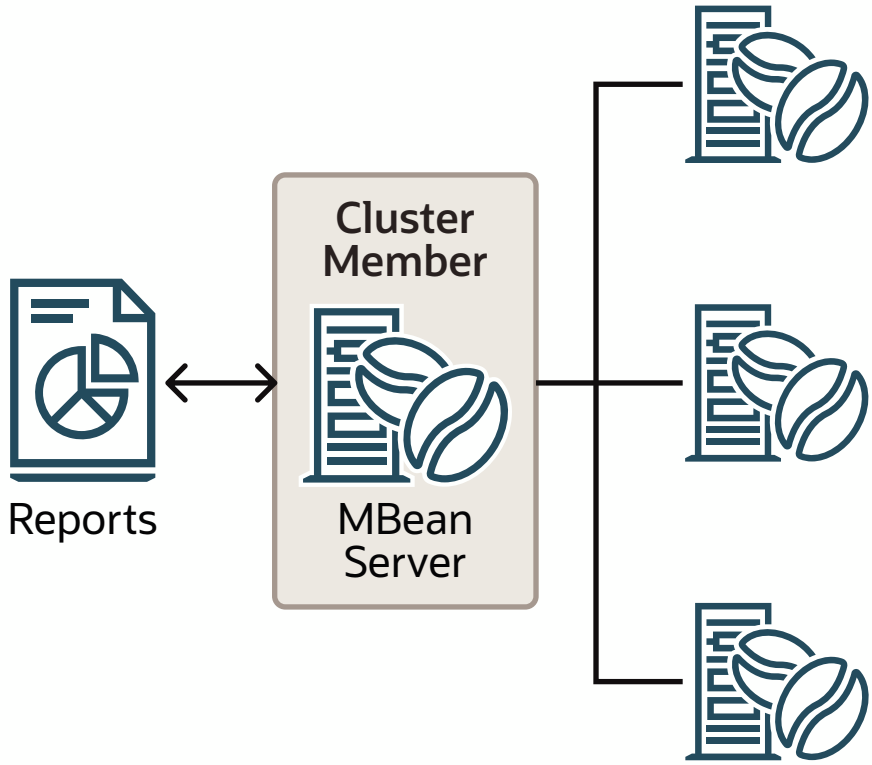
Coherence VisualVM Plug-In

The Coherence VisualVM plug-in aggregates Coherence MBean data and shows a concise operational view of a Coherence cluster within the VisualVM console. Management information is presented over time, which allows real-time analysis and troubleshooting. See Using the Coherence VisualVM Plug-In.

Management Reporting in Oracle Coherence

Management reports display management information over time and provide a historical context that is not possible by simply monitoring Coherence MBeans.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 to 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. See Analyzing Report Contents. The predefined 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 Using Oracle Coherence Reporting.

Overview of Management Configuration

Coherence management and reporting require the use of multiple configuration files. The configuration files enable, control, and customize management features as required. See Understanding Configuration in *Developing Applications with Oracle Coherence*.
The following files are used to configure management:

- 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 management-config in *Developing Applications with Oracle Coherence*.
- 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 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 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. See invocation-scheme in *Developing Applications with Oracle Coherence*.

To learn more about Coherence configuration, see Understanding Configuration in *Developing Applications with Oracle Coherence*.

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 benefits include the ability to:

- 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 Deploying Coherence Applications to WebLogic Server in *Administering Oracle Coherence*.

- Creating a Web Application for Coherence Management Over REST

Creating a Web Application for Coherence Management Over REST

In a large Coherence cluster with many members, you may encounter performance issues when querying Coherence MBeans from the Administration Server REST end point. To avoid such issues, you can directly query the Coherence management over REST API by deploying a web application to a Coherence Managed Server that has an MBean proxy server running.

Here is an example on how to build the web application for Coherence management over REST:

1. Create a directory for the web application:

```
mkdir CohManagement
```

2. Go to the web application directory, and then create `META-INF` and `WEB-INF` directories.
3. Add the following `web.xml` and `weblogic.xml` to `WEB-INF`. You may rename the <`context-root`>, if required. In this example, `cohmgmt` is the <`context-root`>.

`web.xml`

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
         http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
         version="3.1">

  <display-name>Coherence Management REST Resources</display-name>
  <description>Coherence Management REST Resources</description>

  <servlet>
    <servlet-name>coherence.management.rest.Application</servlet-name>

<servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-
class>
    <init-param>
    <param-name>javax.ws.rs.Application</param-name>

<param-
value>com.tangosol.coherence.management.internal.ManagementResourceConfig</
param-value>
    </init-param>
  </servlet>

  <servlet-mapping>
    <servlet-name>coherence.management.rest.Application</servlet-name>
    <url-pattern>/*</url-pattern>
  </servlet-mapping>
</web-app>
```

`weblogic.xml`

```
<?xml version="1.0" encoding="UTF-8"?>

<weblogic-web-app
  xmlns="http://xmlns.oracle.com/weblogic/weblogic-web-app"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.oracle.com/weblogic/weblogic-web-app">

  <context-root>cohmgmt</context-root>
```

```
</weblogic-web-app>
```

4. Build the web application `.war` file. Go to the parent directory of the web application and use the following command to create the `.war` file:

```
jar cvf CohManagement.war -C CohManagement .
```

5. Deploy the `.war` file to a managed Coherence server that has an MBean proxy server running.

To query the Coherence MBeans through the web application, use the managed Coherence server's (WebLogic server) listening port.

For example, if the WebLogic server is listening on port 7010, the URL to query the stats for all the caches will be:

```
http://<Host>:7010/cohmgmt/management/coherence/cluster/caches
```

Managing Oracle Coherence with Fusion Middleware Control

Fusion Middleware Control is a Web browser-based, graphical user interface that manages and monitors Fusion Middleware domains and components. In addition to native management functionality, Fusion Middleware Control also supports many WebLogic Server Administration Console features. See Getting Started Using Oracle Enterprise Manager Fusion Middleware Control in *Administering Oracle Fusion Middleware*. Fusion Middleware Control includes support for managing both managed Coherence servers and standalone Coherence servers and is ideal for managing Coherence together with other Fusion Middleware components from a single console.

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 at Oracle Enterprise Manager Downloads. See Getting Started with Management Pack for Oracle Coherence in *Enterprise Manager Cloud Control Oracle Fusion Middleware Management Guide*.

Managing Oracle Coherence with Oracle Command-Line Interface

The Oracle Coherence command-line interface (CLI), `cohctl`, is a lightweight tool using that can be scripted or used interactively to manage and monitor Coherence clusters.

The CLI accesses clusters using the HTTP Management over REST interface and therefore requires this to be enabled on clusters you want to monitor or manage.

For more information, see the GitHub repository and the official documentation at Docs - Coherence CLI Introduction.

2

Using JMX to Manage Oracle Coherence

JMX management is enabled and configured using cluster operational settings. Once configured, you can access Oracle Coherence MBeans using tools such as VisualVM and JConsole. Management information is also accessible using REST.
This chapter includes the following sections:

- Configuring JMX Management
- Accessing Oracle Coherence MBeans
- Accessing Management Information Using REST
  You can use HTTP to access Coherence MBeans that are exposed as REST resources. REST enables a broad range of clients to access management information.
- Using the Coherence VisualVM Plug-In

Configuring JMX Management

Coherence includes many configuration options, such as enabling management, stopping cluster members from being managed, filtering MBeans, and configuring management refresh properties. 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 management-config in *Developing Applications with Oracle Coherence*.
This section includes the following topics:

- Using Dynamic Management Mode
- Explicitly 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

Using Dynamic Management Mode

Coherence is configured by default to start in dynamic management mode. Dynamic management mode is a high availability feature that automatically selects the senior cluster member as the JMX cluster member. The JMX cluster member hosts an MBean server that is responsible for aggregating management information from all cluster members. If the JMX cluster member is not operational, then the next most senior cluster member is automatically selected as the JMX cluster member.

You can find the JMX service URL by querying the `NameService` service on any cluster member using the `com.tangosol.discovery.NSLookup` class. The class has a command line interface or you can call the `lookupJMXServiceURL` method. For example:

```
java -cp %COHERENCE_HOME%\lib\coherence.jar com.tangosol.discovery.NSLookup -
name management/JMXServiceURL

Cluster JRCluster:      service:jmx:rmi://127.0.0.1:62427/stub/
rO0ABXNyAC5qYXZheC5tYW5hZ2VtZW50LnJlbW90ZS5ybWkuUk1...
```

All cluster members can potentially be selected as the JMX cluster member. To specify that a cluster member should be excluded from being selected as the JMX 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="coherence.management">none
      </managed-nodes>
   </management-config>
</coherence>
```

The default value if no value is specified is `dynamic` and indicates that a cluster member can become the JMX cluster member.

The `coherence.management` system property also excludes a cluster member from being selected as the JMX cluster member. For example:

```
-Dcoherence.management=none
```

Explicitly Enabling Remote JMX Management on a Cluster Member

One or more cluster members can be explicitly configured to 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.

> **Note:**
>
> By default, Coherence is configured to use dynamic management mode and a JMX cluster member is automatically selected. Dynamic management is a recommended best practice. See Using Dynamic Management Mode. These instructions allow you to explicitly configure JMX cluster members. However, cluster members that are not configured as the JMX cluster member must be explicitly disabled from being dynamically selected as the JMX cluster member. See Disabling JMX Management.

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="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 `coherence.management` system property also enables remote JMX management. For example:

```
-Dcoherence.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="coherence.management">local-only
      </managed-nodes>
   </management-config>
</coherence>
```

The `coherence.management` system property also enables local JMX management. For example:

```
-Dcoherence.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. The argument is not required when using dynamic management, which is the default management configuration.

```
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:

```
-Dcoherence.management=all
-Dcoherence.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="coherence.management.remote">false
      </allow-remote-management>
   </management-config>
</coherence>
```

The `coherence.management.remote` system property also disables remote management. For example:

```
-Dcoherence.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="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="coherence.management">none</managed-nodes>
      <allow-remote-management
         system-property="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="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 `coherence.management.exclude` system property also filters MBeans. For example:

```
-Dcoherence.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="coherence.management.refresh.policy">
         refresh-ahead</refresh-policy>
      <refresh-expiry
         system-property="coherence.management.refresh.expiry">1s
      </refresh-expiry>
      <refresh-timeout
         system-property="coherence.management.refresh.timeout">300ms
      </refresh-timeout>
   </management-config>
</coherence>
```

This section includes the following topics:

- Setting the Management Refresh Expiry
- Setting the Management Refresh Policy
- Setting the Management Refresh Timeout

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

```
-Dcoherence.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 `coherence.management.refresh.policy` system property also sets the policy. For example:

```
-Dcoherence.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 `coherence.management.refresh.timeout` system property also sets the timeout. For example:

```
-Dcoherence.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="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="coherence.management">all
      </managed-nodes>
      <default-domain-name>MyDomainName</default-domain-name>
      <server-factory>
         <class-name
            system-property="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 `coherence.management.serverfactory` system property also specifies the implementation class. For example:

```
-Dcoherence.management.serverfactory=package.MyMBeanServerFinder
```

Accessing Oracle Coherence MBeans

Coherence supports many tools for interacting with Coherence MBeans, such as VisualVM, JConsole, the HTML Adapter, and an MBean connector. JMX management must be enabled on a cluster member before you can access Oracle Coherence MBeans. See Configuring JMX Management.
This section includes the following topics:

- Accessing MBeans Locally Using VisualVM
- Accessing MBeans Locally Using the JConsole Utility
- Allowing Remote Access to Oracle Coherence MBeans
- Accessing MBeans of a Running Coherence Cluster Using the JConsole Utility
- Accessing MBeans Using the HTML Adapter Application
- Accessing Coherence MBeans by Using WLST
  When you run Coherence within WebLogic Server in a managed Coherence Servers environment, WebLogic Server domain runtime MBean server collects JMX information from the management proxy. You can access this information by using WLST.
- Setting Up the Oracle Coherence MBean Connector

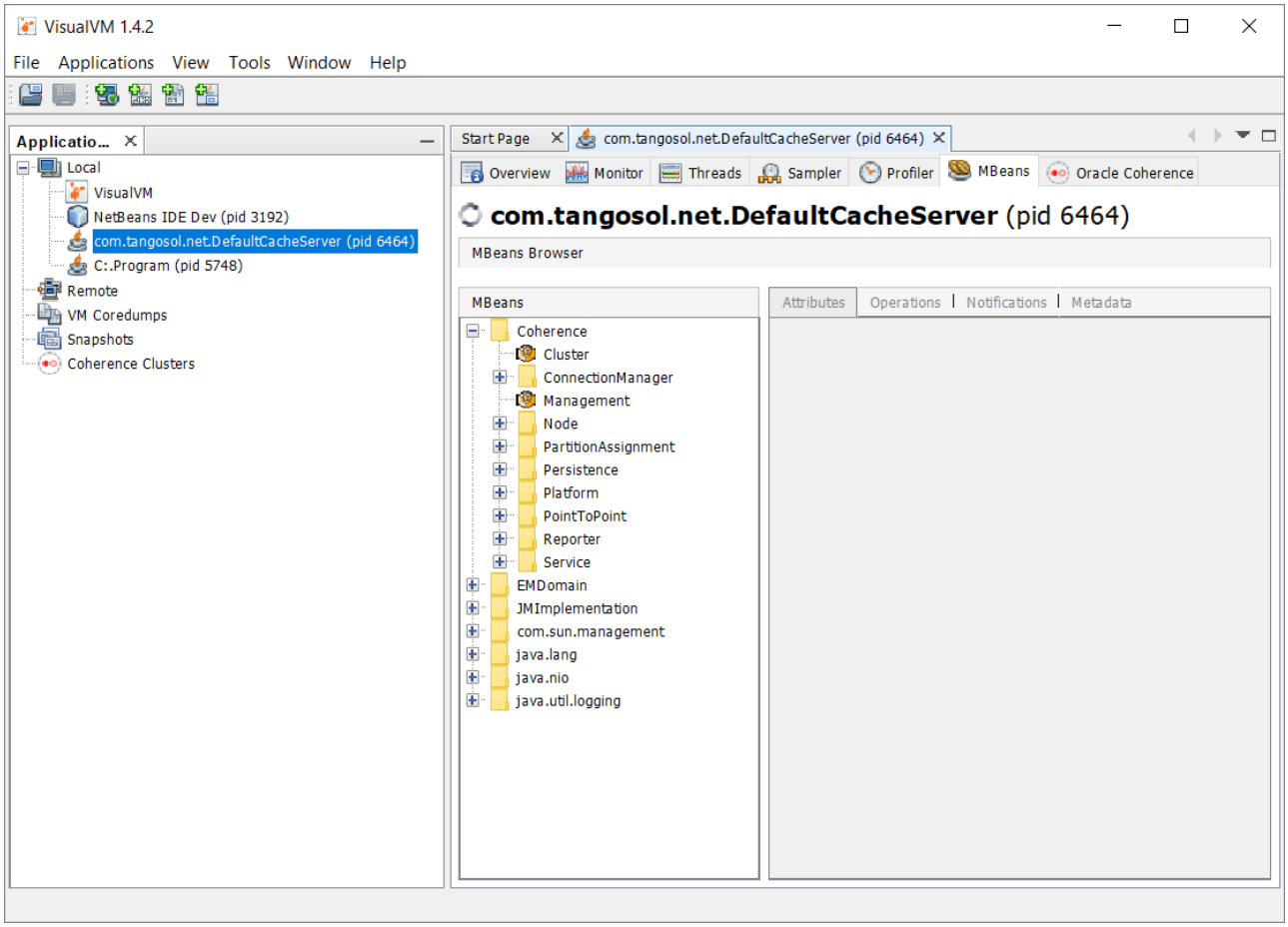
Accessing MBeans Locally Using VisualVM

VisualVM is a management utility that provides the ability to view and interact with MBeans. MBean functionality is provided as a plug-in to VisualVM. The plug-in must be installed from the 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 VisualVM utility:

1. Start a cache server using the *COHERENCE_HOME*\bin\cache-server script. For example:

   COHERENCE_HOME\bin\cache-server
2. Start *VISUALVM_HOME*\bin\visualvm. The 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

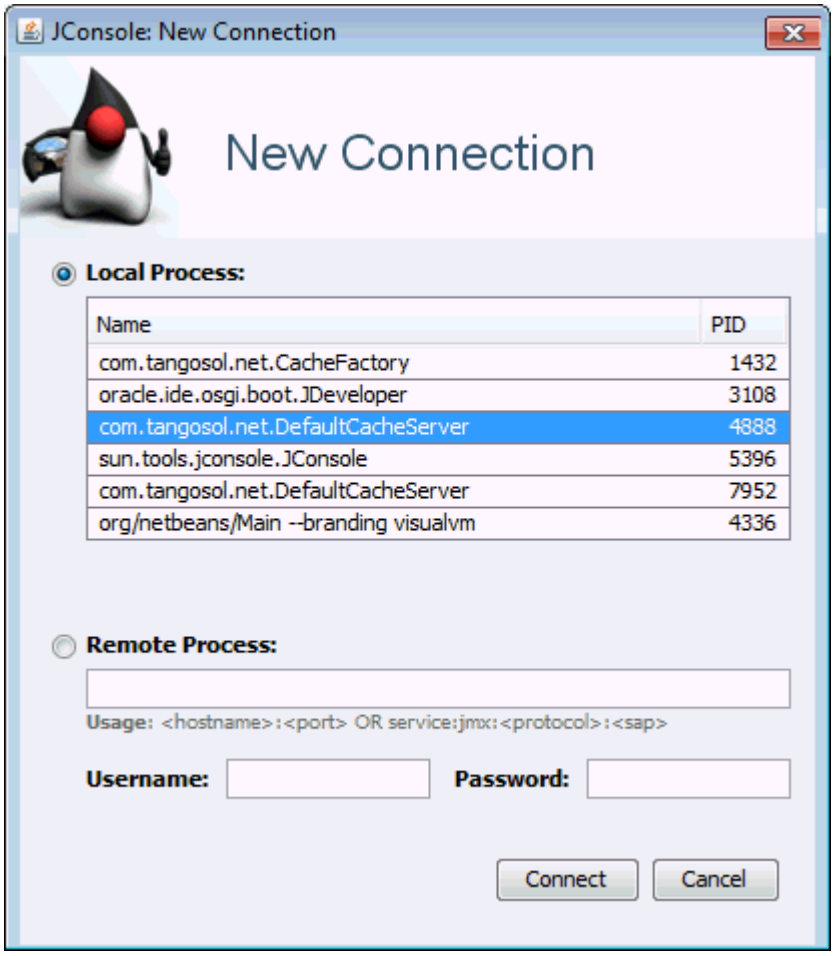
JConsole is a management utility that is included with the JDK (*JDK_HOME*\bin\jconsole) and provides the ability to view and interact with MBeans.

To locally access MBeans using the JConsole utility:

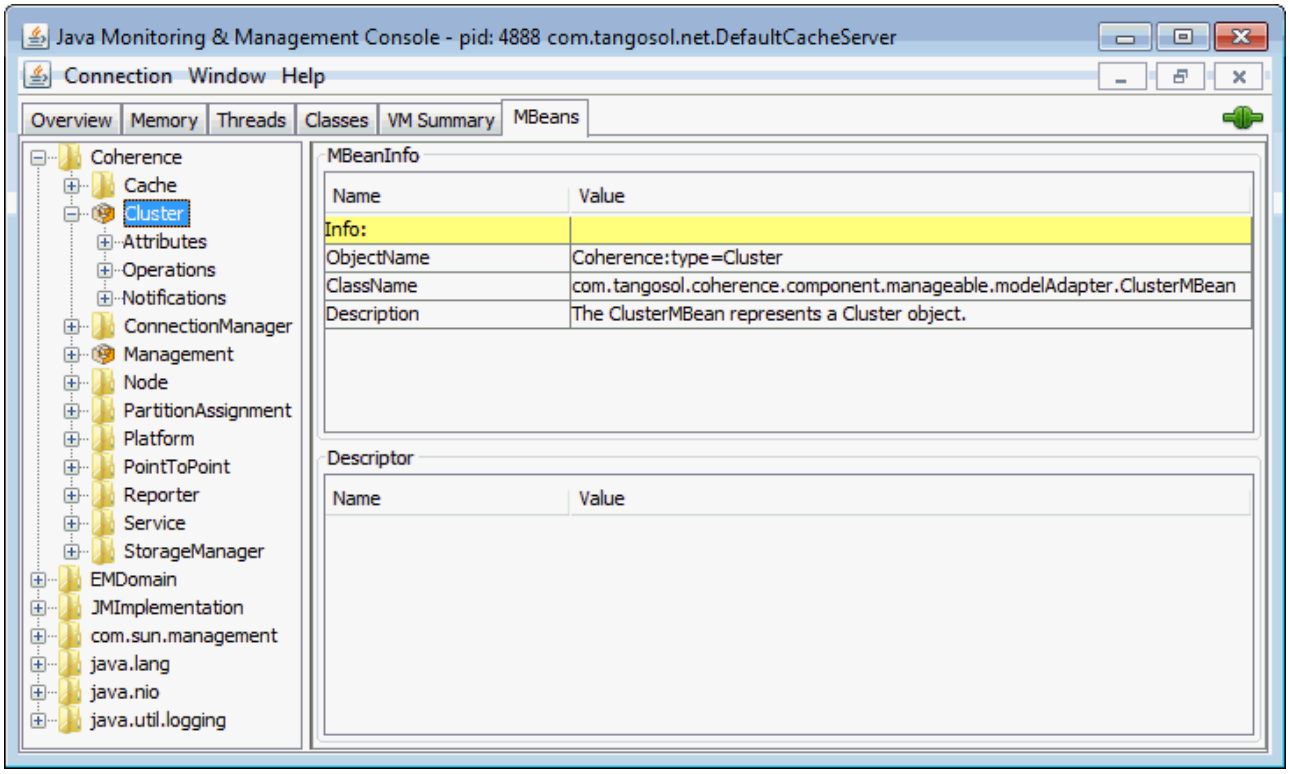
1. Start a cache server using the *COHERENCE_HOME*\bin\cache-server script. For example:

   COHERENCE_HOME\bin\cache-server

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. See Monitoring and Management Using JMX Technology in *Java SE Monitoring and Management Guide*. The following instructions provide only the minimum configuration that is required.

To allow remote access to Oracle Coherence MBeans without security enabled, set the following system properties when you start a cluster member:

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

> **Note:**
>
> Disabling SSL for remote JMX connections is *not* recommended and is insecure. It is recommended that you enable SSL for remote JMX connections. For recommendations on securing remote JMX connections, refer to your JDK documentation.

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
```

For more information, see the Java Platform Standard Edition 11 or Java Platform Standard Edition 17 documentation.

Accessing MBeans of a Running Coherence Cluster Using the JConsole Utility

To remotely access MBeans using the `jconsole` utility, use the Unix or Windows shell script `jmxserviceurl.[sh | cmd]` from the Coherence distribution `coherence/bin` by using the following command:

```
% $JAVA_HOME/bin/jconsole `jmxserviceurl.sh <clusterName> <clusterPort>
<hostName>`
```

In the above command:

- `clusterName` is the Coherence cluster name and is required.
- `clusterPort` is optional when no `<hostName>` parameter is used. It defaults to `7574`.
- `hostName` is the machine name or IP address of a Coherence cluster member. It defaults to `localhost`.

For example:

```
% $JAVA_HOME/bin/jconsole `./jmxserviceurl.sh mycluster 7574 100.111.142.210`
```

> **Note:**
>
> The example assumes that there is a Coherence cluster member for cluster `mycluster` at the default clusterport `7574` running on machine IP address `100.111.142.210`. This machine does not have to be the Coherence management senior, it just has to be running the `NameService` for `mycluster` on the specified clusterport.

When the `jxmserviceurl` shell script is run by itself, it has the following output that is a JMX RMI URL that `jconsole` uses to connect to the MBean server.

```
$ ./jmxserviceurl.sh mycluster 7574 100.111.142.210
service:jmx:rmi://100.111.142.210:42788/stub/
rO0ABXNyAC5qYXZheC5tYW5hZ2VtZW50LnJlbW90ZS5ybWkuUk1JU2VydmVySW1wbF9TdHViAAAAAA
AAAAICAAB4cgAaamF2YS5ybWkuc2VydmVyLlJlbW90ZVN0dWLp
```

The JConsole application starts and auto connects to the Coherence MBean server.

1. Click **Insecure Connection**.

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

Accessing MBeans Using the HTML Adapter Application

The HTML Adapter Web Application 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 at Java Platform Technology Downloads.

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:

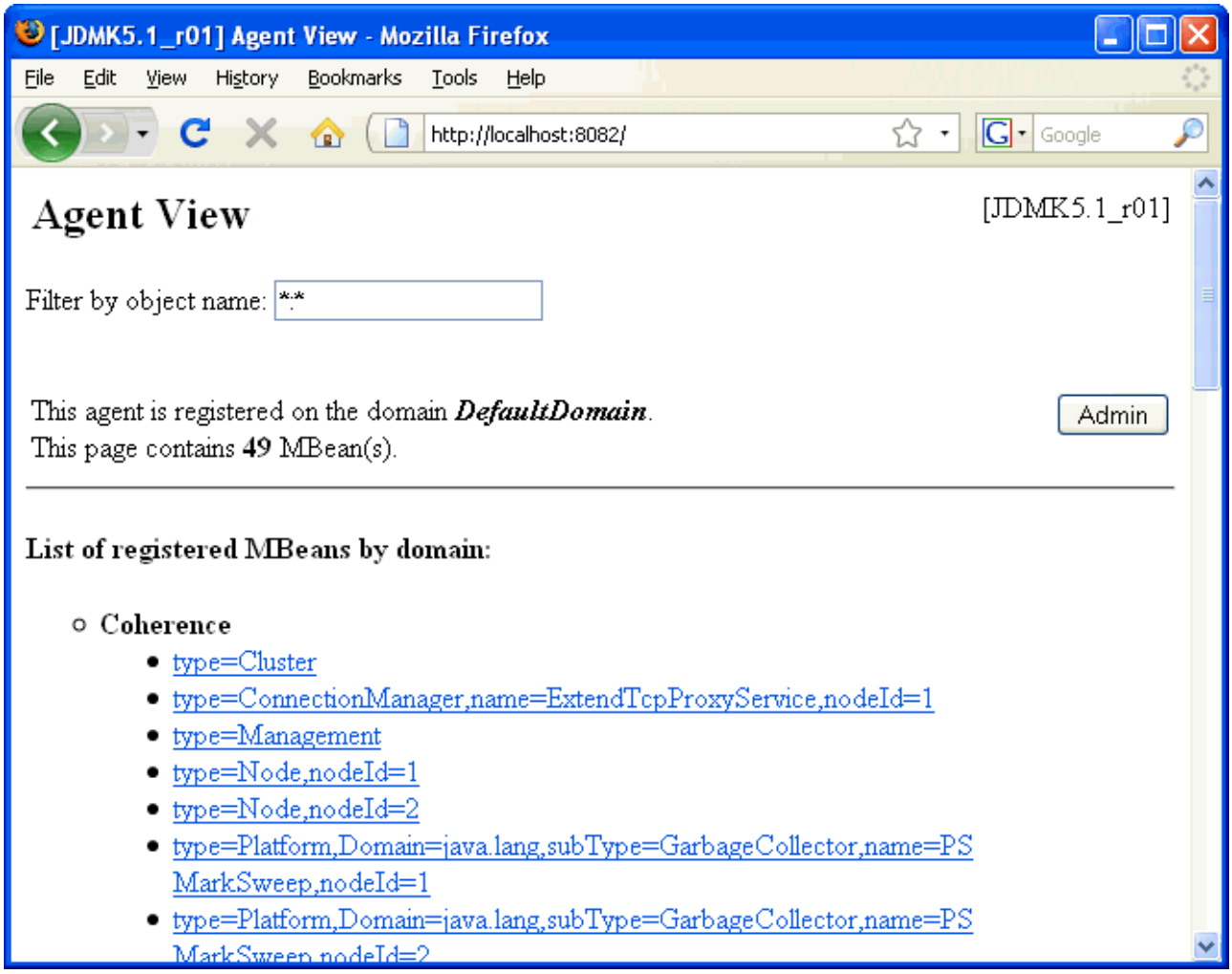
   ```
   -cp "jmxri-1.2.1.jar;jmxtools-1.2.1.jar;%coherence_home%\lib\coherence.jar"
   ```

2. Start a cache factory instance using the script. For example:

   ```
   COHERENCE_HOME\bin\coherence
   ```

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:



Accessing Coherence MBeans by Using WLST

When you run Coherence within WebLogic Server in a managed Coherence Servers environment, WebLogic Server domain runtime MBean server collects JMX information from the management proxy. You can access this information by using WLST.

For details, see Accessing Coherence MBeans by Using WLST in *Administering Clusters for Oracle WebLogic Server*.

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 -Dcoherence.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]
```

> **Note:**
>
> Disabling SSL for remote JMX connections is *not* recommended and is insecure. It is recommended that you enable SSL for remote JMX connections. For recommendations on securing remote JMX connections, refer to your JDK documentation.

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.
`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):
>
> Monitoring and Management Using JMX Technology in *Java SE Monitoring and Management Guide*.

Accessing Management Information Using REST

You can use HTTP to access Coherence MBeans that are exposed as REST resources. REST enables a broad range of clients to access management information.

To secure HTTP Management over the REST server, see Securing Oracle Coherence HTTP Management Over REST Server.

This section includes the following topics:

- Enabling the HTTP Management Server
- Connecting to Management Resources
- Changing the HTTP Management Server Address

Enabling the HTTP Management Server

REST management is enabled through an HTTP management server that runs as a proxy on the Coherence cluster. The HTTP management server requires the *COHERENCE_HOME*\lib\coherence.jar and *COHERENCE_HOME*\lib\coherence-json.jar libraries. The server starts when <http-managed-nodes> element in management-config is set to either all or inherit and the required libraries are found on the classpath. If the libraries are not found, then Coherence log messages indicate that the management libraries are missing and that management over HTTP is not available.

It is a best practice to manage dependencies using Maven. It is assumed that the latest Oracle Coherence artifacts are installed in your local maven repository using the Oracle Maven synchronization plug-in as described in Populating the Maven Repository Manager.

To generate a classpath containing third party libraries, run the following Maven command with the provided pom.xml below. Add the generated classpath to start cache server.

```
mvn dependency:build-classpath
```

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <artifactId>management</artifactId>
  <groupId>management</groupId>
  <version>1.0.0</version>
  <name>Coherence Management Over REST dependencies</name>
  <packaging>pom</packaging>

  <properties>
    <coherence.groupId>com.oracle.coherence</coherence.groupId>
    <coherence.version>14.1.1-2206-1</coherence.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>${coherence.groupId}</groupId>
```

```
      <artifactId>coherence</artifactId>
      <version>${coherence.version}</version>
    </dependency>
    <dependency>
      <groupId>${coherence.groupId}</groupId>
      <artifactId>coherence-json</artifactId>
      <version>${coherence.version}</version>
    </dependency>
  </dependencies>
</project>
```

All the required libraries are automatically downloaded. To see the complete list of libraries, run the following Maven command:

```
mvn dependency:list
```

Connecting to Management Resources

The HTTP management server is not enabled by default. It can be configured to use dynamic management mode to `inherit` and select port 30000 by default. See Changing the HTTP Management Server Address, and `<http-managed-nodes>` in management-config. If the senior cluster member is not operational, then the management server automatically restarts on the next most senior cluster member. The Coherence logs indicate the host and port of the server or you can query the `NameService` service on any cluster member using the `com.tangosol.discovery.NSLookup` class. The class has a command line interface or you can call the `lookupHTTPManagementURL` method. For example:

```
java -cp %COHERENCE_HOME%\lib\coherence.jar com.tangosol.discovery.NSLookup -
name management/HTTPManagementURL

Cluster MyCluster:      [http://127.0.0.1:63660/management/coherence/cluster]
```

The following example uses a cURL request to get cluster information:

```
curl -i -X GET http://127.0.0.1:63660/management/coherence/cluster
```

Response:

```
HTTP / 1.1 200 OK
Content - Type: application / json
content - length: 1330
connection: keep - alive

{
    "links": [{
        "rel": "parent",
        "href": "http://127.0.0.1:63660/management/coherence"
    }, {
        "rel": "self",
        "href": "http://127.0.0.1:63660/management/coherence/cluster"
    }, {
        "rel": "canonical",
        "href": "http://127.0.0.1:63660/management/coherence/cluster"
```

```
    }, {
        "rel": "services",
        "href": "http://127.0.0.1:63660/management/coherence/cluster/services"
    }, {
        "rel": "caches",
        "href": "http://127.0.0.1:63660/management/coherence/cluster/caches"
    }, {
        "rel": "members",
        "href": "http://127.0.0.1:63660/management/coherence/cluster/members"
    }, {
        "rel": "management",
        "href": "http://127.0.0.1:63660/management/coherence/cluster/
management"
    }, {
        "rel": "journal",
        "href": "http://127.0.0.1:63660/management/coherence/cluster/journal"
    }, {
        "rel": "hotcache",
        "href": "http://127.0.0.1:63660/management/coherence/cluster/hotcache"
    }, {
        "rel": "webApplications",
        "href": "http://127.0.0.1:63660/management/coherence/cluster/
webApplications"
    }],
    "refreshTime": "2018-12-12T18:04:34.225-05:00",
    "licenseMode": "Development",
    "clusterSize": 1,
    "localMemberId": 1,
    "version": "19.1.0.0.0",
    "running": true,
    "clusterName": "MyCluster",
    "membersDeparted": [],
    "memberIds": [1],
    "membersDepartureCount": 0,
    "members": ["Member(Id=1, Timestamp=2018-12-12 18:01:38.132,
Address=127.0.0.1:63621, MachineId=46461,
Location=site:Burlington,rack:100A,process:12124,member:COH-001, Role=Cache
Server)"],
    "oldestMemberId": 1,
    "type": "Cluster"
}
```


For a detailed reference of the REST API, see REST API for Managing Oracle Coherence.

Changing the HTTP Management Server Address

You can explicitly define the address and port of the HTTP management server. For most use cases, changing the address is not required and it is a best practice to configure dynamic management mode to enable HTTP management. The default `coherence.management.http.port` of `30000` need not be changed for virtual containers. For scenarios where coherence HTTP management server running in a non-virtual container is experiencing conflicts with other coherence cache servers/applications on default port of 30000, setting the `coherence.management.http.port` to `0` (specifying to use an ephemeral port) avoids potential conflicts. See Connecting to Management Resources.

> **Note:**
>
> If multiple coherence cache servers with HTTP management server configured with `<http-managed-nodes>` element set to `all` and are started on same machine, the `coherence.management.http.port` must be unique for each server. Otherwise, the subsequent ones with same `coherence.management.http.port` value will get an address which is already in use resulting in an error in starting up the HTTP management server. The coherence cache server will start up, but the HTTP management server will only be available on one coherence cache server on that machine.

Override the default HTTP management configuration using the `<http-managed-nodes>` element and set the value to `all`:

```
<?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>
      <http-managed-nodes system-property="coherence.management.http">all
      </http-managed-nodes>
   </management-config>
</coherence>
```

The `coherence.management.http` system property also configures management. For example:

```
-Dcoherence.management.http=all
```

Specify the address and port of the HTTP management server using the following system properties:

```
-Dcoherence.management.http.address
-Dcoherence.management.http.port
```

Using the Coherence VisualVM Plug-In

The Coherence VisualVM Plug-in provides management and monitoring of a single Coherence cluster using the VisualVM management utility. The VisualVM Plug-in is an open-source plug-in and the source is available on the repository in GitHub. See coherence-visualvm.
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. You can connect to clusters through JMX or through management over REST with Coherence versions 14.1.1.0 or above.

The plug-in is an ideal tool for monitoring and managing Coherence clusters during the development and testing lifecycle and supports connecting to both Community Edition and Commercial versions of Coherence.

> **Note:**
>
> - Oracle recommends that you use the open source Coherence VisualVM Plug-in instead of the plug-in that is shipped with the Coherence installer.
> - You can also use the plug-in to monitor production clusters, but you should ensure that the Data Refresh Time is set to a value of 30 seconds or greater, so you do not inadvertently add undue pressure on the Coherence management infrastructure.

This section includes the following topics:

Installing the Coherence VisualVM Plug-In

The Coherence VisualVM Plug-in is available from the list of plug-ins in VisualVM versions 2.1 and above. You can download VisualVM from https://visualvm.github.io/.

To install the Coherence VisualVM plug-in:

1. Launch VisualVM from the `bin` directory under the VisualVM install directory by using the following commands:

   ```
   cd install_directory/bin
   ```

   ```
   ./visualvm
   ```

2. From the VisualVM main menu, select **Tools** and click **Plugins**.
3. If the VisualVM-Coherence plug-in is not displayed in the **Available Plugins** tab, click **Check for Newest**.
4. Select the **Install** checkbox for the VisualVM-Coherence plug-in. The details of the plug-in are displayed in the right pane.
5. Click **Install**.
6. On the Plugin Installer Welcome screen, click **Next**.
7. In the Plugin Installer License Agreement screen, check **I accept** and click **Install**.
8. The plug-in begins to install. When the installation completes, click **Finish**.

Monitoring a Coherence Cluster Using the VisualVM Plug-In

A Coherence cluster is monitored in the 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.

You can connect to a Coherence cluster in one of the following ways:

- **Connecting Directly to a Process**
  After you install the plug-in, double-click the Coherence cluster member process in the left pane, usually `DefaultCacheServer`, after which the right pane of the VisualVM window displays a Coherence tab for the newly selected process.
- **Connecting Through Management Over REST**
  You can also connect through Coherence Management over REST. Right-click the **Coherence Clusters** tree and select **Add Coherence Cluster**. Provide a name for the cluster and use the following URL depending on the type of cluster to which you are connecting:
  - **Standalone Coherence** - `http://<host>:<management-port>/management/coherence/cluster`
  - **WebLogic Server** - `http://<admin-host>:<admin-port>/management/coherence/<version>/clusters` (use `latest` as the version)

  If you are connecting to WebLogic Server, you will prompted for the user name and password of the domain.

> **Note:**
>
> To enable Management over REST for a standalone cluster, see REST API for Managing Oracle Coherence.

It is preferred to use REST to connect to WebLogic Server, but if you want to connect by using JMX, see Connecting to Coherence in WebLogic Server Using the Administration Server.

Changing the Plug-In Behavior Using the Options Tab

You can change the behavior of the plug-in by using the **Options** pane. To open the options, choose the following depending upon your platform:

- **Mac:** Select **VisualVM**, click **Preferences**, and then select the **Coherence** tab.
- **Windows/Linux:** Select **Tools**, click **Options**, and then select the **Coherence** tab.

You will see the preferences of which the most commonly changed value is the **Data Refresh Time**. This value determines frequency at which the plug-in will refresh the data coming from the JMX or the REST connection.

Although there are tool tips for each of the preferences, Table 2-4 provides a summary:

Table 2-4 Coherence VisualVM Preferences

Preference	Default	Usage
Data Refresh Time	30	Time (in seconds) between refreshing data from the cluster. Do not set this value too low because a low value could adversely affect performance in large clusters.

Table 2-4 (Cont.) Coherence VisualVM Preferences

Preference	Default	Usage
Log Query Times	false	Enables logging of query times to the VisualVM log file when retrieving data.
Disable MBean Check	false	Disables the MBean check when connecting to WebLogic Server. This enables the plug-in to start up without checking for the Cluster MBean.
REST Request Timeout	30000	The request timeout (in ms) when using REST to connect to a cluster.
Enable REST Debug	false	Enables HTTP request debugging when using REST to connect to a cluster.
Disable SSL Certificate Validation	false	If selected, will disable SSL certificate validation. **Note**: You should use this option only when you are sure of the identify of the target server.
Enable Persistence List	true	Provides a drop-down list of the snapshots rather than having to enter the snapshot when performing snapshot operations.
Enable Zoom on Graphs	false	Enables additional zoom function for all graphs.
Enable Cluster Snapshot Tab	false	Enables the experimental 'Cluster Snapshot' tab. This tab is useful for viewing all the relevant cluster information on one pane in a text format.
Enable Cluster Heap Dump	false	Enables the 'Cluster Heap Dump' button on the 'Cluster Overview' tab.
Analyze Unavailable Time in Logfile	-	Provides the ability to analyze log files where 'Partition Events Logging' has been enabled for logs generated from Coherence 14.1.1.2206 and later. See Logging Partition Events. **Note:** You can select a Coherence log file to analyze. You do not need to be connected to a running cluster

Note:

The values you set are persisted between restarts of the plug-in.

Monitoring Capabilities of the Plug-In

For all Coherence clusters, the following tabs are displayed:

Table 2-5 Tabs Available in All Coherence Clusters

Name of the Tab	Description
Cluster Overview	Displays a high-level information about the Coherence cluster including cluster name, version, member count, and 'Cluster StatusHA'. Summary graphs show total cluster memory available and used, packet publisher and receiver success rates, and load averages for machines that run Coherence.
Machines	Displays a list of the physical machines that make up the Coherence cluster as well as information about the load averages and available memory on these machines.
Members	Displays the full list of Coherence members/nodes including individual publisher/ receiver success rates, memory, and send queue sizes.
Services	Displays information about the running services including partition counts and statusHA values. If you select a service, on the next data refresh, you will see a detailed thread information for each node of the service as well as graphs of that information.
Caches	Displays information about any caches including size, and memory usage information. To get the correct information to be displayed for memory usage, you must use the binary unit-calculator. If you select a cache, on the next data refresh, you will see detailed information about each node hosting that service and cache.

Depending upon the edition and functionality you are using, the following optional tabs may be displayed:

Table 2-6 Optional Tabs in Coherence Clusters

Name of the Tab	Description
Proxy Servers	If your cluster is running proxy servers, this tab displays information about the proxy servers and the number of connections across each proxy server and the total connections.
HTTP Servers	If your cluster is running proxy servers with HTTP acceptors, this tab displays information about the HTTP servers, the number of connections across each server, total connections and graphs of response codes, errors and requests over time for a selected service.

Table 2-6 (Cont.) Optional Tabs in Coherence Clusters

Name of the Tab	Description
Executors	If your cluster is configured to run the Executor Service, this tab displays information about the number of tasks completed, in-progress, and rejected.
Coherence*Web	If your cluster is configured for Coherence*Web, this tab displays information about the number of applications deployed, the number of HTTP sessions being stored as well as other information regarding session reaping.
Federation	If your cluster is configured with Federated Caching, this tab displays information about each federated service. If you select a service, on the next data refresh, you will see detailed outbound/ inbound federation traffic information for each node of the service as well as graphs of that information.
Persistence	If your cluster is configured with Persistence, this tab displays information about each service configured with Persistence. Graphs showing active space used and any additional latencies incurred are also shown.
Elastic Data	If your cluster is configured with Elastic Data, this tab displays graphs and information about RAM Journal and Flash Journal usage. You can click on each of the usage bars to view detailed node information.
JCache	If your cluster is being used to store JCache caches, this tab displays JCache "Management" and "Statistics" MBean information regarding the configured caches.
HotCache	If your cluster contains HotCache node(s), this tab lists the running HotCache instances. If you select an instance, on the next data refresh, the console will display statistics and graphs for the operations performed. You may click on tabs and cache-ops to view further fine-grained information.
gRPC Proxies	If you cluster is configured with gRPC Proxies, this tab displays information about the requests sent and received as well as successful and failed requests. A graph of message rates and durations is also displayed. This tab will be displayed only when connected through JMX and is not supported for REST connections.

3
Registering Custom MBeans

The Oracle Coherence management framework can manage custom MBeans together with Coherence MBeans. Custom MBeans must be registered with the Coherence MBean servers. This chapter includes the following sections:

Overview of Registering Custom MBeans

Coherence supports managing and monitoring custom MBeans within the Coherence management framework. Custom MBeans are MBeans that are specific to an application (including MBeans for the Java platform). Integrating custom MBeans together with Oracle Coherence MBeans allows you to update and view system and application information for all members in a cluster from a single location.
Custom MBeans are registered to an Oracle Coherence MBean server either declaratively in an XML file or programmatically.

> **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 can be registered with the Coherence management framework using the `custom-mbeans.xml` MBean configuration override file.

At run time, the first `custom-mbeans.xml` file that is found in the classpath is used. MBeans are declared in the configuration file using a class name, class factory name, or a query string that is executed against an MBean server.

> **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 mbeans in *Developing Applications with Oracle Coherence*.

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

```
-Dcoherence.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

Custom MBeans can be registered with the Coherence management framework using the `com.tangosol.net.management.Registry` interface from your application.
The following example registers a standard MBean using the `register` method.

```
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 enables 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 Interface MBeanServer in *Java™ Platform, Standard Edition API Specification*.
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.

> **Note:**
>
> Ensure that an environment is set up properly to access the local MBean server. See Monitoring and Management Using JMX Technology in *Java SE Monitoring and Management Guide*.

Registering Custom MBeans in Managed Coherence Servers

You can use XML to register custom MBeans when using managed Coherence servers. Managed Coherence servers, like standalone Coherence, can use a `custom-mbeans.xml` file to register custom MBeans. See Registering Custom MBeans Declaratively.
There are three ways to specify an MBean configuration file when using managed Coherence server:

- Add the `custom-mbeans.xml` file to the server classpath: When Coherence starts, it looks for the `custom-mbeans.xml` file in the server classpath and uses this as a MBean configuration file. This file and any associated classes can be added to the classpath by putting them in a JAR file that is located in the domain directory within the `/lib` subdirectory.
- Specify a different MBean configuration file: Use the `tangosol.coherence.mbeans` system property when starting a managed Coherence server to specify an MBean configuration file other than the default `custom-mbeans.xml` file.

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

  Classes that are associated with custom MBeans must be added to the classpath.
- Use the `CoherenceClusterSystemResources` MBean: Users can define custom MBeans in an operational override file and specify that file in the `CustomClusterConfigurationFileName` attribute of the `CoherenceClusterSystemResources` MBean. The specified configuration file is loaded on startup. See Registering Custom MBeans Declaratively.

4

Using Oracle Coherence Metrics

Oracle Coherence Metrics enables an alternative means to monitor a running Coherence cluster than JMX MBeans.
This chapter includes the following sections:

- About Coherence Metrics
- Enabling Coherence Metrics Endpoint
  To enable the Coherence Metrics endpoint, set the system property `coherence.metrics.http.enabled` to `true` when starting the Coherence cache server.
- Coherence Metrics Endpoint for WebLogic Managed Coherence Server
  Coherence includes a metrics resource that provides a metrics endpoint. You can use this metrics endpoint to extend the metrics capability for a WebLogic Managed Coherence Server by scraping its metrics using a metrics gathering system such as Prometheus.
- Using Metrics System Properties
- Configuring Prometheus to Scrape Coherence Metrics Endpoint
- Visualizing Metrics in Grafana
- Querying for Coherence Metrics
- Adding Custom Metrics
- Using Coherence Micrometer Metrics
  The `coherence-micrometer` module provides integration between Coherence metrics and Micrometer allowing Coherence metrics to be published through any of the Micrometer registries.

About Coherence Metrics

As of version 12.2.1.4, Coherence provides a metrics endpoint which extends the metrics capability by allowing scraping of metrics via metrics gathering systems such as Prometheus. The Coherence metrics endpoints are enabled on each Coherence Cache server, and a Prometheus scraping job is configured with the location of these endpoints.
If an application requires the Coherence Metrics endpoints to be secure, it is possible to configure the Coherence Metrics endpoint to be secure and to configure the Prometheus scraping job with the necessary security information. See Securing Oracle Coherence Metrics in *Securing Oracle Coherence*.

The Coherence Metrics endpoint supports content-encoding of gzip. By default, the metrics endpoint returns metrics in text, or Prometheus format, but can also return metrics in JSON format if the `Accepts: application/json` HTTP header is set.

To secure Oracle Coherence Metrics endpoint, see Securing Oracle Coherence Metrics in *Securing Oracle Coherence*.

In this release of Coherence, the coherence-metrics module has been deprecated and all the metrics functionality has been moved into `coherence.jar`. Unlike previous versions, Coherence metrics no longer require third-party dependencies. Therefore, all applications should remove any dependency on `coherence-metrics.jar` as this module will be removed in a future release.

Enabling Coherence Metrics Endpoint

To enable the Coherence Metrics endpoint, set the system property `coherence.metrics.http.enabled` to `true` when starting the Coherence cache server.

Coherence Metrics Endpoint for WebLogic Managed Coherence Server

Coherence includes a metrics resource that provides a metrics endpoint. You can use this metrics endpoint to extend the metrics capability for a WebLogic Managed Coherence Server by scraping its metrics using a metrics gathering system such as Prometheus.

To enable the Coherence Metrics endpoint, when you start the managed WebLogic Managed Coherence Server, set the `coherence.metrics.http.enabled` system property to `true`.

The endpoint URL is based on the WebLogic Managed Server listening port. If the port is set to 7003, then the following URL would return complete Coherence metrics: `http://<serverHost>:7003/management/coherence/metrics`.

You can also query specific metrics. For example, to get the cluster size, use this URL:

```
http://<serverHost>:7003/management/coherence/metrics/Coherence.Cluster.Size?cluster=<CoherenceClusterName>
```

Using Metrics System Properties

The metrics system properties are defined in the `metrics-http-config.xml` file in `coherence.jar`.
This section contains the following topics:

- Coherence Metrics System Properties
- Other Metrics System Properties

Coherence Metrics System Properties

The following Coherence Metrics system properties are defined in the default `metrics-http-config.xml` file in `coherence.jar`.

Table 4-1 Coherence Metrics system properties

System Property	Description	Default
`coherence.metrics.http.enabled`	Determines if metrics endpoint enabled	`false`
`coherence.metrics.http.address`	Specifies the address to listen on	`0.0.0.0` (All interfaces)
`coherence.metrics.http.port`	Specifies the port to listen on (0 indicates the ephemeral port)	`9612`

Table 4-1 (Cont.) Coherence Metrics system properties

System Property	Description	Default
`coherence.metrics.http.auth`	Configures the authentication mechanism for the HTTP server `basic`, `auth`, `basic+auth` or `not set`	not set
`coherence.metrics.http.provider`	Defines the id of a `<socket-provider>` in coherence operational override file	none
`coherence.metrics.http.path`	Specifies the metrics context root path.	not set

You can override the default `metrics-http-config.xml` by placing an override file with same file name in classpath before `coherence.jar`.

> **Note:**
>
> If multiple Coherence cache servers with metrics endpoint enabled are started on same machine, the `coherence.metrics.http.port` must be unique for each server. If not, the subsequent ones with same `coherence.metrics.http.port` value will get an address already in use and will result in an error with starting up the `MetricsHttpProxy` service.
>
> The Coherence cache server will start up but the Coherence Metrics endpoint will only be available on one Coherence cache server.

Other Metrics System Properties

The other metrics system property includes:

Table 4-2 Other metrics system properties

System Property	Description	Default
`coherence.metrics.extended`	Includes extended information (type and/or description) when publishing metrics.	`false`

The extended information for Prometheus metrics is #Help, which contains the metrics name. If more data follows, the #Help is considered the docstring for that metric name. For Coherence Metrics, the docstring is typically the JMX MBean attribute description from which the metric value was derived. There is one or no #Help per metrics value in a Prometheus response.

Configuring Prometheus to Scrape Coherence Metrics Endpoint

Configuring a scraping job to scrape the Coherence Metrics endpoints for a Coherence cluster is done using the `static_configs` parameter of a Prometheus configuration file. You need to specify the IP address and `coherence.metrics.http.port` for each Coherence Metrics endpoint in the `static_configs` of a Prometheus configuration file. Once configured and

started, the Prometheus server will scrape the configured static targets at the configured interval, collecting metrics data for the Coherence cluster.
See Prometheus configuration documentation.

> **Note:**
>
> The plain text format metric names published by the Coherence metrics endpoint (for example when being scraped by Prometheus) are in the `vendor:coherence_<metric_name>` format. This legacy name format has been deprecated, and will be changed in a future Coherence release to be `coherence_<metric_name>`. That is, the `vendor:` prefix will be removed. If you want to use the future metric name format, set the `coherence.metrics.legacy.names` system property to `false` when starting Coherence.

Visualizing Metrics in Grafana

After you have configured Prometheus to scrape metrics, you can visualize these metrics using the pre-built Grafana dashboards available in the Coherence Operator. See coherence-operator.

These dashboards provide detailed insight into your Coherence cluster by using the collected metrics and are a valuable tool in monitoring and managing the health of you cluster.

For more information about the dashboards and how to load them, see the Coherence Operator documentation.
For an example of the Coherence Summary Dashboard, see Use dashboards.

> **Note:**
>
> Oracle recommends you to use the minimum Grafana version 8.5.13 to view the dashboards. At this point, due to the many issues with Grafana 9.x, do not use this version yet.

Querying for Coherence Metrics

The metrics endpoint supports Prometheus and JSON formats. It also supports querying for specific metrics by name and tags.
This section contains the following topics:

- Basic Querying
- Querying for Specific Metrics

Basic Querying

The `coherence.metrics.http.port` is assumed to be set to `9612` for the below listed basic querying examples.

http://localhost:9612/metrics

By default, this will produce Prometheus' simple text-based exposition format. If the caller uses the `Accepts: application/json` header, then JSON is returned. The endpoint also supports content negotiation by appending the required type to the url.

http://localhost:9612/metrics/.txt returns Prometheus data whereas http://localhost:9612/metrics/.json returns JSON data.

The `coherence.metrics.extended` system property controls the inclusion of additional help or descriptions. By default, the `coherence.metrics.extended` is set to `false` and additional information is not included. By adding the `extended=true` query parameter, additional data can be included.

http://localhost:9612/metrics?extended=true (this applies to both Prometheus and JSON formats).

Querying for Specific Metrics

To query for specific metrics append the metric name to the URL.

To query for number of members in a cluster:

http://localhost:9612/metrics/Coherence.Cluster.Size

To query for all cache size metrics:

http://localhost:9612/metrics/Coherence.Cache.Size

To query for a metric with matching tags add the tag key and value as query parameters. To query for the size metric for a specific cache where the cache name is `cache_name`.

http://localhost:9612/metrics/Coherence.Cache.Size?name=cache_name

Additionally, if the cache (for example near cache) is included so that there are front and back tiers, then the request could be further restricted.

http://localhost:9612/metrics/Coherence.Cache.Size?name=cache_name&tier=back

http://localhost:9612/metrics/Coherence.Cache.Size?name=cache_name&tier=front

Adding Custom Metrics

You can add your own metrics if you annotate the MBeans that you add to the Coherence Registry.
For example:

```
import com.tangosol.net.management.annotation.MetricsValue;
import com.tangosol.net.management.annotation.MetricsTag;

public interface CustomMBean
    {
    @MetricsValue("custom_value")
    long getValueOne();

    @MetricsValue
    long getValueTwo();

    long getValueThree();

    @MetricsTag("custom_tag")
    String getTagValueOne();
```

```
@MetricsTag
String getTagValueTwo();
}
```

- The method `getValueOne` is annotated with `@MetricValue`. Hence, it will be a metric and the name of the metric will be `custom_value`.
- The method `getValueTwo` is annotated with `@MetricValue`. Hence, it will be a metric. The annotation does not specify a name and the name will be generated from the attribute name (in this case `value_two`).
- The method `getValueThree` is not annotated. Hence, though being an MBean attribute it will not be used as a metric or as a tag.
- The method `getTagValueOne` is annotated with `@MetricTag`. Hence, it will be a metric tag and the name of the tag will be `custom_tag`.
- The method `getTagValueTwo` is annotated with `@MetricTag`. Hence, it will be a metric tag. The annotation does not specify a name and the name will be generated from the attribute name (in this case `tag_value_two`).

For annotations to be detected by Coherence when registering the MBean, the annotations must be wrapped in an `AnnotatedStandardMBean`.

For example:

```
Registry registry   = CacheFactory.ensureCluster().getManagement();
Custom   custom     = new Custom();
String   sMBeanName = registry.ensureGlobalName("type=Custom,name=bar");

registry.register(sMBeanName, new AnnotatedStandardMBean(custom, CustomMBean.class));
```

The code above will register the Mbean with the Object name similar to `Coherence:type=Custom,name=bar,nodeId=1`

The metrics name produced will be prefixed with the value from the type key in the `ObjectName`, in this case `Custom`. This will produce the following metrics:

- `Custom_custom_value` from method `getValueOne`
- `Custom_value_two` from method `getValueTwo`

Both metrics will have common tags such as cluster name, nodeId, machine, role, site, and so on, as well as the custom tags `name=bar` from the ObjectName, `custom_tag` from the annotated method `getTagValueOne` and `tag_value_two` from the annotated method `getTagValueTwo`. The method `getTagValueTwo` is annotated with `@MetricTag`. Hence it will be a metric tag. The annotation does not specify a name and the name will be generated from the attribute name (in this case `tag_value_two`).

Using Coherence Micrometer Metrics

The `coherence-micrometer` module provides integration between Coherence metrics and Micrometer allowing Coherence metrics to be published through any of the Micrometer registries.

This section includes the following topics:

- Enabling the Use of Coherence Micrometer Metrics
- Automatic Global Registry Binding

Enabling the Use of Coherence Micrometer Metrics

To use Coherence Micrometer metrics, you should declare the module as a dependency in the `pom.xml` file and bind the Micrometer registry with the Coherence metrics adapter:

```
<dependency>
    <groupId>${coherence.groupId}</groupId>
    <artifactId>coherence-micrometer</artifactId>
    <version>${coherence.version}</version>
</dependency>
```

The `coherence-micrometer` provides a Micrometer `MeterBinder` implementation class called `CoherenceMicrometerMetrics`. This class is a singleton and cannot be constructed. To access this class, use the `CoherenceMicrometerMetrics.INSTANCE` field.

Micrometer provides many registry implementations to support different metrics applications and formats. For example, to bind Coherence metrics to the Micrometer `PrometheusMeterRegistry`, create the `PrometheusMeterRegistry`. For more information, see the Micrometer Documentation, and call the `CoherenceMicrometerMetrics` class's `bindTo` method:

```
PrometheusMeterRegistry prometheusRegistry = new
PrometheusMeterRegistry(PrometheusConfig.DEFAULT);

// complete the registry configuration...

CoherenceMicrometerMetrics.INSTANCE.bindTo(prometheusRegistry);
```

You can bind the Micrometer registries to Coherence at any time, before or after Coherence starts. As Coherence creates or removes metrics, they will be registered with or removed from the Micrometer registries.

Automatic Global Registry Binding

Micrometer has a global registry available which Coherence binds to automatically if you set the `coherence.micrometer.bind.to.global` system property to `true` (this property is `false` by default).

5

Using the Health Check API

Coherence 14.1.1.2206 introduces a health check API to enable application code to determine the health of the local Coherence member, and corresponding HTTP and management endpoints to allow external applications to query the health of a cluster and its members.

The health API also enables applications to register their own health checks, which is then included in the member's and cluster's health status.

> **Note:**
>
> To enable health checks, you must use the Boostrap API to start the Coherence cluster members. See Using the Bootstrap API.

This chapter includes the following topics:

- About the Health Check API
  You can use the health check API from the application code, to determine whether Coherence is healthy, and also from a HTTP endpoint making it useful for health checks in containerized environments such as Kubernetes and Docker.
- Using the Built-In Health Checks
- Enabling HTTP Health Checks
- Using Application Health Checks
- Using Containerized Health Checks
  When running Coherence applications in containers, for example, in Docker or Kubernetes, it is useful to be able to make use of health and readiness checks. By running Coherence with the health HTTP endpoints enabled, configuring container health is made simple.

About the Health Check API

You can use the health check API from the application code, to determine whether Coherence is healthy, and also from a HTTP endpoint making it useful for health checks in containerized environments such as Kubernetes and Docker.

All health checks in Coherence implement a simple interface.

```
public interface HealthCheck
    {
    /**
     * Returns the unique name of this health check.
     *
     * @return the unique name of this health check
     */
    String getName();

    /**
```

```
 * Return {@code true} if this {@link HealthCheck} should
 * be included when working out this Coherence member's
 * health status.
 *
 * @return {@code true} if this {@link HealthCheck} should
 *         be included in the member's health status
 */
default boolean isMemberHealthCheck()
    {
    return true;
    }

/**
 * Returns {@link true} if the resource represented by
 * this {@link HealthCheck} is ready, otherwise returns
 * {@code false}.
 * <p>
 * The concept of what "ready" means may vary for different
 * types of resources.
 *
 * @return {@link true} if the resource represented by this
 *         {@link HealthCheck} is ready, otherwise {@code false}
 */
boolean isReady();

/**
 * Returns {@link true} if the resource represented by
 * this {@link HealthCheck} is alive, otherwise returns
 * {@code false}.
 * <p>
 * The concept of what "alive" means may vary for different
 * types of resources.
 *
 * @return {@link true} if the resource represented by this
 *         {@link HealthCheck} is alive, otherwise returns
 *         {@code false}
 */
boolean isLive();

/**
 * Returns {@link true} if the resource represented by
 * this {@link HealthCheck} is started, otherwise returns
 * {@code false}.
 * <p>
 * The concept of what "started" means may vary for different
 * types of resources.
 *
 * @return {@link true} if the resource represented by this
 *         {@link HealthCheck} is started, otherwise returns
 *         {@code false}
 */
boolean isStarted();

/**
 * Returns {@link true} if the resource represented by this
 * {@link HealthCheck} is in a safe state to allow a rolling
```

```
     * upgrade to proceed, otherwise returns {@code false}.
     * <p>
     * The concept of what "safe" means may vary for different
     * types of resources.
     *
     * @return {@link true} if the resource represented by this
     *         {@link HealthCheck} is in a safe state to allow
     *         a rolling upgrade to proceed, otherwise returns
     *         {@code false}
     */
    boolean isSafe();
    }
```

The methods were specifically chosen to integrate with other systems where Coherence is run, for example Kubernetes, that use similar, "started", "live" and "ready" health checks. The "safe" check is specific to Coherence to be used for controlling use-cases such as rolling upgrades, where it is important to know that a cluster is "safe" before rolling the next cluster member.

The health check API is part of the Coherence management APIs and can be accessed from the `com.tangosol.net.management.Registry class`. The `Registry` is typically obtained from the current Coherence `Cluster` instance. For example, when Coherence has been started by running `com.tangosol.net.Coherence.main()`, or by using the bootstrap API, you can obtain the management `Registry` as shown below:

```
Cluster  cluster  = Coherence.getInstance().getCluster();
Registry registry = cluster.getManagement();
```

The health check API can only see registered health checks for the local Coherence member; it is not a cluster wide API. For cluster wide health checks, use the corresponding health MBeans through the Coherence management API, JMX, or management over REST.

This section includes the following topics:

- Obtaining All Health Checks
- Obtaining a Health Check by Name
- Checking All Health Checks Are Ready
- Checking All Health Checks Have Started
- Checking All Health Checks Are Live
- Checking All Health Checks Are Safe

Obtaining All Health Checks

To obtain a collection of all the registered health checks, the `getHealthChecks()` method can be called on the `Registry` instance. This method returns an immutable collection of registered `HealthCheck` instances.

For example, the code below obtains a `Set` of names of `HealthCheck` instances that are not ready:

```
Coherence coherence = Coherence.getInstance();
Registry registry = coherence.getManagement();
Collection<HealthCheck> healthChecks = registry.getHealthChecks();
Set<String> names = healthChecks.stream()
```

```
        .filter(hc -> !hc.isReady())
        .map(HealthCheck::getName)
        .collect(Collectors.toSet());
```

Obtaining a Health Check by Name

To obtain a specific health check by name, the `getHealthCheck(String name)` method can be called on the `Registry` instance. This method returns an `Optional` containing the requested `HealthCheck`, if one has been registered with the requested name, or returns an empty `Optional` if no `HealthCheck` has been registered with the request name.

For example, the code below obtains the `HealthCheck` with the name "Foo":

```
Coherence coherence = Coherence.getInstance();
Registry registry = coherence.getManagement();
Optional<HealthCheck> healthChecks = registry.getHealthCheck("Foo");
```

Checking All Health Checks Are Ready

The `allHealthChecksReady()` method on the `Registry` instance can be used to determine whether all locally registered health checks are ready. Only health checks that return `true` from their `isMemberHealthCheck()` method are included in the ready check.

```
Coherence coherence = Coherence.getInstance();
Registry registry = coherence.getManagement();
boolean ready = registry.allHealthChecksReady();
```

Checking All Health Checks Have Started

The `allHealthChecksStarted()` method on the `Registry` instance can be used to determine whether all locally registered health checks have been started. Only health checks that return `true` from their `isMemberHealthCheck()` method are included in the started check.

```
Coherence coherence = Coherence.getInstance();
Registry registry = coherence.getManagement();
boolean started = registry.allHealthChecksStarted();
```

Checking All Health Checks Are Live

The `allHealthChecksLive()` method on the `Registry` instance can be used to determine whether all locally registered health checks are live. Only health checks that return `true` from their `isMemberHealthCheck()` method are included in the live check.

```
Coherence coherence = Coherence.getInstance();
Registry registry = coherence.getManagement();
boolean live = registry.allHealthChecksLive();
```

Checking All Health Checks Are Safe

The `allHealthChecksSafe()` method on the `Registry` instance can be used to determine whether all locally registered health checks are safe. Only health checks that return `true` from their `isMemberHealthCheck()` method are included in the safe check.

```
Coherence coherence = Coherence.getInstance();
Registry registry = coherence.getManagement();
boolean safe = registry.allHealthChecksSafe();
```

Using the Built-In Health Checks

Coherence has a number of health checks that are ready-to-use.

- Each Coherence service has a corresponding health check.
- Instances of `com.tangosol.net.Coherence` provide a corresponding health check.
- When using Coherence gRPC integrations, the gRPC proxy server has a health check.

This section includes the following topics:

Using the Service Health Checks

For Coherence services, heath checks have the following functionality:

- **Started**: The `isStarted()` method for a service health check will return `true` if the corresponding service is running.
- **Live**: The `isLive()` method for a service health check will return `true` if the corresponding service is running.
- **Ready**: For a service, the `isReady()` method will return `false` until a service becomes "safe", after which the "ready" state will remain `true`. This is specifically for use cases such as Kubernetes, where Pods will be removed from a `Service` if not `Ready`. However, this behavior is typically not required for Coherence.
- **Safe**: For all services except a partitioned cache service, the `isSafe()` method will always return `true`.

Using the PartitionedCache Service isSafe Check

A Coherence `PartitionedCache` service is more complex than most services in Coherence, and as such, its health checks also do more. The `isSafe()` check for a `PartitionedCache` service performs a number of checks to ensure that the service is stable and safe. The main use-cases for the "safe" check are when performing a rolling upgrade or safely scaling down a cluster.

- The `isSafe()` health check for a `PartitionedCache` service on a non-storage enabled member will return `true` as long as the service is running.
- The `isSafe()` health check for a `PartitionedCache` service will return `false` if this member is the only storage enabled member for the service, but does not own all the partitions. This can happen just after all the other members of the cluster have been stopped but the partition recovery and reallocation logic is still in progress. Therefore, this member does not yet know that it owns all the partitions.
- The `isSafe()` health check for a `PartitionedCache` service will return `false` if the backup count is configured to be greater than zero and the StatusHA state for the service is `endangered`. You can change this behavior for individual services in the cache configuration file to allow them to be endangered. A service with a backup count of zero is allowed to be endangered for the safe check.
- The `isSafe()` health check for a `PartitionedCache` service will return `false` if partition redistribution is in progress.
- The `isSafe()` health check for a `PartitionedCache` service will return `false` if recovery from persistent storage is in progress.

Excluding Services from Member Health

Sometimes it may be desirable to exclude a Coherence service from the member's overall health check. This can be done by setting the `<member-health-check>` element in service's `<health>` element in the cache configuration file.

For example, the proxy-scheme below has the `<member-health-check>` element value set to `false`. The health checks for the `Proxy` service will still be accessible through the health API, but checks of the overall member health, such as the `Registry` class's `allHealthChecksReady()` method will not include the `Proxy` service.

```
<proxy-scheme>
  <service-name>Proxy</service-name>
  <autostart>true</autostart>
  <health>
    <member-health-check>false</member-health-check>
  </health>
</proxy-scheme>
```

Allowing Endangered Services

Sometimes an application may configure a distributed cache service that can intentionally become endangered. However, this state should not be reflected in the member's overall health. This can be done by setting the `<allow-endangered>` element in the distributed scheme's `<health>` element in the cache configuration file.

For example, the `distributed-scheme` below has the `<allow-endangered>` element value set to `true`. The health checks for the `PartitionedCache` will report that the service is "ready" or "safe" even if the StatusHA value for the service is 'ENDANGERED'.

```
<distributed-scheme>
  <scheme-name>distributed-scheme</scheme-name>
  <service-name>PartitionedCacheOne</service-name>
  <backing-map-scheme>
    <local-scheme/>
```

```
  </backing-map-scheme>
  <autostart>true</autostart>
  <health>
    <allow-endangered>true</allow-endangered>
  </health>
</distributed-scheme>
```

Enabling HTTP Health Checks

The health check HTTP endpoints are enabled when Coherence is run using the bootstrap API, or starting Coherence using `com.tangosol.net.Coherence` as the main class. If Coherence is started by any other method, the health check API is still available, but the HTTP endpoints will not be running. By default, the HTTP server will bind to an ephemeral port, but this can be changed by setting the `coherence.health.http.port` system property or the `COHERENCE_HEALTH_HTTP_PORT` environment variable.
For example, running the following command will start Coherence with the health endpoints on `http://localhost:6676`:

```
java -cp coherence.jar -Dcoherence.health.http.port=6676 \
    com.tangosol.net.Coherence
```

or with Java modules:

```
java -p coherence.jar \
    -Dcoherence.health.http.port=6676 \
    --module com.oracle.coherence
```

The `curl` utility can then be used to poll one of the endpoints, for example `/ready`:

```
curl -i -X GET http://localhost:6676/ready
```

The above command returns output like the following:

```
HTTP/1.1 200 OK
Date: Tue, 19 Apr 2022 17:59:05 GMT
Content-type: application/json
Vary: Accept-Encoding
Content-length: 0
X-content-type-options: nosniff
```

If Coherence health check fails, the response code will be `503`, for `service unavailable`.

- Health HTTP Endpoints

Health HTTP Endpoints

The health check HTTP server has a number of endpoints.

> **Note:**
>
> None of the endpoints accepts a payload or returns a response body. The only response is either a 200 or a 503 status code. This means that although the health endpoints can be configured to use SSL/TLS, there is little need for encryption, making their use by external tooling such as Kubernetes and other container environments simpler.

Table 5-1 Health Check HTTP Server Endpoints

Endpoint	Description
`/started`	This endpoint returns the status code of `200` as response if all the health checks for the member the request is sent to have been "started". If one or more health check is not started, the endpoint returns a `503` response.
`/live`	This endpoint returns the status code of `200` as response if all the health checks for the member the request is sent to are "live". If one or more health check is not live, the endpoint returns a `503` response.
`/ready`	This endpoint returns the status code of `200` as response if all the health checks for the member the request is sent to are "ready". If one or more health check is not ready, the endpoint returns a `503` response.
`/safe`	This endpoint returns the status code of `200` response if all the health checks for the member the request is sent to are "safe". If one or more health check is not safe, the endpoint returns a `503` response.

Using Application Health Checks

The health check API allows application developers to add custom health checks. This can be useful where an application provides a service that should be used to determine the overall health of a Coherence member. For example, an application could include a web server and should not be considered "ready" until the web server is started.
To register a custom health check, just write an implementation of `com.tangosol.util.HealthCheck`.

The `getName()` method for the custom health check should return a unique name that represents this health check. As health checks are exposed as MBeans, the name must be a name that is valid in a JMX MBean object name.

The health check implementation should then use relevant application logic to determine the result to return for each of the methods. Some methods may not apply, in which case they should just return `true`.

It is important to understand how the results of the different health check methods will be used outside the application code. For example, when the application is deployed and managed by an external system that monitors application health. For example, an application deployed into Kubernetes could be killed if it reports not being "live" too many times. An application that does

not report being "ready" may be excluded from request routing, and so on. An application that is not "safe" will block rolling upgrades or safe scaling of a Coherence cluster.

- Excluding Custom Health Checks from Member Health

Excluding Custom Health Checks from Member Health

An application developer may want to add custom health checks for application services, but not have these checks impact the overall Coherence member health. The `HealthCheck` interface has a `isMemberHealthCheck()` method for this purpose. The default implementation of `isMemberHealthCheck()` always returns `true`, so by default all health checks are included in the member's health. To exclude a health check from the member's health, override the `isMemberHealthCheck()` method to return `false`.

Using Containerized Health Checks

When running Coherence applications in containers, for example, in Docker or Kubernetes, it is useful to be able to make use of health and readiness checks. By running Coherence with the health HTTP endpoints enabled, configuring container health is made simple.

When using the health check endpoints in a container, the HTTP port needs to be fixed so that the image's health checks can be configured. The default behavior of binding to an ephemeral port would mean that the system does not know which port the health check API is bound to. The HTTP port can be set using the `coherence.health.http.port` system property or the *COHERENCE_HEALTH_HTTP_PORT* environment variable. When creating images, it is typically simpler to use environment variables, which is what the examples in this section show.

This section includes the following topics:

- Using the Docker Health Checks
- Using the Kubernetes Readiness and Liveness Probes

Using the Docker Health Checks

It is possible to build a Coherence Docker image configured with a health check using the `HEALTHCHECK` configuration in the Dockerfile.

The example `Dockerfile` below, sets the health check port to `6676` using the `ENV COHERENCE_HEALTH_HTTP_PORT=6676` setting. The `Dockerfile` is then configured with a `HEALTHCHECK` where the command will run `curl` against the HTTP endpoint on `http://127.0.0.1:6676/ready`. This command will fail if the response is not `200`.

```
FROM openjdk:11-jre

ADD coherence.jar /coherence/lib/coherence.jar

ENTRYPOINT [ "java" ]
CMD [ "-cp", "/coherence/lib/*", "com.tangosol.net.Coherence" ]

ENV COHERENCE_HEALTH_HTTP_PORT=6676

HEALTHCHECK CMD curl --fail http://127.0.0.1:6676/ready || exit 1
```

The check above assumes that the base image has `curl` installed. This is not always the case, for example, some very slim Linux base images or distroless images will not have any additional tools such as `curl`. In this case, all the image has is Java. Therefore, you can configure the health check to use a Java health check client class `com.tangosol.util.HealthCheckClient` that is built into the Coherence jar. This class can be run with a single parameter, which is the URL of the HTTP endpoint to check.

The example `Dockerfile` below uses a distroless base image that only has a Linux kernel and Java 11 installed. The health check port is set to `6676` using the `ENV COHERENCE_HEALTH_HTTP_PORT=6676` setting. The `Dockerfile` is then configured with a `HEALTHCHECK` where the command will run `java -cp /coherence/lib/coherence.jar com.tangosol.util.HealthCheckClient http://127.0.0.1:6676/ready`. This command will fail if the response is not `200`.

```
FROM gcr.io/distroless/java11

ADD coherence.jar /coherence/lib/coherence.jar

ENTRYPOINT [ "java" ]
CMD [ "-cp", "/coherence/lib/*", "com.tangosol.net.Coherence" ]

ENV COHERENCE_HEALTH_HTTP_PORT=6676

HEALTHCHECK CMD java -cp /coherence/lib/coherence.jar
com.tangosol.util.HealthCheckClient http://127.0.0.1:6676/ready
```

Using the Kubernetes Readiness and Liveness Probes

In Kubernetes, there are various readiness and liveness probes that can be configured. The image itself does not need a health check (see Using the Docker Health Checks) as Kubernetes readiness and liveness is independent of the image. For complete details about configuring Kubernetes readiness and liveness, see the Kubernetes documentation.

The example below is just a simple `Pod` using a Coherence image and health checks. The `COHERENCE_HEALTH_HTTP_PORT` environment variable is used to fix the health check HTTP port to `6676`. The `readinessProbe` is then configured to use a HTTP GET request on port `6676` using the request path `/ready`. The host for the request defaults to the Pod IP address, so will effectively be the same as `http://<pod-ip>:6676/ready`.

```
apiVersion: v1
kind: Pod
metadata:
  name: coherence
spec:
  containers:
  - name: coherence
    image: ghcr.io/oracle/coherence-ce:22.06
    env:
      - name: COHERENCE_HEALTH_HTTP_PORT
        value: "6676"
      - name: COHERENCE_WKA
        value: coherence_wka.svc.cluster.local
    readinessProbe:
      httpGet:
        path: "/ready"
```

```
  port: 6676
initialDelaySeconds: 30
periodSeconds: 30
```

6

Using Oracle Coherence Reporting

Oracle Coherence provides a reporting feature that collects management data and generates a series of reports that are updated at regular intervals. Reporting supports many configuration options, such as controlling which reports get generated, where the reports are saved, and how often the report data is refreshed.
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.

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 6-1 summarizes the reports.

Table 6-1 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
YYYYMMDDHH-network-health.txt	Contains the publisher success rates and receiver success rates for the entire cluster
YYYYMMDDHH-network-health-detail.txt	Contains the publisher success rates and receiver success rates for each member
YYYYMMDDHH-nodes.txt	Contains a list of members of the cluster

Table 6-1 (Cont.) Default Oracle Coherence Reports

File Name	Description
YYYYMMDDHH-persistence.txt	Contains cache persistence information for a particular service
YYYYMMDDHH-persistence-detail.txt	Contains cache persistence information for a particular service and node
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

- Understanding Reporting Configuration
- Customizing Reports

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 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-group.xml`, and `report-all.xml`. See 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 Creating Custom Reports.

Enabling Oracle Coherence Reporting on a Cluster Member

Reporting is disabled by default and must be explicitly enabled. You can edit the reporting configuration to enable reporting on a cluster member and generate management reports. Ensure that JMX management is enabled on the cluster member before you enable reporting. See Configuring JMX Management.
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="coherence.management.report.autostart">true
         </autostart>
      </reporter>
   </management-config>
</coherence>
```

The `coherence.management.report.autostart` system property also enables reporting. For example:

```
-Dcoherence.management.report.autostart=true
```

Switching the Default Report Group Configuration File

You can control which reports are generated by specifying a different report group configuration file.

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. See Running Performance Reports in *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="coherence.management.report.configuration">
            reports/report-all.xml</configuration>
      </reporter>
   </management-config>
</coherence>
```

The `coherence.management.report.configuration` system property also switches the default report group configuration file. For example:

```
-Dcoherence.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. You can configure the output directory on a cluster member where reports are saved.
The output directory is overridden using the `coherence.reporter.output.directory` system property. For example:

```
-Dcoherence.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

The report time zone and time and date format can be change as required. Report timestamps display the local time zone and a default time and date format (`EEE MMM dd HH:mm:ss zzz yyyy`). 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="coherence.management.report.timezone">
            US/Eastern
         </timezone>
         <timeformat
            system-property="coherence.management.report.timeformat">
            MMM dd, yyyy  hh:mm:ss a
         </timeformat>
      </reporter>
   </management-config>
</coherence>
```

The `coherence.management.report.timezone` and the `coherence.management.report.timeformat` system properties also change the report timestamp format. For example:

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

Administering Oracle Coherence Reporting Using the Reporter MBean

The `ReporterMBean` MBean provides a convenient way to manage Coherence reporting and perform common reporting tasks.

The MBean contains many attributes and operations for administering reporting in real time. See Accessing 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.

The `ReporterMBean` MBean performs the following tasks:

- 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.

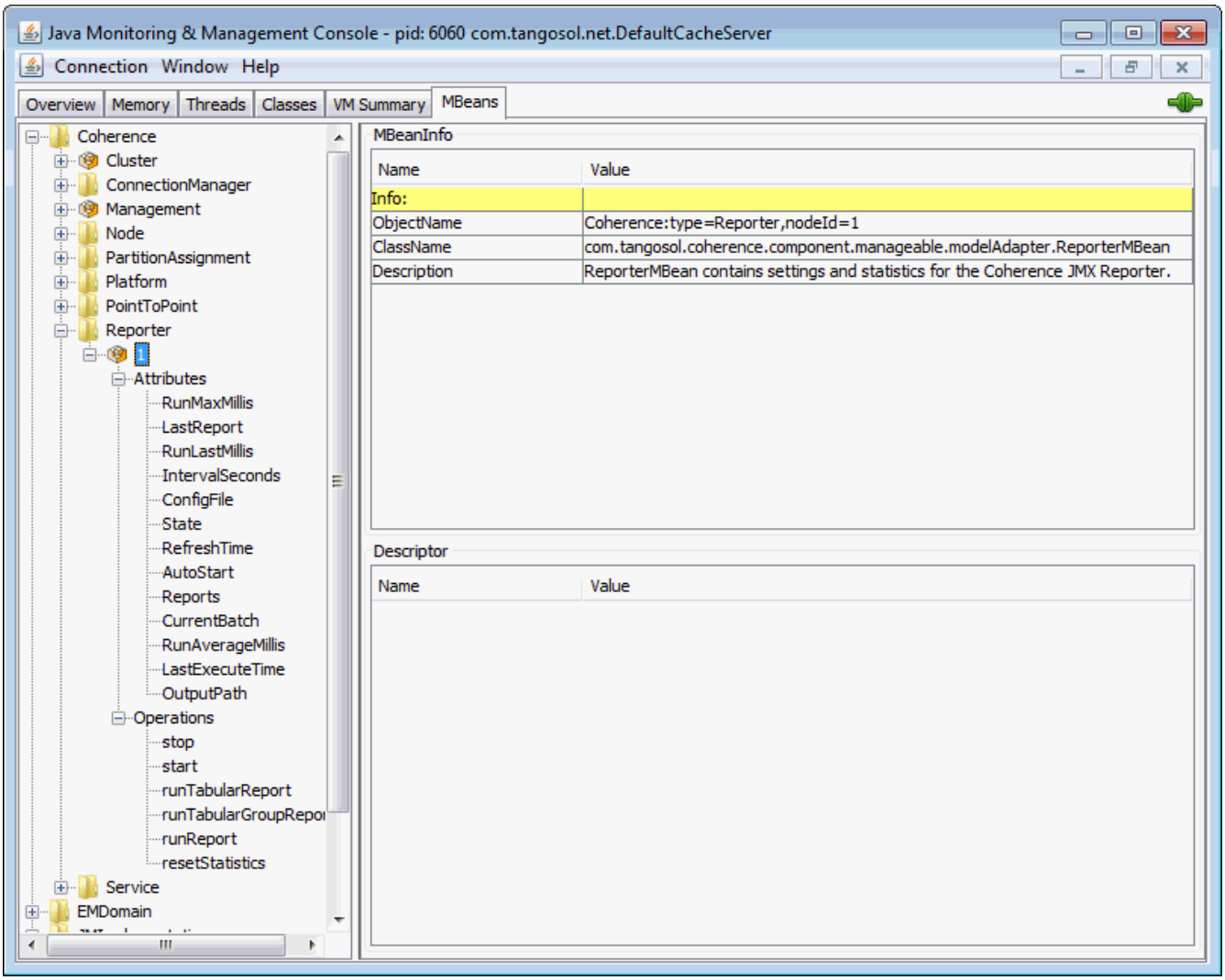
  Specify the full or relative path of a report configuration file or a report group configuration file that is located on the classpath. For example, enter `reports/report-memory-summary.xml` to run the memory summary report that is located in the `coherence.jar` file.

  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.

Figure 6-1 shows the `ReporterMBean` MBean attributes and operations within JConsole. See also Reporter MBean.

Figure 6-1 ReporterMBean MBean in JConsole



Running Oracle Coherence Reporting in a Distributed Configuration

Distributed reporting configures each member to log only local JMX statistics while allowing for centralized management of the reporters. A distributed configuration is intended for situations where cluster stability has been observed. In this configuration, the execution times of the reports on each cluster member 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.

To enable distributed configuration, start the managing member using the following system properties:

```
-Dcoherence.management.report.autostart=false
-Dcoherence.management.report.distributed=true
-Dcoherence.management=all
```

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

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

7

Creating Custom Reports

You can create your own reports in addition to using the predefined reports that are included with Coherence. Custom reports allow you to decide what management data to display and how the data is organized and processed.

This chapter includes the following sections:

Overview of Creating Custom Reports

Custom reports are created using MBeans that are registered in the Oracle Coherence MBean server (including custom MBeans). The MBeans allow different management data to be combined in meaningful ways that are specific to a Coherence solution. For example, you can create custom reports for use during testing to correlate data and monitor trends when troubleshooting and planning for production.

Report Configuration Files

> **Tip:**
>
> Extract the predefined reporting configuration files from the `coherence.jar/reports` directory before creating custom reports. Use the files as a starting point for creating custom reports.

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

To create a custom report, construct a report configuration file that is based on the `coherence-report-config.xsd` file. See 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 7-1 describes the elements that configure general report characteristics.

Table 7-1 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`.

This section includes the following topics:

- file-name Macros
- file-name Macro Examples

file-name Macros

Table 7-2 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 7-2 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.

Table 7-2 (Cont.) Macros for file-name Element

Macro	Description
`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 7-1 returns all registered `Service` MBeans. See query.

Example 7-1 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. See Using Filters in Reports. Example 7-2 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 7-2 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.

This section includes the following topics:

- How to Include an Attribute
- How to Include Part of the Key
- How to Include Information from Composite Attributes
- How to Include Information from Multiple MBeans
- Including Multiple MBean Information Example
- How to Use Report Macros
- How to Include Constant Values

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 7-3 illustrates how to include the `RoleName` attribute from the query pattern `Coherence:type=Node,*`.

Example 7-3 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 7-4 illustrates how to include the `nodeId` key part from the query pattern `Coherence:type=Node,*`.

Example 7-4 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 7-5 illustrates how to include the `startTime` of the `LastGCInfo` attribute from the `GarbageCollector` MBean.

Example 7-5 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 7-6 shows the configuration for this scenario.

Example 7-6 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 7-7 illustrates how to include the execution time into the report.

Example 7-7 Including Execution Time in a Report

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

Example 7-8 illustrates how to include the Report Batch/Count.

Example 7-8 Including the Report Batch/Count in a Report

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

Example 7-9 illustrates how to include the execution member.

Example 7-9 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 7-10 illustrates how to include a constant double of 1.0 in a report.

Example 7-10 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 7-11 illustrates how to include the constant string `dist-Employee` in a report.

Example 7-11 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 7-12 illustrates how to define an `equals` filter where `RoleRef` and `StringRef` are defined columns.

Example 7-12 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 7-13 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 7-13 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 7-14 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 7-14 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 7-15 illustrates how to define an `and` filter (assuming that all `column-ref` values are valid).

Example 7-15 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 7-16 illustrates how to define an `or` filter (assuming that all `column-ref` values are valid).

Example 7-16 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 7-17 illustrates how to define a `not equals` filter, where `RoleRef` and `StringRef` are defined columns.

Example 7-17 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.

This section includes the following topic:

- Function Examples

Function Examples

Example 7-18 illustrates how to add two column values (`Attribute1` and `Attribute2`) and place the results into a third column (`Addition`).

Example 7-18 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 7-19 illustrates how to subtract one column value (`Attribute2`) from another column value (`Attribute1`) and place the results into a third column (`Subtraction`).

Example 7-19 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 7-20 illustrates how to multiply two column values (`Attribute1` and `Attribute2`) and place the results into a third column (`Multiplication`).

Example 7-20 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 7-21 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 7-21 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 7-3 describes the available aggregate types.

Table 7-3 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 7-22 illustrates how to sum the values in the `size` column.

Example 7-22 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 7-23 illustrates how to average the values in the `size` column.

Example 7-23 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 7-24 illustrates how to find the maximum value in the `size` column.

Example 7-24 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 7-25 illustrates how to find the minimum value in the `size` column.

Example 7-25 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.

This section includes the following topic:

- Delta Function Examples

Delta Function Examples

Example 7-26 illustrates how to include a delta calculation of an attribute. (Assume that `PacketsSent` is a defined column.)

Example 7-26 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 7-27 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 7-27 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

To specify which reports to generate, create a report group configuration file that is based on the `coherence-report-group-config.xsd` file. See Report Group Configuration Elements. 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.
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. 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=
      "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=
      "coherence.reporter.output.directory">./</output-directory>
   ...
```

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

```
-Dcoherence.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

The reporter configuration includes the option to select a custom report group configuration file. All the reports that are listed in the report group configuration file are generated at run time.

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

The `coherence.management.report.autostart` and `coherence.management.report.configuration` system properties also enable reporting and configure a report group configuration file. For example:

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

8

Analyzing Report Contents

Oracle Coherence includes many predefined reports. Developers and system administrators use the reports to monitor and analyze operational statistics and troubleshoot potential problems.

An additional set of predefined reports are provided for Oracle Coherence*Web. The Coherence*Web reports are not discussed in this chapter. See Running Performance Reports in *Administering HTTP Session Management with 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 8-1 describes the contents of a cache size report.

Table 8-1 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 Storage Report

The cache storage report provides detailed metrics including index, querying, and eviction details for a cache. The name of the executor report is *`timestamp`*`-report-cache-storage.txt` where the timestamp is in `YYYYMMDDHH` format. For example, a file named `2009013101-report-proxy-connections.txt` represents an executor report for January 31, 2009 at 1:00 a.m.

> **Note:**
>
> This report is not included in `report-group.xml` but is available by running `report-all.xml`.

Table 8-2 describes the contents of the cache storage report.

Table 8-2 Contents of the Cache Storage Report

Attribute	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 Date	Date	A timestamp for each report refresh.
Service	String	The service name.
Cache	String	The cache name.
NodeId	String	The numeric member identifier.
Query Contention Count	Long	The total number of times a query had to be re-evaluated due to a concurrent update since the last report refresh. This statistic provides a measure of the impact of concurrent updates on query performance. If the total number of queries is *Q* and the number of contentions is *C*, then the expected performance degradation factor should be no more than (*Q* + *C*)/*Q*.
Eviction Count	Long	The total number of evictions for the cache across the cluster since the last report refresh.
Insert Count	Long	The number of inserts into the backing map since the last report refresh. 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 the data out.
Remove Count	Long	The number of removes from the backing map since the last report refresh. Removes are caused by operations such as clear, remove, or invoke.
Optimized Query Count	Long	The total number of queries that were fully resolved using indexes since the last report refresh.
Optimized Query Total Millis	Long	The total execution time, in milliseconds, for queries that were fully resolved using indexes since the last report refresh.
Non-optimized Query Count	Long	The total number of parallel queries that could not be resolved (or that were partially resolved) using indexes since the last report refresh.
Non-optimized Query Total Millis	Long	The total execution time, in milliseconds, for queries that could not be resolved (or that were partially resolved) using indexes since the last report refresh.
Indexing Total Millis	Long	The cumulative duration, in milliseconds, of index builds since the last report refresh. **Note:** This attribute is available only from 14.1.1.2206.
Index Total Units	Long	The total units used by all indices on the associated cache.

Table 8-2 (Cont.) Contents of the Cache Storage Report

Attribute	Type	Description
`Max Query Threshold Millis`	`Long`	The query statistics threshold, defining when a query has been running long enough to be interesting for recording.
`Max Query Duration Millis`	`Long`	The duration, in milliseconds, of the longest query execution since statistics were last reset.
`MaxQuery Description`	`String`	A string representation of a query with the longest execution time exceeding the `MaxQuery Threshold Millis` attribute's value since statistics were last reset.
`Optimized Query Average Millis`	`Long`	The average execution time, in milliseconds, for queries that were fully resolved using indexes since statistics were last reset.
`Non-optimized Query Average Millis`	`Long`	The average execution time, in milliseconds, for queries that could not be resolved (or that were partially resolved) using indexes since statistics were last reset.
`Locks Granted`	`Long`	The number of locks currently granted for the portion of the partitioned cache managed by `StorageManager`.
`Locks Pending`	`Long`	The number of pending lock requests for the portion of the partitioned cache managed by the `StorageManager`.
`Listener Key Count`	`Long`	The number of key-based listeners registered with the `StorageManager`.
`Listener Filter Count`	`Long`	The number of filter-based listeners registered with the `StorageManager`.
`Listener Registration Count`	`Long`	The number of listener registrations since the last report refresh.

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 8-3 describes the contents of the cache usage report.

Table 8-3 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`.

Table 8-3 (Cont.) Contents of the Cache Usage Report

Column	Data Type	Description
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 on 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 on 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 on get() invocation that is a hit across the cluster since the last report refresh.
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 on 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 Executor Report

The executor report provides information about the running executors for a cluster.The name of the executor report is *timestamp*-executors.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013101-executors.txt represents an executor report for January 31, 2009 at 1:00 a.m.
Table 8-4 describes the contents of the executor report.

Table 8-4 Contents of the Executor Report

Attribute	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.
ReportData	Date	A timestamp for each report refresh.
Name	String	The logical name of the executor.
Id	String	The unique ID of this executor.
MemberId	Long	The location where the executor is running.
State	String	The state of the executor.
Description	String	The description of this executor. The following options are available: • JOINING - Indicates that the executor is joining the orchestration. • RUNNING - Indicates that the executor is accepting and executing the tasks. • CLOSING_GRACEFULLY - Indicates that the executor has commenced graceful closing. No new tasks will be accepted, but existing ones will run to completion. • CLOSING - Indicates that the executor has commenced closing. • CLOSED - Indicates that the executor is closed. • REJECTING - Indicates that the executor is rejecting tasks.
TaksInProgressCount	Integer	The in progress task count.
TasksCompleted	Integer	The completed tasks count for the time period.
TasksRejectedCount	Integer	The rejected tasks count for the time period.

Understanding the Federation Destination Report

The federation destination report indicates out-going replication statistics from the perspective of a federation participant who receives replicated data. The name of the federation destination report is *timestamp*-federation-destination.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013101-federation-destination.txt represents a report for January 31, 2009 at 1:00 a.m.
Table 8-5 describes the contents of a federation destination report.

Table 8-5 Contents of the Federation Destination 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 8-5 (Cont.) Contents of the Federation Destination Report

Column	Data Type	Description
`Report Time`	`Date`	A timestamp for each report refresh.
`Node ID`	`Long`	The member for the federation statistics.
`Name`	`String`	The name of the sender.
`State`	`String`	The state of the participant. For example: `INITIAL`, `IDLE`, `CONNECTING`, `STOPPED`, `PAUSED`, `ERROR`, or `YIELDING`
`Status`	`Integer`	The status of the participant. One of the following: • `0` – Ok • `1` – Warning • `2` – Error
`CurrentBandwidth`	`Double`	The current utilized bandwidth in Megabits per second for sending replicate message.
`TotalBytesSent`	`Long`	The total number of bytes that were sent.
`TotalEntriesSent`	`Long`	The total number of cache entries that were sent.
`TotalRecordsSent`	`Long`	The total number of journal records that were sent. A journal record can consist of multiple cache entries that are part of the same transaction.
`TotalMsgSent`	`Long`	The total number of replication messages that were sent. A replication message can contain multiple journal records.
`TotalMsgUnacked`	`Long`	The total number of un-acknowledged replication messages.
`RecordBacklogDelayTimePercentileMillis`	`Long`	The 90-percentile value of the time (in milliseconds) the journal records are in the cache waiting to be replicated.
`MsgNetworkRoundTripTimePercentileMillis`	`Long`	The 90-percentile value of the round trip time (in milliseconds) taken for the transmission of replication messages, application of changes on the destination cluster, and the receipt of the corresponding acknowledgment messages over the network.
`MsgApplyTimePercentileMillis`	`Long`	The 90-percentile value of the time (in milliseconds) it took to apply the replication messages on the destination.
`BytesSent/sec`	`Long`	The bytes sent per second.
`MsgsSent/sec`	`Long`	The messages sent per second.
`MaxBandwidth`	`Double`	The maximum bandwidth in megabits per second for sending replicate messages. A value of `-1.0` means the maximum bandwidth is not configured.

Table 8-5 (Cont.) Contents of the Federation Destination Report

Column	Data Type	Description
ErrorDescription	String	An error description. A value is only returned if the sender is in an ERROR state.
SendTimeoutMillis	Long	The send timeout that is configured for the participant.
GeoIp	String	The location metadata that is configured for the participant.
ReplicateAllPercentComplete	Long	The percent of work completed for a replicateAll request.
ReplicateAllTotalTime	Long	The total time (in milliseconds) the replicateAll request took.
ReplicateAllPartitionCount	Integer	The total number of partitions in the replicateAll request.
ReplicateAllPartitionCompletedCount	Integer	The total number of partitions completed for the replicateAll request.
ReplicateAllPartitionErrorCount	Integer	The number of partitions with error responses for thereplicateAll request. Check ReplicateAllLastException in the DestinationMBean for details of the last exception and the log for details of all the errors.

Understanding the Federation Origin Report

The federation origin report indicates in-coming replication statistics from the perspective of a federation participant who sends replicated data. The name of the federation origin report is *timestamp*-federation-origin.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013101-federation-origin.txt represents a report for January 31, 2009 at 1:00 a.m.
Table 8-6 describes the contents of a federation origin report.

Table 8-6 Contents of the Federation Origin 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 federation statistics.
TotalBytesReceived	Long	The total number of bytes that were received.
TotalRecordsReceived	Long	The total number of journal records that were received. A journal record could consist of multiple cache entries that are part of the same transaction.

Table 8-6 (Cont.) Contents of the Federation Origin Report

Column	Data Type	Description
TotalEntriesReceived	Long	The total number of cache entries that were received.
TotalMsgReceived	Long	The total number of replication messages that were received. A replication message could contain multiple journal records.
TotalMsgUnacked	Long	The total number of un-acknowledged replication messages.
MsgApplyTimePercentileMillis	Long	The 90-percentile value of the time (in milliseconds) it took to apply the replication messages on the destination.
RecordBacklogDelayTimePercentileMillis	Long	The 90-percentile value of the time (in milliseconds) the journal records are in the cache waiting to be replicated.
BytesReceived/sec	Long	The bytes received per second.
MsgsReceived/sec	Long	The messages received per second.

Understanding the Federation Status Report

The cache size report indicates the status for a federation participant. The name of the federation status report is *timestamp*-federation-status.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013101-federation-status.txt represents a cache size report for January 31, 2009 at 1:00 a.m.
Table 8-7 describes the contents of a federation status report.

Table 8-7 Contents of the Federation Status Report

Column	Data Type	Description
Node ID	Long	The member for the federation statistics.
Name	String	The name of the sender.
State	String	The state of the participant. One of: • INITIAL • IDLE • READY • SENDING • CONNECTING • CONNECT_WAIT • STOPPED • PAUSED • ERROR • YIELDING • BACKLOG_EXCESSIVE • BACKLOG_NORMAL • DISCONNECTED

Table 8-7 (Cont.) Contents of the Federation Status Report

Column	Data Type	Description
Error Description	String	An error description. A value is only returned if the sender is in an ERROR state.

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 8-8 describes the contents of the flash journal report.

Table 8-8 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.
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 8-9 describes the contents of the JCache configuration report.

Table 8-9 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 8-10 describes the contents of the JCache statistics report.

Table 8-10 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.

Table 8-10 (Cont.) Contents of the JCache Statistics Report

Column	Data Type	Description
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 8-11 describes the contents of the management report.

Table 8-11 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. 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 8-12 describes the contents of the memory status report.

Table 8-12 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.
Gc Name	String	The name of the garbage collector.

Table 8-12 (Cont.) Contents of the Memory Status Report

Column	Data Type	Description
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.
HeapCommittedMB	Long	The heap in megabytes committed at the time the report ran.
HeapInitMB	Long	The heap in megabytes initialized at the time the report ran.
HeapMaxMB	Long	The maximum number of megabytes used by the JVM since its start.
HeapUsedMB	Long	The heap in megabytes 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 8-13 describes the contents of the network health detail report.

Table 8-13 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.
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 8-14), 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 Rx Success for the batch in the Network Health File (see Table 8-14), 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.
TransportSentMessages	Long	The number of messages sent by the service-dedicated transport since the last report refresh.

Table 8-13 (Cont.) Contents of the Network Health Detail Report

Column	Data Type	Description
TransportReceivedMessages	Long	The number of messages received by the service-dedicated transport since the last report refresh.
TransportBacklogDelay	Long	The total number of milliseconds that the requests were delayed due to draining the backlog 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 8-14 describes the contents of the network health report.

Table 8-14 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.
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 member-identity in *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 8-15 describes the contents of the node list report.

Table 8-15 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 Persistence Detail Report

The persistence report provides detailed information about how cache persistence is performing for a particular service and node. The name of the persistence detail report is *timestamp*-persistence-detail.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013101-persistence-detail.txt represents a persistence detail report for January 31, 2009 at 1:00 a.m.
Table 8-16 describes the contents of the persistence detail report.

Table 8-16 Contents of the Persistence 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.
ReportTime	Date	A timestamp for each report refresh.
Service	String	The name of the partitioned cache service.

Table 8-16 (Cont.) Contents of the Persistence Detail Report

Column	Data Type	Description
PersistenceMode	String	The current persistence mode for this service: • active – all mutating cache operations are persisted using the configured persistence environment. • on-demand – a persistence environment has been configured and is available but is not being actively used. • n/a – persistence is not configured for this service.
NodeId	Long	The member for the persistence statistics.
PersistenceLatencyAverage	Float	The average latency (in milliseconds) added to a mutating cache operation by active persistence operations.
PersistenceLatencyMax	Long	The maximum latency (in milliseconds) added to a mutating cache operation by an active persistence operation.
PersistenceActiveSpaceUsed	Long	The amount of space (in bytes) that is used by active persistence.
PersistenceActiveSpaceTotal	Long	The total size (in bytes) of the file system for use by active persistence.
PersistenceActiveSpaceAvailable	Long	The remaining space (in bytes) available on the file system for active persistence.
PersistenceSnapshotSpaceTotal	Long	The total size (in bytes) of the file system to store snapshots.
PersistenceSnapshotSpaceAvailable	Long	The remaining space (in bytes) available on the file system to store snapshots.
PersistenceBackupSpaceUsed	Long	The total size (in bytes) used by the persistence layer to persist backup cache data.
PersistenceBackupSpaceTotal	Long	The total size (in bytes) of the file system used by the persistence layer to persist backup cache data.
PersistenceBackupSpaceAvailable	Long	The total remaining free space (in bytes) of the file system used by the persistence layer to persist backup cache data.

Understanding the Persistence Report

The persistence report provides information about how cache persistence is performing for a particular service. The name of the persistence report is *timestamp*-persistence.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013101-persistence.txt represents a persistence report for January 31, 2009 at 1:00 a.m. Table 8-17 describes the contents of the persistence report.

Table 8-17 Contents of the Persistence 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.
Service	String	The name of the partitioned cache service.
PersistenceMode	String	The current persistence mode for this service: • active – all mutating cache operations are persisted using the configured persistence environment. • on-demand – a persistence environment has been configured and is available but is not being actively used. • n/a – persistence is not configured for this service.
PersistenceTotalActiveSpaceUsed	Long	The amount of space (in bytes) that is used by active persistence
PersistenceLatencyAverageAcrossNodes	Float	The average latency for all nodes (in milliseconds) added to a mutating cache operation by active persistence operations.
PersistenceLatencyMaxAcrossNodes	Long	The maximum latency for all nodes (in milliseconds) added to a mutating cache operation by an active persistence operation.
PersistenceBackupSpaceUsed	Long	The total size (in bytes) used by the persistence layer to persist backup cache data.

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 8-18 describes the contents of the proxy report.

Table 8-18 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.

Table 8-18 (Cont.) Contents of the Proxy Report

Column	Data Type	Description
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 Proxy Connections Report

The proxy connections report provides information about the client connections for proxy servers in a cluster. The name of the proxy connections report is *timestamp*-report-proxy-connections.txt, where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013101-report-proxy-connections.txt represents a proxy connections report for January 31, 2009 at 1:00 a.m.
Table 8-19 describes the contents of the proxy connections report.

Table 8-19 Contents of the Proxy Connections Report

Attribute	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.
ReportDate	Date	The timestamp for each report refresh.
ServiceName	String	The service name of the proxy server.
NodeId	String	The numeric member identifier.
UUID	Integer	The unique ID of this client connection.
RemoteAddress	String	The remote address of the client.
RemotePort	Integer	The remote port of the client.

Table 8-19 (Cont.) Contents of the Proxy Connections Report

Attribute	Type	Description
ClientAddress	String	The client address of the client (may be different from the RemoteAddress depending upon the load balancer configuration).
ClientProcessName	String	The name or process id of the client.
ClientRole	String	The role of the client.
ConnectionTimeMillis	Long	The time in milliseconds that the client has been connected.
OutgoingByteBacklog	Long	The number of bytes queued to be sent to the client.
OutgoingMessageBacklog	Long	The number of messages queued to be sent to the client.
BytesSent	Long	The number of bytes sent to the client since the last report refresh.
BytesReceived	Long	The number of bytes received from the client since the last report refresh.
MessagesSent	Long	The number of messages sent to the client since the last report refresh.
MessagesReceived	Long	The number of messages received from the client since the last report refresh.
Member	String	The string representation of the client.

Understanding the Proxy HTTP Report

The proxy HTTP report provides information about HTTP acceptors that are configured on a proxy server. The name of the proxy report is *timestamp*-report-proxy-http.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013101-report-proxy-http.txt represents a proxy report for January 31, 2009 at 1:00 a.m.
Table 8-20 describes the contents of the proxy HTTP report.

Table 8-20 Contents of the Proxy HTTP 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.
HTTPServerType	String	The type of HTTP server or n/a if the HTTP protocol is not used.

Table 8-20 (Cont.) Contents of the Proxy HTTP Report

Column	Data Type	Description
HostIp	String	The IP Address and Port of the proxy service.
NodeId	String	The numeric member identifier.
AverageRequestSize	Float	The average size of the HTTP requests.
AverageResponseSize	Float	The average size of the HTTP responses.
AverageRequestTime	Float	The average processing time in milliseconds for HTTP requests.
TotalErrorCount	Long	The number of HTTP requests that caused errors.
TotalRequestCount	Long	The number of requests since the HTTP server was started or the statistics were reset.
ResponseCount1xx	Long	The number of HTTP responses in the 100-199 range.
ResponseCount2xx	Long	The number of HTTP responses in the 200-299 range.
ResponseCount3xx	Long	The number of HTTP responses in the 300-399 range.
ResponseCount4xx	Long	The number of HTTP responses in the 400-499 range.
ResponseCount5xx	Long	The number of HTTP responses in the 500-599 range.

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 8-21 describes the contents of the ram journal report.

Table 8-21 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.
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 8-22 describes the contents of the service report.

Table 8-22 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.
`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.
`Owned Partitions Primary`	`Integer`	The number of partitions that this member owns (responsible for the primary storage).
`Owned Partitions Backups`	`Integer`	The number of partitions that this member backs up (responsible for the backup storage).
`Partitions Endangered`	`Integer`	The total number of partitions that are not currently backed up.

Table 8-22 (Cont.) Contents of the Service Report

Column	Data Type	Description
`Partitions Vulnerable`	`Integer`	The total number of partitions that are backed up on the same machine where the primary partition owner resides.
`Partitions Unbalanced`	`Integer`	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.
`Outgoing Partition Count`	`Integer`	The number of partitions that are currently being transferred by this service member to other members.
`Thread Count`	`Integer`	The number of threads in the service thread pool. To configure a thread count, set the `thread-count-min` and `thread-count-max` elements to the same value.
`Thread Idle Count`	`Integer`	The number of currently idle threads in the service thread pool.
`Thread Utilization %`	`Double`	The percentage of threads in the pool that are being used. The percentage is calculated based on the thread count and the number of threads that are idle.

Understanding the Service Partitions Report

The service partitions report provides detailed metrics, including partition count, average and maximum partition sizes, and average and maximum storage sizes for a service. The name of the service partitions report is *`timestamp`*`-service-partitions.txt`, where *`timestamp`* is in the YYYYMMDDHH format. For example, a file named `2009013101-service-partitions.txt` represents an executor report for January 31, 2009 at 1:00 a.m.

> **Note:**
>
> This report is not included in `report-group.xml` but is available by running `report-all.xml` after you have installed the Cumulative Patch Update (CPU) 36410371 or later.

Table 8-23 describes the contents of the service partitions report.

Table 8-23 Contents of the Service Partitions Report

Attribute	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 Date`	`Date`	A time stamp for each report refresh.

Table 8-23 (Cont.) Contents of the Service Partitions Report

Attribute	Type	Description
Service	String	The name of the service.
Partition Count	Integer	The configured number of partitions for the service.
Backup Count	Integer	The configured number of partition backups to be maintained by the service.
Fair Share Primary	Integer	The number of primary partitions per storage-enabled service member that a strategy attempts to maintain.
Fair Share Backup	Integer	The number of backup partitions per storage-enabled service member that a strategy currently attempts to maintain.
Service Node Count	Integer	The number of storage-enabled nodes running the service.
Service Machine Count	Integer	The number of machines that host storage-enabled nodes running the service.
Service Rack Count	Integer	The number of racks that host storage-enabled nodes running the service.
Service Site Count	Integer	The number of sites that host storage-enabled nodes running this service.
Strategy Name	String	The name of the partition assignment strategy that is in use.
HA Status	String	The high availability status for the service. The valid values are: • MACHINE-SAFE: The loss of a machine will not result in data loss. • RACK-SAFE: The loss of a rack will not result in data loss. • SITE-SAFE: The loss of a site will not result in data loss. • NODE-SAFE: A cluster node can be stopped without any data loss. • ENDANGERED: Abnormal termination of a cluster node that runs the service can cause data loss. **Note**: To achieve RACK-SAFE and SITE-SAFE, each cluster node must have its rack and site name configured.

Table 8-23 (Cont.) Contents of the Service Partitions Report

Attribute	Type	Description
HA Target	String	The high availability status that the strategy attempts to achieve. Valid values are the same as the HAStatus attribute.
Remaining Distribution Count	Integer	The number of partition transfers that remain to be completed before the service achieves the goals set by the strategy.
Average Partition Size KB	Integer	The average partition storage size in kilobytes.
Max Partition Size KB	Integer	The maximum partition storage size in kilobytes.
Average Storage Size KB	Integer	The average node storage size in kilobytes.
Max Storage Size KB	Integer	The maximum node storage size in kilobytes.
Max Load Node Id	Integer	The node identified with the maximum node storage size.

Understanding the Topic Report

The Topic report provides detailed metrics for Topics defined within a cluster. The name of the Topic report is *timestamp*-topic.txt where the *timestamp* is in the YYYYMMDDHH format. For example, a file named 2009013101-topics.txt represents a Topic report for January 31, 2009 at 1:00 a.m.

> **Note:**
>
> This report is not included in report-group.xml but is available by running report-all.xml after you have installed the Cumulative Patch Update (CPU) 35122413 or later.

Table 8-24 describes the contents of the Topic report.

Table 8-24 Contents of the Topic Report

Attribute	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 Date	Date	A time stamp for each report refresh.

Table 8-24 (Cont.) Contents of the Topic Report

Attribute	Type	Description
Service	String	The name of the service.
Name	String	The name of the Topic.
NodeId	String	The numeric member identifier.
ChannelCount	Integer	The number of channels in the Topic.
PublishedCount	Long	The number of published messages since the last report refresh.
PublishedFifteenMinuteRate	Double	The rate of publishing messages over the last 15 minutes.
PublishedFiveMinuteRate	Double	The rate of publishing messages over the last five minutes.
PublishedOneMinuteRate	Double	The rate of publishing messages over the last one minute.
PublishedMeanRate	Double	The mean rate at which messages are published.

Understanding the Topic Subscribers Report

The Topic Subscribers report provides detailed metrics for Topic Subscribers defined within a cluster. The name of the Topic Subscribers report is *timestamp*-topic-subscribers.txt where the *timestamp* is in the YYYYMMDDHH format. For example, a file named 2009013101-topic-subscribers.txt represents a Topic Subscribers report for January 31, 2009 at 1:00 a.m.

> **Note:**
>
> This report is not included in report-group.xml but is available by running report-all.xml after you have installed the Cumulative Patch Update (CPU) 35122413 or later.

Table 8-25 describes the contents of the Topic Subscribers report.

Table 8-25 Contents of the Topic Subscribers Report

Attribute	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 Date	Date	A time stamp for each report refresh.

Table 8-25 (Cont.) Contents of the Topic Subscribers Report

Attribute	Type	Description
Service	String	The name of the service.
Name	String	The name of the Topic.
SubscriberGroup	String	The subscriber group to which the subscriber belongs.
Id	Long	The Id of the subscriber.
NodeId	String	The numeric member identifier.
Backlog	Long	The number of outstanding receive requests.
ChannelAllocations	String	The subscriber's allocated channels.
ChannelCount	Integer	The number of channels in the Topic.
Disconnections	Long	The number of times the subscriber has disconnected since the last report refresh.
Notifications	Long	The number of channel notifications received since the last report refresh.
Polls	Long	The total number of polls for messages since the last report refresh.
ReceiveCompletionsCount	Long	The number completed receive requests since the last report refresh.
ReceiveCompletionsFifteenMinuteRate	Double	The rate at which receive requests are completed over the last 15 minutes.
ReceiveCompletionsFiveMinuteRate	Double	The rate at which receive requests are completed over the last five minutes.
ReceiveCompletionsOneMinuteRate	Double	The rate at which receive requests are completed over the last one minute.
ReceiveCompletionsMeanRate	Double	The mean rate at which receive requests are completed.
ReceiveEmpty	Long	The number of empty receive requests since the last report refresh.
ReceiveErrors	Long	The number of exceptionally completed receive requests since the last report refresh.
ReceivedCount	Long	The number of elements received since the last report refresh.

Table 8-25 (Cont.) Contents of the Topic Subscribers Report

Attribute	Type	Description
State	Integer	The state of the subscriber. Valid values are: • 0 - Initial • 1 - Connected • 2 - Disconnected • 3 - Closing • 4 - Closed
StateName	String	The state of the subscriber as a string.
Waits	Long	The number of elements received since the last report refresh.

Understanding the Topic Subscriber Groups Report

The Topic Subscriber Groups report provides detailed metrics for Topic Subscriber Groups defined within a cluster. The name of the Topic Subscriber Groups report is *timestamp*-topic-subscriber-groups.txt where the *timestamp* is in the YYYYMMDDHH format. For example, a file named 2009013101-topic-subscriber-groups.txt represents a Topic Subscriber Groups report for January 31, 2009 at 1:00 a.m.

> **Note:**
>
> This report is not included in report-group.xml but is available by running report-all.xml after you have installed the Cumulative Patch Update (CPU) 35122413 or later.

Table 8-26 describes the contents of the Topic Subscriber Groups report.

Table 8-26 Contents of the Topic Subscriber Groups Report

Attribute	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 Date	Date	A time stamp for each report refresh.
Service	String	The name of the service.
Topic	String	The name of the Topic.
Name	String	The name of the subscriber group.
NodeId	String	The numeric member identifier.

Table 8-26 (Cont.) Contents of the Topic Subscriber Groups Report

Attribute	Type	Description
ChannelCount	Integer	The number of channels in the Topic.
PolledCount	Long	The total number of polled messages since the last report refresh.
PolledFifteenMinuteRate	Double	The rate at which messages are polled over the last 15 minutes.
PolledFiveMinuteRate	Double	The rate at which messages are polled over the last five minutes.
PolledOneMinuteRate	Double	The rate at which messages are polled over the last one minute.
PolledMeanRate	Double	The mean rate at which messages are polled.

Understanding the Transaction Manager Report

The transaction manager report provides detailed transaction manager statics from all transaction service instances in a cluster. The name of the transaction manager report is *timestamp*-report-transaction.txt where the *timestamp* is in YYYYMMDDHH format. For example, a file named 2009013101-report-transaction.txt represents a transaction report for January 31, 2009 at 1:00 a.m.
Table 8-27 describes the contents of the transaction manager report.

Table 8-27 Contents of the Transaction Manager Report

Attribute	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 Date	Date	A timestamp for each report refresh.
Service	String	The name of the service.
NodeId	String	The numeric member identifier.
TotalActive	Long	The total number of currently active transactions. An active transaction is counted as any transaction that contains at least one modified entry and is 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.

Table 8-27 (Cont.) Contents of the Transaction Manager Report

Attribute	Type	Description
TimeoutMillis	Long	The transaction timeout value in milliseconds. This value applies only to transactional connections obtained after the value is set. This attribute is currently not supported.
CommittedCount	Long	The total number of transactions that have been committed by the transaction manager for the time period. The coordinator member for this transaction maintains the count even though multiple members may have participated in the transaction.
RecoveredCount	Long	The total number of transactions that have been recovered by the transaction manager for the time period. The coordinator member for this transaction maintains the count even though multiple members may have participated in the transaction.
RolledbackCount	Long	The total number of transactions that have been rolled back by the transaction manager for the time period. The coordinator member for this transaction maintains the count even though multiple members may have participated in the transaction.
TransactionMillis	Long	The cumulative time (in milliseconds) that was spent on active transactions for the time period.

Understanding the View Report

The view report provides information about the views defined for a cluster.The name of the view report is *timestamp*-view-usage.txt where the timestamp is in YYYYMMDDHH format. For example, a file named 2009013101-view-usage.txt represents a view report for January 31, 2009 at 1:00 a.m.
Table 8-28 describes the contents of the view report.

Table 8-28 Contents of the View Report

ViewMBean Attribute	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.
ReportData	Date	A timestamp for each report refresh.
Service	String	The service to which the view cache belongs.
NodeId	String	The numeric member identifier.
Name	String	The name of the view cache.
Size	Long	The number of entries in the view-cache.

A

Oracle Coherence MBeans Reference

The MBean reference provides a detailed description of all Coherence MBeans including their attributes and operations. In addition to the MBean reference, the `Registration` interface provides descriptions of the MBeans. In addition, an MBean-capable agent (such as JConsole) also shows MBean information. See Using JMX to Manage Oracle Coherence .
This reference includes the following MBeans:

- Cache MBean
- Cluster MBean
- ClusterNode MBean
- ConnectionManager MBean
- Connection MBean
- Destination MBean
- Executor MBean
- FederationManager MBean
- FlashJournalRM MBean
- Health MBean
- Management MBean
- Origin MBean
- PagedTopic MBean
  The `PagedTopic` MBean provides statistics for Topic services running in a cluster. A cluster contains zero or more instances of this MBean, each instance representing an instance of a Topic on a member.
- PagedTopicSubscriber MBean
  The `PagedTopicSubscriber` MBean provides statistics for Topic Subscribers running in a cluster. A cluster contains zero or more instances of this MBean, each instance representing an instance of a Topic Subscriber on a member.
- PagedTopicSubscriberGroup MBean
  The `PagedTopicSubscriberGroup` MBean provides statistics for Topic Subscriber Groups running in a cluster. A cluster contains zero or more instances of this MBean, each instance representing an instance of a Topic Subscriber Group on a member.
- PersistenceManager MBean
- PointToPoint MBean
- RamJournalRM
- Reporter MBean
- Service MBean
- SimpleStrategy MBean
- StorageManager MBean
- Topology MBean

Cache MBean

The `Cache` MBean represents a cache and provides operational and performance statistics. Some MBean attributes are writable and allow the behavior of a cache to be changed in real time. 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).
The object name of the MBean is:

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

> **Note:**
>
> The `Cache` 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 Cache MBeans for Transactional Caches in *Developing Applications with Oracle Coherence*.

Attributes

Table A-1 describes the attributes for the `Cache` MBean.

Table A-1 Cache MBean Attributes

Attribute	Type	Access	Description
`AverageGetMillis`	`Double`	read-only	The average number of milliseconds for each `get()` invocation since the last time statistics were reset. This statistic is only tracked for caches that may incur expensive hits (for example the front of a near cache).
`AverageHitMillis`	`Double`	read-only	The average number of milliseconds for each `get()` invocation that is a hit. This statistic is only tracked for caches that may incur expensive hits (for example the front of a near cache).
`AverageMissMillis`	`Double`	read-only	The average number of milliseconds for each `get()` invocation that is a miss. This statistic is only tracked for caches that may incur expensive hits (for example the front of a near cache).
`AveragePutMillis`	`Double`	read-only	The average number of milliseconds for each `put()` invocation since the cache statistics were last reset. This statistic is only tracked for caches that may incur expensive hits (for example the front of a near cache).

Table A-1 (Cont.) Cache MBean Attributes

Attribute	Type	Access	Description
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.
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. This statistic is only tracked for caches that may incur expensive hits (for example the front of a near cache).
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. This statistic is only tracked for caches that may incur expensive hits (for example the front of a near cache).
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
CacheStoreType	String	read-only	The cache store type for this cache. Possible values include: NONE, READ-ONLY, WRITE-THROUGH, WRITE-BEHIND.
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.

Table A-1 (Cont.) Cache MBean Attributes

Attribute	Type	Access	Description
`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
`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 cache store type is `WRITE-BEHIND`.
`QueueSize`	`Integer`	read-only	The size of the write-behind queue. This attribute is only applicable if the cache store 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.
`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 cache store type is `NONE`.

Table A-1 (Cont.) Cache MBean Attributes

Attribute	Type	Access	Description
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 cache store 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 cache store 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 cache store 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 cache store 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 cache store 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 cache store 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 cache store 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. This statistic is only tracked for caches that may incur expensive hits (for example the front of a near cache).
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. This statistic is only tracked for caches that may incur expensive hits (for example the front of a near cache).
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 `Cache` MBean includes a `resetStatistics` operation that resets all cache statistics.

Cluster MBean

The `Cluster` MBean represents a cluster and provides operational statistics. Each cluster member includes a single instance of this managed bean.
The object name of the MBean is:

```
type=Cluster
```

Attributes

Table A-2 describes the attributes for `Cluster` MBean.

Table A-2 Cluster MBean 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-3 describes the operations for the `Cluster` MBean.

Table A-3 Cluster MBean Operations

Operation	Parameters	Return Type	Description
`suspendService`	`String sService`	`Void`	Suspends all members of the service identified by the specified name. A suspended service is in a deactivated state in preparation to be shutdown. Once suspended, a service can be resumed or reactivated with the `resumeService` operation. If `Cluster` is used as the service name, then all services (including the cluster service) are suspended.
`shutdown`	Not applicable	`Void`	Shuts down the cluster service on this member.
`resumeService`	`String sService`	`Void`	Resumes all suspended members of the service identified by the specified name. If `Cluster` is used as the service name, then all services (including the cluster service) are resumed.
`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 Specifying a Cluster Member's Identity in *Developing Applications with Oracle Coherence*. If the `sRole` parameter is not specified (no value in the field), then state is logged on all cluster members.
`ensureRunning`	Not applicable	`Void`	Ensures that the cluster service is running on this member.
`configureTracing`	`String sRole, float fRatio`	`Void`	Configures the tracing sampling ratio for those members matching the specified role, or if sRole is null or empty, configures the tracing ratio for all cluster members. Valid values for fRatio are -1 and 0 through 1.0 (inclusive).
`dumpClusterHeap`	`String sRole`	`Void`	Creates a heap dump for each cluster member matching the specified role. If null is specified as the role, the heap dump will be created for all the cluster members.

ClusterNode MBean

The `ClusterNode` MBean represents a cluster member and provides operational and performance statistics. Many of the attributes are writable and allow the behavior of the member to be changed in real time. Each cluster member includes a single instance of this managed bean.
The object name of the MBean is:

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

Attributes

Table A-4 describes the attributes for `ClusterNode` MBean.

Table A-4 ClusterNode MBean 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 on which this member is running.
`FlowControlEnabled`	`Boolean`	read-only	Indicates whether `FlowControl` is enabled. See flow-control 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`, `log4j2`, 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.
`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.

Table A-4 (Cont.) ClusterNode MBean Attributes

Attribute	Type	Access	Description
`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.
`Priority`	`Integer`	read-only	The priority, or weight, of the member. This value is used to determine tie-breakers.

Table A-4 (Cont.) ClusterNode MBean Attributes

Attribute	Type	Access	Description
`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.

Table A-4 (Cont.) ClusterNode MBean Attributes

Attribute	Type	Access	Description
RefreshTime	Date	read-only	The timestamp when this model was last retrieved from a corresponding member. For local servers, it is the local time.
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 on which this member is running.
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.

Table A-4 (Cont.) ClusterNode MBean Attributes

Attribute	Type	Access	Description
TracingSamplingRatio	Float	read/write	The ratio of spans to capture when tracing is enabled. A value of -1 disables tracing. A value of 0 means Coherence will not initiate tracing unless an active span is already present. A value between 0 (exclusively) and 1.0 (inclusively) represents the percentage of tracing spans that will be captured.
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.
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-5 describes the operations for the `ClusterNode` MBean.

Table A-5 ClusterNode MBean 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).

ConnectionManager MBean

The `ConnectionManager` MBean represents an Oracle Coherence*Extend proxy and provides statistics for throughput and connection information. A cluster member includes zero or more instances of this managed bean depending on the number of configured proxies.
The object name of the MBean is:

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

Attributes

Table A-6 describes the attributes for the `ConnectionManager` MBean.

Table A-6 ConnectionManager MBean Attributes

Attribute	Type	Access	Description
`AverageRequestTime`	`Float`	read-only	The average processing time in milliseconds for HTTP requests
`ConnectionCount`	`Integer`	read-only	The number of client connections
`HostIP`	`String`	read-only	The IP address and port of the proxy host
`HTTPServerType`	`String`	read-only	The type of HTTP server or `n/a` if the HTTP protocol is not used
`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
`MessagingDebug`	`Boolean`	read/write	The debug flag. If the flag set to `true` and the logging level on the node is set to 6 or higher, then sent and received messages are logged for all the connections under this service.
`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
`Protocol`	`String`	read-only	The protocol associated with this `ConnectionManager` MBean instance. Valid values are `tcp` or `http`.
`RefreshTime`	`Date`	read-only	The timestamp when this model was last retrieved from a corresponding member. For local servers, it is the local time.
`RequestsPerSecond`	`Float`	read-only	The number of HTTP requests per second since the statistics were reset
`ResponseCount1xx`	`Long`	read-only	The number of HTTP responses in the 100-199 range
`ResponseCount2xx`	`Long`	read-only	The number of HTTP responses in the 200-299 range
`ResponseCount3xx`	`Long`	read-only	The number of HTTP responses in the 300-399 range

Table A-6 (Cont.) ConnectionManager MBean Attributes

Attribute	Type	Access	Description
ResponseCount4xx	Long	read-only	The number of HTTP responses in the 400-499 range
ResponseCount5xx	Long	read-only	The number of HTTP responses in the 500-599 range
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
TotalErrorCount	Long	read-only	The number of HTTP requests that caused errors
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
TotalRequestCount	Long	read-only	The number of requests since the HTTP server was started or the statistics were reset
UnauthorizedConnectionAttempts	Long	read-only	The number of connection attempts from unauthorized hosts.

Operations

The `ConnectionManager` MBean includes a `resetStatistics` operation that resets destination statistics.

Connection MBean

The `Connection` MBean represents a remote client connection through Oracle Coherence*Extend and provides performance and usage statistics for the connection. A cluster member includes zero or more instances of this managed bean depending on the number of active remote connections to the cluster.
The object name of the MBean is:

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

Attributes

Table A-7 describes the attributes for the `Connection` MBean.

Table A-7 Connection MBean Attributes

Attribute	Type	Access	Description
ClientAddress	String	read-only	The member client's IP address.
ClientProcessName	String	read-only	The member client's process name.
ClientRole	String	read-only	The member's role.
ConnectionTimeMillis	Long	read-only	The time duration (in milliseconds) that the client has been connected.

Table A-7 (Cont.) Connection MBean Attributes

Attribute	Type	Access	Description
`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 Specifying a Cluster Member's Identity in *Developing Applications with Oracle Coherence*.
`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 or load balancer, if one is configured.
`RemotePort`	`Integer`	read-only	The port of the corresponding client or load balancer, if one is configured.
`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-8 describes the operations for the `Connection` MBean.

Table A-8 Connection MBean Operations

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

Destination MBean

The `Destination` MBean represents a federation participant and provides performance statistics. One or more managed beans is registered depending on the number of federation participants that can receive data.
The object name of the MBean is:

```
Coherence:type=Federation,service=federated,subType=Destination,name=participant
name,nodeId=cluster node id
```

Attributes

Table A-9 describes the attributes for the `Destination` MBean.

Table A-9 Destination MBean Attributes

Attribute	Type	Access	Description
`BytesSentSecs`	`Long`	read-only	The bytes sent per second.
`ConnectRetryTimeoutMillis`	`Long`	read-only	The configured connect retry timeout.
`CurrentBandwidth`	`Double`	read-only	The current utilized bandwidth in Megabits per second for sending replicate message.
`ErrorDescription`	`String`	read-only	An error description. A value is only returned if the sender is in an `ERROR` state.
`EstimatedReplicateAllRemainingTime`	`Long`	read-only	The estimated remaining time (in milliseconds) in completing the `replicateAll` request.
`GeoIp`	`String`	read-only	The location metadata that is configured for the participant.
`MaxBandwidth`	`Double`	read-only	The maximum bandwidth, per member, for sending federated data to a destination participant. This value is loaded from the source member's configuration of the destination participant. A value of `-1.0` means the maximum bandwidth is not configured.
`Member`	`String`	read-only	The member information of the destination node.
`MsgApplyTimePercentileMillis`	`Long`	read-only	The 90-percentile value of the time (in milliseconds) it took to apply the replication messages on the destination.
`MsgNetworkRoundTripTimePercentileMillis`	`Long`	read-only	The 90-percentile value of the round trip time (in milliseconds) taken for the transmission of replication messages, application of changes on the destination cluster, and the receipt of the corresponding acknowledgment messages over the network.
`MsgsSentSecs`	`Long`	read-only	The messages sent per second.
`Name`	`String`	read-only	The name of the sender.
`ParticipantType`	`String`	read-only	The participant type. Valid types are `cluster` and `interceptor`.
`RecordBacklogDelayTimePercentileMillis`	`Long`	read-only	The 90-percentile value of the time (in milliseconds) the journal records are in the cache waiting to be replicated.
`ReplicateAllPercentComplete`	`Long`	read-only	The percent of work completed for a `replicateAll` request.
`ReplicateAllTotalTime`	`Long`	read-only	The total time (in milliseconds) the `replicateAll` request took.

Table A-9 (Cont.) Destination MBean Attributes

Attribute	Type	Access	Description
`ReplicateAllLastException`	`String`	read-only	The last exception recorded for a `replicateAll` request.
`ReplicateAllPartitionCompletedCount`	`Integer`	read-only	The total number of partitions completed for the `replicateAll` request.
`ReplicateAllPartitionCount`	`Integer`	read-only	The total number of partitions in the `replicateAll` request.
`ReplicateAllPartitionErrorCount`	`Integer`	read-only	The number of partitions with error responses for the `replicateAll` request. Check `ReplicateAllLastException` for details of the last exception and the log for details of all the errors.
`SendTimeoutMillis`	`Long`	read-only	The send timeout that is configured for the participant.

Table A-9 (Cont.) Destination MBean Attributes

Attribute	Type	Access	Description
State	String	read-only	The state of the participant. One of the following: • INITIAL - The initial starting state. At this point, no changes have been made to any caches on any owned partitions for the member. • STARTING - The member is transitioning from the INITIAL state to the CONNECTING state. • CONNECTING - The member is connecting to the destination participant. • CONNECT_WAIT - The member had a failed connect attempt. It will make another connect attempt in the future. • IDLE - The member is connected to the destination participant. There are no pending changes to send or pending acknowledgments. • READY - The member is connected to the destination participant. The member is transitioning to the SENDING state. • SENDING - The member is connected to the destination participant. The member is preparing changes to send to the destination participant. • STOPPED - The member is not connected. Sending changes is disabled. • PAUSED - The member is not connected. Pending changes are accumulated. • ERROR - An error occurred resulting in stopping federation. The member is not connected and sending changes is disabled. • YIELDING - The member is connected. The member is temporarily paused waiting for acknowledgements or for a network backlog to clear. • DISCONNECTED - The member has lost its connection to the destination participant. The member is transitioning to the CONNECTING state to establish a new connection. • DISCONECTING - The member's connection to the destination participant is closing. After the connection is closed, the member will transition to the DISCONNECTED state.

Table A-9 (Cont.) Destination MBean Attributes

Attribute	Type	Access	Description
StateCode	Integer	read-only	The state code of the participant. One of the following: • 0: INITIAL • 1: STARTING • 2: CONNECTING • 3: CONNECT_WAIT • 4: IDLE • 5: READY • 6: SENDING • 7: STOPPED • 8: PAUSED • 9: ERROR • 10: YIELDING • 11: DISCONNECTED • 12: DISCONNECTING For a description of each of these state codes, see State.
Status	Integer	read-only	The status of the participant. Statuses are: • 0 – OK • 1 – Warning • 2 – Error
TotalBytesSent	Long	read-only	The total number of bytes that were sent
TotalEntriesSent	Long	read-only	The total number of cache entries that were sent
TotalErrorResponses	Long	read-only	The total number of responses with error. Check the log for details of the errors.
TotalMsgSent	Long	read-only	The total number of replication messages that were sent. A replication message can contain multiple journal records.
TotalMsgUnacked	Long	read-only	The total number of un-acknowledged replication messages
TotalRecordsSent	Long	read-only	The total number of journal records that were sent. A journal record can consist of multiple cache entries that are part of the same transaction
TraceLogging	Boolean	read/write	Used to enable/disable federation trace logging. Valid values are true to enable federation trace logging and false to disable federation trace logging.

Operations

The Destination MBean includes a resetStatistics operation that resets destination statistics.

Executor MBean

The `Executor` MBean provides statistics for the executor services that run in a cluster. A cluster contains zero or more instances of this MBean, each instance representing an instance of an executor service on a member.
The object name of the MBean is:

```
Type=Executor,name=executor name,nodeId=node
```

Attributes

Table A-10 describes the attributes for the `Executor` MBean.

Table A-10 Executor MBean Attributes

Attribute	Type	Access	Description
`MemberId`	`Integer`	read-only	The member id where the executor is running.
`Name`	`String`	read-only	The logical name of this executor.
`Id`	`String`	read-only	The unique ID of this executor.
`Description`	`String`	read-only	The description of this executor.
`Location`	`String`	read-only	The location where the executor is running.
`State`	`String`	read-only	The state of the executor (see description below).
`StateCode`	`Integer`	read-only	The State of the executor: • 1 = (JOINING) indicates the executor is joining the orchestration. • 2 = (RUNNING) indicates the executor is accepting and executing tasks. • 3 = (CLOSING_GRACEFULLY) indicates the executor has commenced graceful closing. No new tasks will be accepted, but existing ones will run to completion. • 4 = (CLOSING) indicates the executor has commenced closing. • 5 = (CLOSED) indicates the executor is closed. • 6 = (REJECTING) indicates the executor is rejecting tasks.
`TasksCompletedCount`	`Integer`	read-only	The count of the tasks completed.
`TasksRejectedCount`	`Integer`	read-only	The count of the tasks rejected.
`TaksInProgressCount`	`Integer`	read-only	The count of tasks in progress.
`TraceLogging`	`Boolean`	read/write	Indicates the executor `traceLogging` is enabled (`true`) or not (`false`).

Operations

The `Executor` MBean includes a `resetStatistics` operation that resets all executor statistics.

FederationManager MBean

The `FederationManager` MBean represents a `FederatedCache` service instance and provides operations that control replication of data to service participants. One or more managed beans is registered depending on the number of service instances.
The object name of the MBean is:

```
Coherence:type=Federation,service=service name,responsibility=Coordinator
```

Attributes

Table A-11 describes the attributes for the `FederationManager` MBean.

Table A-11 FederationManager MBean Attributes

Attribute	Type	Access	Description
`CoordinatorId`	`Integer`	read-only	The member identifier for the federation coordinator node
`LocalParticipantName`	`String`	read-only	The name of the local participant for this cluster
`TraceLogging`	`Boolean`	read/write	Indicates whether the federation trace logging is enabled or not.

Operations

Table A-12 describes the operations for `FederationManager` MBean.

Table A-12 FederationManager MBean Operations

Operation	Parameters	Return Type	Description
`start`	`String p1`	`void`	Start federating data to the entered participant. A value of `null` starts federating to all participants. The connection is established lazily only when there is data to be replicated.
`startWithNoBacklog`	`String p1`	`void`	Start federating to the entered participant and clear any initial backlog. A value of `null` starts federating to all participants. The connection is established lazily only when there is data to be federated.
`startWithSync`	`String p1`	`void`	Start federating to the entered participant and perform a `ReplicateAll` operation to synchronize the participant. A value of `null` starts federating to all participants. The connection is established lazily only when there is data to be federated.
`stop`	`String p1`	`void`	Stop federating data to the entered participant. A value of `null` stops federating to all participants.
`pause`	`String p1`	`void`	Pause federating data to the entered participant. A value of `null` pauses federating to all participants.

Table A-12 (Cont.) FederationManager MBean Operations

Operation	Parameters	Return Type	Description
`reportState`	`String p1`	`Map`	Return a `Map` that indicates the state of the entered participant. The key in the map represents the state and the value represents the percent of service members that are in that state.
`replicateAll`	`String p1`	`void`	Asynchronously replicate all caches to the entered participant
`retrievePendingOutgoingMessages`	Not applicable	`Integer`	Number of pending outgoing federation messages
`retrievePendingIncomingMessages`	Not applicable	`Integer`	Number of incoming federation messages currently being processed
`listNonFederatedCaches`	`String p1`	`String[]`	A list of the names of the caches that belong to a Federated Cache Service but are not being federated.

FlashJournalRM MBean

The `Journal` MBean provides an interface for the flash journal resource manager (`FlashJournalRM`) and provides operational statistics. The resource manager stores data to flash memory. Each cluster member includes a single instance of this managed bean. The object name of the MBean is:

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

Attributes

Table A-13 describes the attributes for the `Journal` MBean that pertain to the flash journal resource manager.

Table A-13 Journal MBean 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.

Table A-13 (Cont.) Journal MBean Attributes for FlashJournalRM

Attribute	Type	Access	Description
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
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 `Journal` MBean has no operations.

Health MBean

The `Health` MBean provides information about health checks configured in a cluster. A cluster contains zero or more instances of this MBean, each instance representing an instance of a health check on a member.
The object name of the MBean is:

```
Type=Health,subType=subType,name=HealthName,nodeId=node
```

Attributes

Table A-14 describes the attributes for `Health` MBean.

Table A-14 Health MBean Attributes

Attribute	Type	Access	Description
`ClassName`	`String`	read-only	The name of the class implementing this health check.
`Description`	`String`	read-only	The description of this health check.
`SubType`	`String`	read-only	The sub-type of this health check.
`MemberHealthCheck`	`Boolean`	read-only	Indicates if this health check should be included when working out this Coherence member's health status.
`Started`	`Boolean`	read-only	Indicates if the resource represented by this health check is started.
`Live`	`Boolean`	read-only	Indicates if the resource represented by this health check is alive.
`Ready`	`Boolean`	read-only	Indicates if the resource represented by this health check is ready.
`Safe`	`Boolean`	read-only	Indicates if the resource represented by this health check is safe.

Management MBean

The `Management` MBean represents the grid JMX infrastructure and provides operational settings. Some of the attributes are writable and allow management behavior to be changed in real time. Each cluster member includes a single instance of this managed bean.
The object name of the MBean is:

```
type=Management
```

Attributes

Table A-15 describes the attributes for `Management` MBean.

Table A-15 Management MBean 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 `Management` MBean includes a `resetStatistics` operation that resets the `RefreshCount`, `RefreshExcessCount`, and `RefreshPredictionCount` statistics.

Origin MBean

The `Origin` MBean represents a federation participant and provides performance statistics. One or more managed beans is registered depending on the number of federation participants that can send data.
The object name of the MBean is:

```
Coherence:type=Federation,service=federated,subType=Origin,name=participant
name,nodeId=cluster node id
```

Attributes

Table A-16 describes the attributes for `Origin` MBean.

Table A-16 Origin MBean Attributes

Attribute	Type	Access	Description
`BytesReceivedSecs`	`Long`	read-only	The bytes received per second
`CurrentConnectionCount`	`Integer`	read-only	The count of the incoming participant members currently connected to the member.
`MapMembers`	`Map<String, String>` **Note**: The key is endpoint and the value is member information.	read-only	A map of the incoming participant members, specified by the endpoint, currently connected to the member.
`Member` **Note**: This attribute is replaced by `MapMembers`. Therefore, it is deprecated in WebLogic Server 14.1.1.2206 and will be removed in the next release.	`String`	read-only	The member information of the in-coming participant node.
`MsgApplyTimePercentileMillis`	`Long`	read-only	The 90-percentile value of the time (in milliseconds) it took to apply the replication messages on the destination.
`MsgsReceivedSecs`	`Long`	read-only	The messages received per second.
`RecordBacklogDelayTimePercentileMillis`	`Long`	read-only	The 90-percentile value of the time (in milliseconds) the journal records are in the cache waiting to be replicated.
`ReplicateMessageLastException`	`String`	read-only	The last exception recorded for a `ReplicateMessage` request.
`TotalBytesReceived`	`Long`	read-only	The total number of bytes that were received.
`TotalEntriesReceived`	`Long`	read-only	The total number of cache entries that were received.
`TotalErrorRequests`	`Integer`	read-only	The total number of `ReplicateMessage` requests with error. Check the log for details of the errors.
`TotalMsgReceived`	`Long`	read-only	The total number of replication messages that were received. A replication message could contain multiple journal records.
`TotalMsgUnacked`	`Long`	read-only	The total number of un-acknowledged replication messages.
`TotalRecordsReceived`	`Long`	read-only	The total number of journal records that were received. A journal record could consist of multiple cache entries that are part of the same transaction.

Operations

The `Origin` MBean includes a `resetStatistics` operation that resets destination statistics.

PagedTopic MBean

The `PagedTopic` MBean provides statistics for Topic services running in a cluster. A cluster contains zero or more instances of this MBean, each instance representing an instance of a Topic on a member.

> **Note:**
>
> The `PagedTopic` MBean is available only after you have installed the Cumulative Patch Update (CPU) 35122413 or later.

The object name of the MBean is:

```
type=PagedTopic,service=ServiceName,name=TopicName,nodeId=node
```

Attributes

Table A-17 describes the attributes for the `PagedTopic` MBean.

Table A-17 PagedTopic MBean Attributes

Attribute	Type	Access	Description
`AllowUnownedCommits`	`Boolean`	read-only	Indicates if the system allows unowned commits.
`ChannelCount`	`Integer`	read-only	The number of channels in the Topic.
`Channels`	`TabularData`	read-only	The channel statistics.
`ElementCalculator`	`String`	read-only	The element calculator.
`PageCapacity`	`Integer`	read-only	The capacity of a page.
`PublishedCount`	`Long`	read-only	The number of published messages.
`PublishedFifteenMinuteRate`	`Double`	read-only	The rate of publishing messages over the last 15 minutes.
`PublishedFiveMinuteRate`	`Double`	read-only	The rate of publishing messages over the last five minutes.
`PublishedOneMinuteRate`	`Double`	read-only	The rate of publishing messages over the last one minute.
`PublishedMeanRate`	`Double`	read-only	The mean rate at which messages are published.
`ReconnectRetry`	`Integer`	read-only	The number of reconnect retries that are allowed for the Topic.

Table A-17 (Cont.) PagedTopic MBean Attributes

Attribute	Type	Access	Description
ReconnectTimeout	Integer	read-only	The maximum time period to attempt a reconnect before timeout occurs.
ReconnectWait	Integer	read-only	The waiting time before attempting a reconnect.
RetainConsumed	Boolean	read-only	Indicates if the system should retain consumed values.
SubscriberTimeout	Integer	read-only	The timeout value for the subscriber.

Operations

The `PagedTopic` MBean includes the `disconnectAll` operation that forces the Topic to disconnect all its subscribers.

PagedTopicSubscriber MBean

The `PagedTopicSubscriber` MBean provides statistics for Topic Subscribers running in a cluster. A cluster contains zero or more instances of this MBean, each instance representing an instance of a Topic Subscriber on a member.

> **Note:**
>
> The `PagedTopicSubscriber` MBean is available only after you have installed the Cumulative Patch Update (CPU) 35122413 or later.

The object name of the MBean is:

```
type=PagedTopicSubscriber,service=ServiceName,topic=TopicName,subtype=SubType,
group=SubscriberGroup,id=SubscriberId,nodeId=node
```

Attributes

Table A-18 describes the attributes for the `PagedTopicSubscriber` MBean.

Table A-18 PagedTopicSubscriber MBean Attributes

Attribute	Type	Access	Description
Backlog	Integer	read-only	The number of outstanding receive requests.
ChannelAllocations	String	read-only	The subscriber's allocated channels.

Table A-18 (Cont.) PagedTopicSubscriber MBean Attributes

Attribute	Type	Access	Description
ChannelCount	Integer	read-only	The number of channels in the Topic.
Channels	TabularData	read-only	The channel statistics.
CompleteOnEmpty	Boolean	read-only	An option indicating whether the subscriber completes the receive requests with a null message when the Topic is empty.
Converter	String	read-only	The optional converter being used to transform messages.
Disconnections	Long	read-only	The number of times this subscriber has disconnected.
Filter	String	read-only	The optional filter being used to filter messages.
Id	Long	read-only	The subscriber's identifier.
IdentifyingName	String	read-only	An optional name to help identify this subscriber.
MaxBacklog	Long	read-only	The maximum number of outstanding receive requests allowed before flow control blocks receive calls.
Member	String	read-only	The cluster member owning this subscriber.
NotificationId	Long	read-only	The subscriber's notification identifier.
Notifications	Long	read-only	The number of channel notifications received.
Polls	Long	read-only	The total number of polls for messages.
ReceiveCompletionsCount	Long	read-only	The number of completed receive requests.
ReceiveCompletionsFifteenMinuteRate	Double	read-only	The rate at which receive requests are completed over the last 15 minutes.
ReceiveCompletionsFiveMinuteRate	Double	read-only	The rate at which receive requests are completed over the last five minutes.

Table A-18 (Cont.) PagedTopicSubscriber MBean Attributes

Attribute	Type	Access	Description
ReceiveCompletionsOneMinuteRate	Double	read-only	The rate at which receive requests are completed over the last one minute.
ReceiveCompletionsMeanRate	Double	read-only	The mean rate at which receive requests are completed.
ReceiveEmpty	Long	read-only	The number of empty receive requests.
ReceiveErrors	Long	read-only	The number of exceptionally completed receive requests.
ReceivedCount	Long	read-only	The number of elements received.
Serializer	String	read-only	The serializer used to deserialize messages.
State	Integer	read-only	The state of the subscriber. Valid values are: • 0 - Initial • 1 - Connected • 2 - Disconnected • 3 - Closing • 4 - Closed
StateName	String	read-only	The state of the subscriber as a string.
SubTypeCode	Integer	read-only	Indicates if the subscriber is Durable (1) or Anonymous (0).
SubscriberGroup	String	read-only	The subscriber group to which the subscriber belongs.
Type	String	read-only	The type of this subscriber.
Waits	Long	read-only	The number of waits on an empty channel.

Operations

Table A-19 describes the operations for `PagedTopicSubscriber` MBean.

Table A-19 PagedTopicSubscriber MBean Operations

Operations	Parameters	Return Type	Description
connect	Not applicable	void	Ensure that this subscriber is connected.

Table A-19 (Cont.) PagedTopicSubscriber MBean Operations

Operations	Parameters	Return Type	Description
`disconnect`	Not applicable	`void`	Force this subscriber to disconnect and reset itself.
`heads`	Not applicable	`TabularData`	Retrieve the current head positions for each channel.
`notifyPopulated`	`Integer nChannel`	`void`	Send a channel populated notification to this subscriber.
`remainingMessages`	Not applicable	`TabularData`	Retrieve the count of remaining messages for each channel.

PagedTopicSubscriberGroup MBean

The `PagedTopicSubscriberGroup` MBean provides statistics for Topic Subscriber Groups running in a cluster. A cluster contains zero or more instances of this MBean, each instance representing an instance of a Topic Subscriber Group on a member.

> **Note:**
>
> The `PagedTopicSubscriberGroup` MBean is available only after you have installed the Cumulative Patch Update (CPU) 35122413 or later.

The object name of the MBean is:

```
type=PagedTopicSubscriberGroup,service=ServiceName,topic=TopicName,subtype=SubType,name=SubscriberGroup,nodeId=node
```

Attributes

Table A-20 describes the attributes for the `PagedTopicSubscriberGroup` MBean.

Table A-20 PagedTopicSubscriberGroup MBean Attributes

Attribute	Type	Access	Description
`ChannelCount`	`Integer`	read-only	The number of channels in the Topic.
`Channels`	`TabularData`	read-only	The channel statistics.
`Filter`	`String`	read-only	The filter.
`PolledCount`	`Long`	read-only	The number of polled messages.
`PolledFifteenMinuteRate`	`Double`	read-only	The rate at which messages are polled over the last 15 minutes.

Table A-20 (Cont.) PagedTopicSubscriberGroup MBean Attributes

Attribute	Type	Access	Description
PolledFiveMinuteRate	Double	read-only	The rate at which messages are polled over the last five minutes.
PolledOneMinuteRate	Double	read-only	The rate at which messages are polled over the last one minute.
PolledMeanRate	Double	read-only	The mean rate at which messages are polled.
Transformer	String	read-only	The transformer.

Operations

The `PagedTopicSubscriberGroup` MBean includes the `disconnectAll` operation that forces the subscriber group to disconnect all its subscribers.

PersistenceManager MBean

The `PersistenceManager` MBean provides attributes and operations related to persisting and recovering the contents of distributed cache services. Each service instance includes a single instance of this managed bean.
The object name of the MBean is:

```
type=Persistence,service=service name,responsibility=PersistenceCoordinator
```

Attributes

Table A-21 describes the attributes for `PersistenceManager` MBean.

Table A-21 PersistenceManager MBean Attributes

Attribute	Type	Access	Description
OperationsStatus	String	read-only	The status of the current operation issued by the persistence coordinator
ManagerId	Integer	read-only	The member id of the service node that is the persistence coordinator
Snapshots	String[]	read-only	The list of snapshot identifiers that are available to recover from
Idle	Boolean	read-only	Whether or not the persistence coordinator is idle

Operations

Table A-22 describes the operations for `PersistenceManager` MBean.

Table A-22 PersistenceManager MBean Operations

Operation	Parameters	Return Type	Description
archiveSnapshot	String sName	void	Archive the snapshot to a centralized location. This operation is asynchronous.
createSnapshot	String sName	void	Create a snapshot of the service with the specified name. This operation is asynchronous.
forceRecovery	Not applicable	void	Proceed with recovery despite the dynamic quorum policy objections. This may lead to the partial or full data loss at the corresponding cache service.
listArchivedSnapshots	Not applicable	String[]	Return a list of archived snapshots for the service
listArchivedSnapshotsStores	String sName	String[]	Return a list of stores for the archived snapshot with the specified name
removeArchivedSnapshot	String sName	void	Remove the archived snapshot from a centralized location
recoverSnapshot	String sName	void	Recover from a snapshot of the service with the specified name. This operation is asynchronous.
retrieveArchivedSnapshot	String sName	void	Retrieve the archived snapshot from a centralized location. This operation is asynchronous.
removeSnapshot	String sName	void	Remove the snapshot of the service with the specified name. This operation is asynchronous.

Notifications

Table A-23 describes the notifications for PersistenceManager MBean.

Table A-23 PersistenceManager MBean Notifications

Notification	Description
ARCHIVE_SNAPSHOT_BEGIN	Indicates the start of archiving a snapshot
ARCHIVE_SNAPSHOT_END	Indicates the end of archiving a snapshot
CREATE_SNAPSHOT_BEGIN	Indicates the start of snapshot creation
CREATE_SNAPSHOT_END	Indicates the end of snapshot creation
RECOVER_BEGIN	Indicates the start of recovery
RECOVER_DISALOWED	Indicates recovery being disallowed by the quorum
RECOVER_END	Indicates the end of recovery
RECOVER_SNAPSHOT_BEGIN	Indicates the start of recovering to a snapshot
RECOVER_SNAPSHOT_END	Indicates the end of recovering to a snapshot
REMOVE_ARCHIVED_SNAPSHOT_BEGIN	Indicates the start of removing an archived snapshot
REMOVE_ARCHIVED_SNAPSHOT_END	Indicates the end of removing an archived snapshot
REMOVE_SNAPSHOT_BEGIN	Indicates the start of removing a snapshot

Table A-23 (Cont.) PersistenceManager MBean Notifications

Notification	Description
REMOVE_SNAPSHOT_END	Indicates the end of removing a snapshot
RETRIEVE_ARCHIVED_SNAPSHOT_BEGIN	Indicates the start of retrieving an archived snapshot
RETRIEVE_ARCHIVED_SNAPSHOT_END	Indicates the end of retrieving an archiving snapshot

PointToPoint MBean

The `PointToPoint` MBean represents the network status between two cluster members and 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. Each cluster member includes a single instance of this managed bean.
The object name of the MBean is:

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

Attributes

Table A-24 describes the attributes for `PointToPoint` MBean.

Table A-24 PointToPoint MBean 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 flow-control 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 flow-control 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

Table A-24 (Cont.) PointToPoint MBean Attributes

Attribute	Type	Access	Description
`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 flow-control 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 flow-control in *Developing Applications with Oracle Coherence*.
`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 flow-control 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 flow-control in *Developing Applications with Oracle Coherence*.
`ViewedMemberId`	`Integer`	read/write	The ID of the member being viewed

Table A-24 (Cont.) PointToPoint MBean Attributes

Attribute	Type	Access	Description
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-25 describes the operations for PointToPoint MBean.

Table A-25 PointToPoint MBean 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 Journal MBean provides an interface for the RAM journal resource manager (RamJournalRM) and provides operational statistics. The resource manager is responsible for storing data to RAM memory. Each cluster member includes a single instance of this managed bean.
The object name of the MBean is:

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

Attributes

Table A-26 describes the attributes for JournalMBean that pertain to the RAM journal resource manager.

Table A-26 Journal MBean 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.

Table A-26 (Cont.) Journal MBean Attributes for RamJournalRM

Attribute	Type	Access	Description
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.
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 Journal MBean has no operations.

Reporter MBean

The Reporter MBean represents the Oracle Coherence Reporter and provides 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. Each cluster member includes a single instance of this managed bean.
The object name of the MBean is:

```
type=Reporter
```

Attributes

Table A-27 describes the attributes for `Reporter` MBean.

Table A-27 Reporter MBean 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-28 describes the operations for `Reporter` MBean.

Table A-28 Reporter MBean 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`).

Table A-28 (Cont.) Reporter MBean Operations

Operation	Parameters	Return Type	Description
`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

Service MBean

The `Service` MBean represents a clustered service and provides usage and performance statistics. 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. A cluster member includes zero or more instances of this managed bean depending on the number of clustered services that are started.
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 `Service` MBean.

- 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-29 describes the attributes for `Service` MBean.

Table A-29 Service MBean 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
JoinTime	Date	read-only	The date and time (in cluster time) that this member joined the service
MemberCount	Integer	read-only	The total number of cluster nodes running this service
MessagesLocal	Long	read-only	The total number of messages which were self-addressed messages since the last time the statistics were reset. Such messages are used for servicing process-local requests and do not have an associated network cost
MessagesReceived	Long	read-only	The total number of messages received by this service since the last time the statistics were reset. This value accounts for messages received by any (local, dedicated or shared) transport
MessagesSent	Long	read-only	The number of messages sent by this service since the last time the statistics were reset. This value accounts for messages sent by any (local, dedicated or shared) transport
OutgoingTransferCount	Integer	read-only	The number of partitions that are currently being transferred by this service member to other members
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

Table A-29 (Cont.) Service MBean Attributes

Attribute	Type	Access	Description
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 machine where the primary partition owner resides
PersistenceActiveSpaceAvailable	Long	read-only	The remaining space (in bytes) available on the file system for active persistence
PersistenceActiveSpaceTotal	Long	read-only	The total size (in bytes) of the file system for use by active persistence
PersistenceActiveSpaceUsed	Long	read-only	The amount of space (in bytes) that is used by active persistence
PersistenceBackupSpaceUsed	Long	read-only	The total size (in bytes) used by the persistence layer to persist the backup cache data.
PersistenceBackupSpaceTotal	Long	read-only	The total size (in bytes) of the file system used by the persistence layer to persist the backup cache data.
PersistenceBackupSpaceAvailable	Long	read-only	The total remaining free space (in bytes) of the file system used by the persistence layer to persist backup cache data.
PersistenceEnvironment	String	read-only	A description of the configured persistence environment or n/a if one has not been configured.
PersistenceLatencyAverage	Float	read-only	The average latency (in milliseconds) added to a mutating cache operation by active persistence operations
PersistenceLatencyMax	Long	read-only	The maximum latency (in milliseconds) added to a mutating cache operation by an active persistence operation
PersistenceMode	String	read-only	The current persistence mode for this service: • active – all mutating cache operations are persisted using the configured persistence environment. • on-demand – a persistence environment has been configured and is available but is not being actively used. • n/a – persistence is not configured for this service.
PersistenceSnapshotArchiver	String	read-only	A description of the configured snapshot archiver or n/a if one has not been configured.
PersistenceSnapshotSpaceAvailable	Long	read-only	The remaining space (in bytes) available on the file system to store snapshots
PersistenceSnapshotSpaceTotal	Long	read-only	The total size (in bytes) of the file system to store snapshots
QuorumStatus	String	read-only	The current state of the service quorum

Table A-29 (Cont.) Service MBean Attributes

Attribute	Type	Access	Description
RefreshTime	Date	read-only	The time stamp 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
StatusHA	String	read-only	The High Availability (HA) status for this service. • MACHINE-SAFE: The loss of a machine will not result in data loss. • RACK-SAFE: The loss of a rack will not result in data loss. • SITE-SAFE: The loss of a site will not result in data loss. • NODE-SAFE: A cluster node can be stopped without any data loss. • ENDANGERED: Abnormal termination of a cluster node that runs this service can cause data loss. **Note**: To achieve RACK-SAFE and SITE-SAFE, each cluster node must have its rack and site name configured.
StorageEnabled	Boolean	read-only	Specifies whether the local storage is enabled for this cluster member

Table A-29 (Cont.) Service MBean Attributes

Attribute	Type	Access	Description
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
ThreadCount	Integer	read/write	The number of threads in the service thread pool. To configure a thread count, set the thread-count-min and thread-count-max elements to the same value.

Table A-29 (Cont.) Service MBean Attributes

Attribute	Type	Access	Description
ThreadCountMax	Integer	read/write	The maximum number of daemon threads. Usage of daemon threads varies for different service types. If zero or negative, the service does not use daemon threads and all relevant tasks are performed on the service thread. Furthermore, if negative, tasks are performed on the caller's thread where possible.
ThreadCountMin	Integer	read/write	The minimum number of daemon threads. Usage of daemon threads varies for different service types. If zero or negative, the service does not use daemon threads and all relevant tasks are performed on the service thread. Furthermore, if negative, tasks are performed on the caller's thread where possible.
ThreadCountUpdateTime	Date	read-only	The last time an update was made to the thread count. This attribute is only valid when the ThreadPoolSizingEnabled attribute is 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. To enable dynamic thread pools, the thread-count-min and thread-count-max elements must be set on the service.
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-29 (Cont.) Service MBean 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-30 describes the operations for `Service` MBean.

Table A-30 Service MBean 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.

SimpleStrategy MBean

The `SimpleStrategy` MBean represents a simple partition assignment strategy and provides operational statistics. Each partitioned service registers a single instance of this managed bean. The MBean is attached to a single instance of the `PartitionAssignmentStrategy` object which exists on the member that is the distribution coordinator for the service. The associated MBean is not explicitly unregistered, but its name is rebound to a new MBean instance if and when a different service member becomes the distribution coordinator.
The object name of the MBean is:

```
Coherence:type=PartitionAssignment,service=service name,
responsibility=DistributionCoordinator
```

Attributes

Table A-31 describes the attributes for `SimpleStrategy` MBean.

Table A-31 SimpleStrategy MBean Attributes

Attribute	Type	Access	Description
`AveragePartitionSizeKB`	`Long`	read-only	The average partition storage size in kilobytes
`AverageStorageSizeKB`	`Long`	read-only	The average node storage size in kilobytes
`BackupCount`	`Integer`	read-only	The configured number of partition backups to be maintained by the service
`CoordinatorId`	`Integer`	read-only	The member identifier of the service node that is the ownership distribution coordinator
`FairShareBackup`	`Integer`	read-only	The number of backup partitions per storage-enabled service member that this strategy currently attempts to maintain
`FairSharePrimary`	`Integer`	read-only	The number of primary partitions per storage-enabled service member that this strategy currently attempts to maintain
`HAStatus`	`String`	read-only	The high availability status for this service. Valid values are: • `MACHINE-SAFE`: The loss of a machine will not result in data loss. • `RACK-SAFE`: The loss of a rack will not result in data loss. • `SITE-SAFE`: The loss of a site will not result in data loss. • `NODE-SAFE`: A cluster node can be stopped without any data loss. • `ENDANGERED`: Abnormal termination of a cluster node that runs this service can cause data loss. **Note**: To achieve `RACK-SAFE` and `SITE-SAFE`, each cluster node must have its rack and site name configured.
`HATarget`	`String`	read-only	The high availability status that this strategy attempts to achieve. Valid values are the same as the `HAStatus` attribute.
`LastAnalysisTime`	`Date`	read-only	The last time a distribution analysis was performed
`MaxLoadNodeId`	`Integer`	read-only	The node identified with the maximum node storage size
`MaxPartitionSizeKB`	`Long`	read-only	The maximum partition storage size in kilobytes
`MaxStorageSizeKB`	`Long`	read-only	The maximum node storage size in kilobytes
`PartitionCount`	`Integer`	read-only	The configured number of partitions for the service
`RemainingDistributionCount`	`Integer`	read-only	The number of partition transfers that remain to be completed before the service achieves the goals set by this strategy
`ServiceMachineCount`	`Integer`	read-only	The number of machines that host storage-enabled nodes running this service

Table A-31 (Cont.) SimpleStrategy MBean Attributes

Attribute	Type	Access	Description
ServiceNodeCount	Integer	read-only	The number of storage-enabled nodes running this service
ServiceRackCount	Integer	read-only	The number of racks that host storage-enabled nodes running this service
ServiceSiteCount	Integer	read-only	The number of sites that host storage-enabled nodes running this service
StrategyName	String	read-only	The name of the partition assignment strategy that is in use

Operations

Table A-32 describes the operations for SimpleStrategy MBean.

Table A-32 SimpleStrategy MBean Operations

Operation	Parameters	Return Type	Description
reportScheduledDistributions	Boolean fVerbose	String	Report partitions that remain to be transferred to achieve the goals set by this strategy. If the fVerbose parameter is set to true, the report includes details for each scheduled transfer.

StorageManager MBean

The StorageManager MBean represents a storage instance for a storage-enabled distributed cache service and provides usage statistics for the storage-enabled cache and also includes statistics for queries. 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 object name of the MBean is:

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

Attributes

Table A-33 describes the attributes for StorageManager MBean.

Table A-33 StorageManager MBean Attributes

Attribute	Type	Access	Description
ClearCount	Long	read-only	The number of clear() operations since the last time the statistics were reset.

Table A-33 (Cont.) StorageManager MBean 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).
`IndexingTotalMillis`	`Long`	read-only	The cumulative duration, in milliseconds, of index builds since the last report refresh.
`IndexTotalUnits`	`Long`	read-only	The total units used by all indices on the associated cache.

Table A-33 (Cont.) StorageManager MBean 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. A call to NamedCache.lock() increments the value, while a call to NamedCache.unlock() decrements the value.
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

Table A-33 (Cont.) StorageManager MBean Attributes

Attribute	Type	Access	Description
`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
`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

Table A-34 describes the operations for `StorageManager` MBean.

> **Note:**
>
> The `clearCache` and `truncateCache` operations will be available only after you have installed the Cumulative Patch Update (CPU) 14.1.1.2206.5 or later.

Table A-34 StorageManager MBean Operations

Operation	Parameters	Return Type	Description
`clearCache`	Not applicable	`Void`	Remove all items from this cache. Invoking the `clear()` operation against a distributed cache can be both a memory and CPU intensive task. Therefore, it is generally not recommended.
`reportPartitionStats`	`String sFormat`	`String`	Return the partition statistics across all partitions and members for the current cache. The format can be `json` or `csv` and the data returned includes: • `partitionId` - the partition ID. • `count` - the count of entries in this partition for this cache. • `totalSize` - the total size of keys and values in this partition for the cache. • `MaxEntrySize` - the maximum entry size (key + value). • `MemberId` - the member ID owning the partition.
`resetStatistics`	Not applicable	`Void`	Reset the storage manager statistics. This operation does not reset the `EvictionCount`, `InsertCount`, or `RemoveCount` attributes.
`truncateCache`	Not applicable	`Void`	Remove all items from this cache. The removal of entries caused by this truncate operation will not be observable. This includes any registered listeners, triggers, or interceptors. However, a `CacheLifecycleEvent` event is raised to notify subscribers of the execution of this operation.

Topology MBean

The `Topology` MBean represents federation participants in the context of a topology. A federation topology defines how data is synchronized among federation participants. An instance of this managed bean is registered for each topology being used by a federated cache service.
The object name of the MBean is:

```
Coherence:type=Federation,subType=Topology,name=topology name nodeId=cluster node id
```

Attributes

Table A-35 describes the attributes for `Topology` MBean.

Table A-35 Topology MBean Attributes

Attribute	Type	Access	Description
`CacheNames`	`Collection`	read-only	The names of the caches using this topology
`ParticpantTypes`	`Map`	read-only	The map of participants and their corresponding type
`RoleParticipants`	`Map`	read-only	The map of roles and the participants in that role
`TopologyType`	`String`	read-only	The topology type

Operations

The `Topology` MBean has no operations.

TransactionManager MBean

The `TransactionManager` MBean represents a transaction manager in the transactional framework and provides global transaction manager statics by aggregating service-level statistics from all transaction service instances. A cluster member includes zero or more instances of this managed bean depending on the number of configured transactional caches. 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-36 describes the attributes for `TransactionManager` MBean.

Table A-36 TransactionManager MBean 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.
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 `TransactionManager` MBean includes a `resetStatistics` operation that resets all transaction manager statistics.

View MBean

The `View` MBean provides statistics for view caches that run in a cluster.A cluster contains zero or more instances of this MBean, each instance representing an instance of a view cache. The object name of the MBean is:

```
Type=View,service=service,name=view name,nodeId=node
```

Attributes

Table A-37 describes the attributes for `View` MBean.

Table A-37 View MBean Attributes

Attribute	Type	Access	Description
`ViewName`	`String`	read-only	The name of the view cache.
`ReadOnly`	`Boolean`	read-only	Indicates if the view cache is read-only.
`Transformed`	`Boolean`	read-only	Indicates if the cache transforms values.
`Transformer`	`String`	read-only	The implementation of a `com.tangosol.util.ValueExtractor` extractor used to transform values retrieved from the underlying cache, before storing them locally. If specified, this view can be set to read-only.
`Filter`	`String`	read-only	The implementation of a `com.tangosol.util.Filter` filter used by the associated view-scheme.
`ReconnectInterval`	`Long`	read-only	Indicates the period (milliseconds) in which re-synchronization with the underlying cache will be delayed in the case the connection is severed.
`CacheValues`	`Boolean`	read-only	Determines whether cache should cache values or only keys.
`Size`	`Long`	read-only	The number of entries in the view-cache.

Operations

The `View` MBean includes a `resetStatistics` operation that resets all view statistics.

B

Report File Configuration Elements

The report file configuration reference provides a detailed description of the report file deployment descriptor elements.

This appendix includes the following sections:

- Report File Deployment Descriptor
- Report File 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. See Analyzing Report Contents. Modify the reports or create new reports as required.
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/1.1/coherence-report-config.xsd

The `<report-config>` element is the root element of the deployment descriptor and includes the XSD and namespace declarations. 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">
```

Note:

- 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.

Report File Element Reference

The report file element reference includes all non-terminal report file configuration elements. Each section includes instructions on how to use the element and also includes descriptions for all valid subelements.

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-1 describes the subelements of the `column` element.

Table B-1 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-1 (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-2 describes the subelements of the `filter` element.

Table B-2 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-3 describes the subelements of the `filters` element.

Table B-3 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-4 describes the subelements of the `params` element.

Table B-4 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-5 describes the subelements of the `query` element.

Table B-5 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-6 describes the subelements of the `report` element.

Table B-6 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-7 describes the subelements of the `report-config` element.

Table B-7 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-8 describes the subelements of the `row` element.

Table B-8 row Subelements

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

C

Report Group Configuration Elements

The report group configuration reference provides a detailed description of the report group deployment descriptor elements.

This appendix includes the following sections:

- Report Group Configuration Deployment Descriptor
- Report Group 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 Switching the Default Report Group Configuration File.
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/1.1/coherence-report-group-config.xsd

The `<report-group>` element is the root element of the descriptor and includes the XSD and namespace declarations. 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">
```

> **Note:**
>
> - 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.

Report Group Element Reference

The report group element reference includes all non-terminal report group configuration elements. Each section includes instructions on how to use the element and also includes descriptions for all valid subelements.

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-1 describes the subelements of the `init-param` element.

Table C-1 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-2 describes the subelements of the `init-params` element.

Table C-2 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-3 describes the subelements of the `report-config` element.

Table C-3 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-4 describes the subelements of the `report-group` element.

Table C-4 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 `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-5 describes the subelements of the `report-list` element.

Table C-5 report-list Subelements

Element	Required/ Optional	Description
report-config	Required	Specifies the location of the report configuration file and corresponding initialization parameters