

# **Oracle® Process Manager and Notification Server**

Administrator's Guide

10g Release 2 (10.1.2)

**B13996-02**

July 2005

Primary Author: Kurt Heiss

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software—Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

---

---

# Contents

<b>Preface</b> .....	ix
Audience .....	ix
Documentation Accessibility .....	ix
Related Documentation .....	x
Conventions .....	x
 <b>1 OPMN: Overview</b>	
1.1 What is OPMN? .....	1-1
1.2 How OPMN Works .....	1-2
1.2.1 Oracle Notification Server .....	1-3
1.2.2 Oracle Process Manager .....	1-3
1.2.3 PM Modules .....	1-4
1.3 What Oracle Application Server Components Does OPMN Manage? .....	1-4
1.3.1 Oracle Enterprise Manager 10g Application Server Control Console .....	1-5
1.4 OPMN Configuration Files .....	1-5
1.5 Automatic Restart .....	1-6
1.6 opmnctl .....	1-7
1.6.1 opmnctl Syntax .....	1-7
1.7 opmnctl Command Quick Reference .....	1-8
1.8 opmnctl Detailed Command Description .....	1-8
1.8.1 Server Control Commands .....	1-8
1.8.1.1 Server Control Commands on Microsoft Windows .....	1-9
1.8.1.2 opmnctl start .....	1-9
1.8.1.3 opmnctl startall .....	1-9
1.8.1.4 opmnctl stopall .....	1-9
1.8.1.5 opmnctl shutdown .....	1-10
1.8.1.6 opmnctl reload .....	1-11
1.8.2 Process Control Commands .....	1-11
1.8.2.1 opmnctl startproc, opmnctl restartproc and opmnctl stopproc .....	1-12
1.8.3 Command Definitions .....	1-13
1.8.3.1 Scope .....	1-13
1.8.3.2 Attributes .....	1-14
1.8.3.3 Verbose .....	1-15
1.8.4 Status Commands .....	1-15
1.8.4.1 opmnctl status .....	1-16

1.8.4.1.1	Options for the Status Command of opmnctl .....	1-17
1.8.4.2	opmnctl dmsdump .....	1-18
1.8.4.3	opmnctl ping .....	1-19
1.8.5	Help Commands .....	1-19
1.8.5.1	opmnctl help .....	1-20
1.8.5.2	opmnctl usage .....	1-20
1.8.5.3	opmnctl validate .....	1-21
1.9	Event Scripts .....	1-21
1.10	Start Order Dependencies .....	1-22
1.11	OPMN Log Files .....	1-22
1.12	Security .....	1-22
1.12.1	Remote Security .....	1-24
1.13	DCM .....	1-24
1.14	iHAT .....	1-24

## 2 Using OPMN

2.1	Starting OPMN .....	2-1
2.2	Starting and Stopping OPMN-Managed Processes for a Local Oracle Application Server Instance .....	2-1
2.3	Starting and Stopping all OPMN Managed Processes for a Remote Oracle Application Server Instance .....	2-2
2.4	Starting and Stopping an Oracle Application Server Component in a Local Oracle Application Server Instance .....	2-2
2.5	Starting and Stopping an Oracle Application Server Process Type in a Local Oracle Application Server Instance .....	2-2
2.6	Starting and Stopping a Multi-Oracle Application Server Instance Environment .....	2-2
2.7	Starting a Component on an Oracle Application Server Farm .....	2-3

## 3 opmn.xml Common Configuration

3.1	Example of opmn.xml Elements and Attributes .....	3-1
3.2	opmn.xml Element and Attribute Descriptions .....	3-3

## 4 Configuring Oracle HTTP Server

4.1	Oracle HTTP Server Process Module Configuration .....	4-1
4.2	Oracle HTTP Server Minimum Configuration .....	4-2
4.3	Oracle HTTP Server Complete Configuration .....	4-2
4.4	Oracle HTTP Server Attribute Descriptions .....	4-2
4.5	Oracle HTTP Server 2 .....	4-5
4.6	Generic Apache (UNIX only) .....	4-6

## 5 Configuring OC4J

5.1	OC4J Process Module Configuration .....	5-1
5.2	OC4J Minimum Configuration .....	5-1
5.3	OC4J Complete Configuration .....	5-1
5.4	OC4J Attribute Descriptions .....	5-2

## **6 Configuring Oracle Application Server Web Cache**

6.1	OracleAS Web Cache Process Module Configuration .....	6-1
6.2	OracleAS Web Cache Minimum Configuration.....	6-1
6.3	OracleAS Web Cache Complete Configuration .....	6-2
6.4	OracleAS Web Cache Attribute Descriptions .....	6-2

## **7 Configuring Oracle Internet Directory**

7.1	Oracle Internet Directory Process Module Configuration.....	7-1
7.2	Oracle Internet Directory Minimum Configuration .....	7-1
7.3	Oracle Internet Directory Complete Configuration.....	7-2
7.4	Oracle Internet Directory Attribute Descriptions .....	7-2

## **8 Configuring Oracle Application Server Port Tunnel**

8.1	OracleAS Port Tunnel Process Module Configuration .....	8-1
8.2	OracleAS Port Tunnel Minimum Configuration.....	8-1
8.3	OracleAS Port Tunnel Complete Configuration .....	8-1
8.4	OracleAS Port Tunnel Attribute Descriptions .....	8-2

## **9 Configuring Oracle Application Server Wireless**

9.1	OracleAS Wireless Process Module Configuration .....	9-1
9.2	OracleAS Wireless Minimum Configuration.....	9-1
9.3	OracleAS Wireless Complete Configuration .....	9-5
9.4	OracleAS Wireless Attribute Descriptions.....	9-10

## **10 Configuring Oracle Application Server Reports Services**

10.1	Introduction .....	10-1
10.1.1	Reports Server .....	10-1
10.1.2	Reports Bridge.....	10-2
10.2	OracleAS Reports Services Process Module Configuration .....	10-2
10.3	Reports Server Minimum Configuration .....	10-2
10.3.1	In-process Reports Server Minimum Configuration.....	10-2
10.3.2	Standalone Reports Server Minimum Configuration .....	10-2
10.4	Reports Server Full Configuration .....	10-3
10.4.1	In-process Reports Server Full Configuration.....	10-3
10.4.2	Standalone Reports Server Full Configuration .....	10-3
10.5	Reports Bridge Minimum Configuration .....	10-4
10.6	Reports Bridge Full Configuration.....	10-4
10.7	OracleAS Reports Services Attribute Descriptions.....	10-4

## **11 Configuring Oracle Business Intelligence Discoverer**

11.1	OracleBI Discoverer Process Module Configuration.....	11-1
11.2	OracleBI Discoverer Minimum Configuration.....	11-1
11.3	OracleBI Discoverer Complete Configuration.....	11-2
11.4	OracleBI Discoverer Attribute Descriptions .....	11-2

## **12 Configuring Log Loader**

12.1	Log Loader Process Module Configuration.....	12-1
12.2	Log Loader Minimum Configuration.....	12-1
12.3	Log Loader Complete Configuration.....	12-2
12.4	Log Loader Attribute Descriptions .....	12-2

## **13 Configuring DCM Daemon**

13.1	DCM Daemon Process Module Configuration .....	13-1
13.2	DCM Daemon Minimum Configuration.....	13-1
13.3	DCM Daemon Complete Configuration .....	13-1
13.4	DCM Daemon Attribute Descriptions .....	13-2

## **14 Configuring Oracle Application Server Integration Business Activity Monitoring**

14.1	OracleAS Integration BAM Process Module Configuration .....	14-1
14.2	OracleAS Integration BAM Minimum Configuration.....	14-1
14.3	OracleAS Integration BAM Complete Configuration .....	14-2
14.4	OracleAS Integration BAM Attribute Descriptions .....	14-3

## **15 Configuring Custom Process**

15.1	Custom Process Module Configuration .....	15-1
15.2	Custom Process Minimum Configuration .....	15-1
15.3	Custom Process Complete Configuration.....	15-1
15.3.1	Ping.....	15-2
15.4	Custom Process Attribute Descriptions.....	15-3

## **A OPMN Troubleshooting**

A.1	Problems and Solutions .....	A-1
A.1.1	Oracle Application Server Process Does Not Start.....	A-2
A.1.2	Determining if Oracle Application Server Processes are Dying or Unresponsive....	A-2
A.1.3	opmnctl Command Execution Times Out.....	A-3
A.1.4	Oracle Application Server Component Automatically Restarted by OPMN .....	A-3
A.1.5	Unexpected opmnctl start Behavior.....	A-4
A.1.6	Disabled Element in the opmn.xml File .....	A-4
A.1.7	Management of Oracle9iAS Release 2 (9.0.2 and 9.0.3) Instances .....	A-5
A.1.8	Unable to Start OC4J .....	A-5
A.1.9	Unable to Stop Component .....	A-5
A.1.10	globalInitNLS Error .....	A-6
A.1.11	Start Remote Hosts of a Cluster Independently.....	A-6
A.1.12	Log Loader Not Started .....	A-6
A.1.13	dcm-daemon Not Started .....	A-7
A.1.14	SSL Handshake Failed .....	A-8
A.1.15	OPMN Start Up Consumes CPU Processing Capability .....	A-8
A.1.16	Error Messages During Start-up of OPMN.....	A-8
A.2	Diagnosing OPMN Problems.....	A-9

A.2.1	OPMN log Files .....	A-9
A.2.2	opmnctl debug .....	A-10
A.2.3	iHAT .....	A-10
A.2.4	Oracle Enterprise Manager 10g Application Server Control Console .....	A-11
A.2.5	Troubleshooting with Event Scripts.....	A-11
A.2.6	opmn.xml Environment Variables .....	A-12
A.3	Need More Help? .....	A-12

## Index





---

---

# Preface

This guide describes how to administer Oracle Process Manager and Notification Server (OPMN) in order to manage Oracle Application Server components.

This preface contains these topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documentation](#)
- [Conventions](#)

## Audience

The *Oracle Process Manager and Notification Server Administrator's Guide* is intended for administrators of Oracle Application Server.

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

### Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

### Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

### TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

## Related Documentation

For more information, see these Oracle resources:

- Oracle Application Server Documentation Library
- Oracle Application Server Platform-Specific Documentation on Oracle Application Server Disk 1

## Conventions

The following text conventions are used in this document:

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

---

# OPMN: Overview

This chapter provides an overview of OPMN for Oracle Application Server. It features the following topics:

- [Section 1.1, "What is OPMN?"](#)
- [Section 1.2, "How OPMN Works"](#)
- [Section 1.3, "What Oracle Application Server Components Does OPMN Manage?"](#)
- [Section 1.4, "OPMN Configuration Files"](#)
- [Section 1.5, "Automatic Restart"](#)
- [Section 1.6, "opmnctl"](#)
- [Section 1.7, "opmnctl Command Quick Reference"](#)
- [Section 1.8, "opmnctl Detailed Command Description"](#)
- [Section 1.9, "Event Scripts"](#)
- [Section 1.10, "Start Order Dependencies"](#)
- [Section 1.11, "OPMN Log Files"](#)
- [Section 1.12, "Security"](#)
- [Section 1.13, "DCM"](#)
- [Section 1.14, "iHAT"](#)

## 1.1 What is OPMN?

OPMN is installed and configured with every Oracle Application Server installation type and is essential for running Oracle Application Server.

OPMN features the following functionality:

- Provides a command-line interface for process control and monitoring for single or multiple Oracle Application Server components and instances.
- Provides an integrated way to manage Oracle Application Server components.
- Enables management of Oracle Application Server subcomponents and sub-subcomponents.
- Channels all events from different Oracle Application Server component instances to all Oracle Application Server components that can utilize them.
- Solves interdependency issues between Oracle Application Server components by enabling you to start and stop components in order.

- Enables customizing of enterprise functionality by using event scripts.
- Enables gathering of host and Oracle Application Server process statistics and tasks.
- Provides automatic restart of Oracle Application Server processes when they become unresponsive, terminate unexpectedly, or become unreachable as determined by ping and notification operations.
- Provides automatic death detection of Oracle Application Server processes.
- Does not depend on any other Oracle Application Server component being up and running before it can be started and used.

The OPMN server should be started as soon as possible after turning on the host. OPMN must be running whenever OPMN-managed components are turned on or off.

---

**Note:** On the Microsoft Windows operating system, OPMN is installed as a Windows service (Oracle<OracleHomename>ProcessManager). It starts up automatically when you start or restart your computer. Refer to [Section 1.8.1.3, "opmnctl startall"](#) for more information.

---

Oracle Application Server components managed by OPMN should never be started or stopped manually. Do not use command line scripts or utilities from previous versions of Oracle Application Server for starting and stopping Oracle Application Server components. OPMN must be the last service turned off whenever you restart or turn off your computer.

Use the Application Server Control Console and the `opmnctl` command line utility to start or stop Oracle Application Server components.

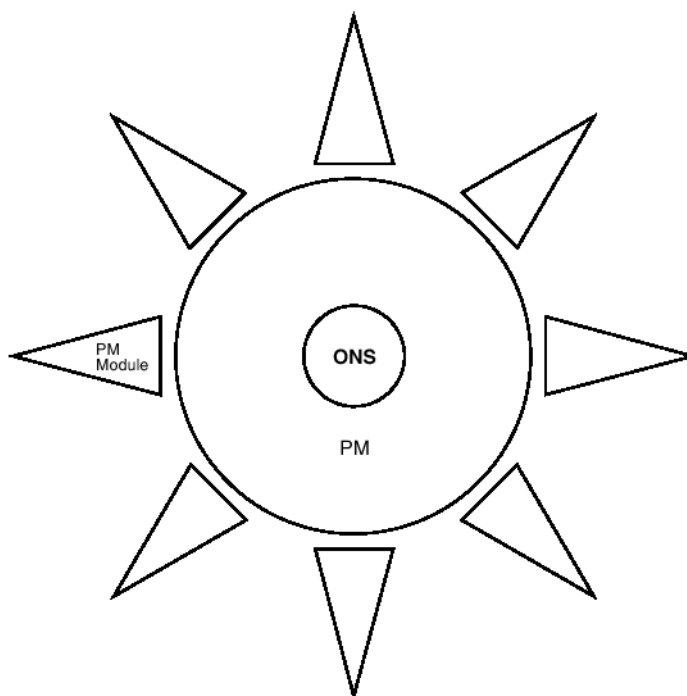
## 1.2 How OPMN Works

OPMN consists of a core grouping of three components that interpret and convey notification information sent between Oracle Application Server processes within the same or different OPMN servers.

The core of OPMN consists of the following three components:

- [Oracle Notification Server](#)
- [Oracle Process Manager](#)
- [PM Modules](#)

[Figure 1-1](#) shows the architecture of the core of OPMN.

**Figure 1–1 OPMN Architecture**

### 1.2.1 Oracle Notification Server

Oracle Notification Server (ONS) is the transport mechanism for failure, recovery, startup, and other related notifications between components in Oracle Application Server. It operates according to a publish-subscribe model: an Oracle Application Server component receives a notification of a certain type for each subscription to ONS. When such a notification is published, ONS sends it to the appropriate subscribers.

### 1.2.2 Oracle Process Manager

Oracle Process Manager (PM) is the centralized process management mechanism in Oracle Application Server and is used to manage Oracle Application Server processes. The PM is responsible for starting, restarting, stopping, and monitoring every process it manages. The PM handles all requests sent to OPMN associated with controlling a process or obtaining status about a process. The PM is also responsible for performing death-detection and automatic restart of the processes it manages. The Oracle Application Server processes that PM is configured to manage are specified in the `opmn.xml` file.

The PM waits for a user command to start a specific, or all Oracle Application Server processes. When a process is stopped, the PM receives a request as specified by the request parameters.

The OPMN server consists of 2 processes. The first OPMN server process has only one purpose: to start the second OPMN server process when necessary. The second OPMN server process handles all request traffic and does all the work. If the second OPMN server process goes down as part of an `opmnctl reload` command or an unexpected crash it will be restarted by the first OPMN server process.

On Microsoft Windows, the second OPMN server process will not be restarted if it is deliberately terminated. Instead, the first OPMN server process will exit as well. Recovering from this situation is accomplished by restarting the OPMN server from the command line or service manager.

The PM uses the ONS to:

- detect that a process has completed initialization and is ready to receive requests
- determine what ports are in use
- obtain component specific runtime information

The Application Server Control Console also uses PM to manage processes.

### 1.2.3 PM Modules

The Oracle Process Manager Modules (PM Modules) implement Oracle Application Server component-specific process management functionality. The PM Modules pass notification information returned by other Oracle Application Server component PM Modules within the same or different OPMN servers.

The PM Modules:

- handle any communications originating from the running component.
- construct Oracle Application Server component specific control information (how to start, stop, restart the component).
- test responsiveness in an Oracle Application Server component specific manner to determine if a component is responding to requests.

## 1.3 What Oracle Application Server Components Does OPMN Manage?

OPMN manages all Oracle Application Server components except the OracleAS Metadata Repository or the Application Server Control Console.

OPMN enables you to explicitly manage the following list of Oracle Application Server 10g Release 2 (10.1.2) components:

- Oracle HTTP Server
- Oracle Application Server Containers for J2EE (OC4J)
- Distributed Configuration Management (DCM) daemon (server)
- OracleAS Log Loader
- Oracle Internet Directory
- OracleAS Port Tunnel
- OracleAS Web Cache
- OracleBI Discoverer
- OracleAS Wireless

OPMN also manages other Oracle Application Server components implicitly; the constituent parts of the implicit Oracle Application Server components are managed by OPMN as part of one or more other Oracle Application Server components. For example, OracleAS Portal while not explicitly managed by OPMN is implicitly managed because it is operational using OC4J and Oracle HTTP Server.

The following components are implicitly managed by OPMN:

- OracleAS Single Sign-On
- Oracle HTTP Server Adapter of OracleAS Integration InterConnect
- Oracle Distributed Authoring and Versioning (OraDAV)
- OracleAS Portal
- OracleAS SOAP
- OracleAS UDDI Registry Server
- Oracle Ultra Search
- OracleAS TopLink

You can also configure OPMN to manage other processes using the Custom PM Module. See [Chapter 15, "Configuring Custom Process"](#) for more information.

Because of the extensible design of OPMN, add-on components are managed by OPMN without having to update OPMN itself.

**See Also:**

- [Chapter 8, "Configuring Oracle Application Server Port Tunnel"](#)
- [Chapter 15, "Configuring Custom Process"](#)

### 1.3.1 Oracle Enterprise Manager 10g Application Server Control Console

In addition to OPMN, you can also manage your enterprise using the Application Server Control Console. Application Server Control Console leverages the functionality of OPMN to manage your Oracle Application Server enterprise. Using a Web browser, Application Server Control Console provides a graphical interface that enables management of all Oracle Application Server components in your network and enterprise.

**See Also:** *Oracle Application Server Administrator's Guide*

## 1.4 OPMN Configuration Files

The `opmn.xml` file does not contain component-specific element names. Component specific management code is located in the PM modules which get loaded by OPMN at startup according to what has been specified in the **modules** section of the `opmn.xml` file.

Each level has a specific set of configurations. In addition, there are several configuration elements that are accepted at more than one level to provide the flexibility of applying a configuration across an entire Oracle Application Server component or just part of a component.

OPMN reads the following configuration files when started or reloaded:

- `ORACLE_HOME/opmn/conf/opmn.xml`

The `opmn.xml` file is the main configuration file for OPMN. The `opmn.xml` file contains information for the ONS, the PM, and Oracle Application Server component specific configuration. The `opmn.xml` file shows you which Oracle Application Server components OPMN is managing on your system.

The `opmn.xml` file contains Oracle Application Server component entries arranged in the following hierarchical structure:

```
<ias-component>
```

```
<process-type>
  <process-set>
```

**<ias-component>**: This entry represents the Oracle Application Server component. It enables management of the component for processes such as starting and stopping.

**<process-type>**: This subcomponent of the **<ias-component>** entry declares the type of process to run by association with a specific PM module.

**<process-set>**: This sub-subcomponent of the **<ias-component>** entry enables you to declare different sets of optional runtime arguments and environments for the Oracle Application Server component.

[Example 1–1](#) shows an example of the **<ias-component>**, **<process-type>**, **<process-set>** entries in an `opmn.xml` file.

#### **Example 1–1** *opmn.xml* file

```
<ias-component id="OC4J">
  <process-type id="home">
    <process-set id="default_island">....
```

You can edit `opmn.xml` using Application Server Control Console. Click the **Process Management** link at the bottom of the Oracle Application Server instance home page. Do not stop the OPMN server after you edit the `opmn.xml` file. Application Server Control Console automatically reloads the updated `opmn.xml` file after you edit the file.

If you manually edit the `opmn.xml` file run the `dcmctl updateConfig` command on the command line. `dcmctl updateConfig` reloads the updated file and updates the configuration repository with the manual changes.

- `ORACLE_HOME/opmn/conf/ons.conf`

OPMN uses the `ons.conf` file to find all the different Oracle Application Server instances in the farm. The `ons.conf` file is automatically generated and maintained by DCM. Do not edit the `ons.conf` file; edits to this file are overwritten by DCM.

- `ORACLE_HOME/dcm/config/dcm.conf`

The `dcm.conf` file specifies the instance name and cluster name of the local Oracle Application Server installation. The `dcm.conf` file is automatically generated and maintained by DCM. The `dcm.conf` file should not be edited manually.

## 1.5 Automatic Restart

OPMN gives the user control over automatic death detection and restart of components; you can configure the parameters by which OPMN determines a process has died and disable automatic restart for individual components.

OPMN monitors the operation of its managed processes by the following methods:

- Operating system level detection of Oracle Application Server process death
- Periodic ping requests to Oracle Application Server processes
- Periodic status notification from Oracle Application Server processes ("reverse-ping")



The ping and notification functionality is only used where appropriate according to the functionality of the Oracle Application Server component.

OPMN automatically restarts Oracle Application Server components that terminate unexpectedly. OPMN will also restart processes that are unresponsive according to the result of notification and ping operations.

**See Also:**

- [Chapter 3, "opmn.xml Common Configuration"](#)
- [Section A.1.5, "Unexpected opmnctl start Behavior"](#)

## 1.6 opmnctl

opmnctl is the supported tool for starting and stopping all components in an Oracle Application Server instance, with the exception of the Oracle Application Server Metadata Repository and the Oracle Enterprise Manager 10g Application Server Control Console (Application Server Control Console). opmnctl provides a centralized way to control and monitor Oracle Application Server components from the command line. You can use opmnctl to execute control and monitoring commands across multiple Oracle Application Server instances simultaneously.

opmnctl also enables you to perform operations on a specified Oracle Application Server instance on the application server farm, all instances in the farm, and all instances in a cluster using an optional parameter called *scope*. You can also use the *scope* option to control an individual Oracle Application Server process.

The opmnctl command is located in the following directory locations:

(UNIX) `ORACLE_HOME/opmn/bin/opmnctl`

(Windows) `ORACLE_HOME\opmn\bin\opmnctl`

---

**Note:** Oracle Application Server components managed by OPMN should never be started or stopped manually. Do not use command line scripts or utilities from previous versions of Oracle Application Server for starting and stopping Oracle Application Server components. Use the Application Server Control Console and the opmnctl command line utility to start or stop Oracle Application Server components.

---



---

**Note:** Oracle recommends starting OPMN as the user that has installed Oracle Application Server.

---

### 1.6.1 opmnctl Syntax

The following command shows an example of the syntax of the opmnctl command:

```
opmnctl [verbose] [<scope>] <command> [<options>]
```

[Table 1–1](#) provides a description about opmnctl syntax.

**Table 1–1** *opmnctl Syntax*

Syntax	Description
verbose	Prints detailed execution message, if available.

**Table 1–1 (Cont.) opmnctl Syntax**

Syntax	Description
scope	Specifies where the request is routed. Refer to <a href="#">Section 1.8.3.1, "Scope"</a> for a list of options.
command	Specifies an opmnctl command. Refer to <a href="#">Example 1–2</a> for a list of commands.
options	Specifies options for the command. Refer to <a href="#">Section 1.8.4.1.1, "Options for the Status Command of opmnctl"</a> for a list of options.

## 1.7 opmnctl Command Quick Reference

[Example 1–2](#) lists opmnctl commands for quick reference. You can obtain the same output information by executing the opmnctl help command.

### Example 1–2 opmnctl Commands

```
prompt >
opmnctl help
```

scope	command	options
	start	- Start opmn
	startall	- Start opmn and all managed processes
	stopall	- Stop opmn and all managed processes
	shutdown	- Shutdown opmn and all managed processes
[<scope>]	startproc	[<attr>=<val>...] - Start opmn managed processes
[<scope>]	restartproc	[<attr>=<val>...] - Restart opmn managed processes
[<scope>]	stopproc	[<attr>=<val>...] - Stop opmn managed processes
[<scope>]	reload	- Trigger opmn to reread opmn.xml
[<scope>]	status	[<options>] - Get managed process status
[<scope>]	dmsdump	[<attr>=<val>&...] - Get DMS stats
	ping	[<max_retry>] - Ping local opmn
	validate	[<filename>] - Validate the given xml file
	help	- Print brief usage description
	usage	[<command>] - Print detailed usage description

## 1.8 opmnctl Detailed Command Description

The following sections contains detailed descriptions of the opmnctl commands listed in [Example 1–2](#). The opmnctl commands are displayed in the following sections:

- [Section 1.8.1, "Server Control Commands"](#)
- [Section 1.8.2, "Process Control Commands"](#)
- [Section 1.8.3, "Command Definitions"](#)
- [Section 1.8.4, "Status Commands"](#)
- [Section 1.8.5, "Help Commands"](#)

### 1.8.1 Server Control Commands

The opmnctl start, startall, reload, stopall, and shutdown commands enable you to control the OPMN server.

- [Section 1.8.1.2, "opmnctl start"](#)
- [Section 1.8.1.3, "opmnctl startall"](#)

- [Section 1.8.1.4, "opmnctl stopall"](#)
- [Section 1.8.1.5, "opmnctl shutdown"](#)
- [Section 1.8.1.6, "opmnctl reload"](#)

Output is not generated for the successful execution of an `opmnctl` server control command. Refer to [Appendix A, "OPMN Troubleshooting"](#) if you receive any error messages during `opmnctl` command execution.

### 1.8.1.1 Server Control Commands on Microsoft Windows

On the Microsoft Windows operating system, OPMN is installed as a Windows service (Oracle<OracleHomename>ProcessManager) and it starts up automatically when you restart your computer. When you start or stop OPMN using Windows Services you start or stop *all* OPMN-managed components on the local instance.

Use the Application Server Control Console and the `opmnctl` command line utility to start or stop Oracle Application Server components.

#### 1.8.1.2 opmnctl start

Syntax: `opmnctl start`

Use this command to start the OPMN server for a local Oracle Application Server instance without starting OPMN-managed processes.

Execute this command as soon as possible after starting your computer.

OPMN starts up automatically on Microsoft Windows when you start or restart your computer. All OPMN-managed processes are started.

**See Also:** [Chapter 2, "Using OPMN"](#)

#### 1.8.1.3 opmnctl startall

Syntax: `opmnctl startall [timeout=<seconds>]`

Use this command to start OPMN as well as the OPMN managed processes for a local Oracle Application Server instance. The `startall` is equivalent to the `start` command and the `startproc` command without arguments. Oracle recommends using the `start` or `startproc` command.

This command operates synchronously and waits for the operation to complete before returning. To set a timeout for the request, specify the timeout value in seconds.

Components with `id-matching="true"` will not be started.

Enter the following command for additional detailed information:

```
prompt > opmnctl usage startall
stopall
start startall startproc
```

On Microsoft Windows, you can also perform an `opmnctl startall` by starting the Oracle<OracleHomename>ProcessManager service in the Windows services control panel. The Oracle<OracleHomename>ProcessManager starts automatically when you start or restart your computer.

#### 1.8.1.4 opmnctl stopall

Syntax: `opmnctl stopall`

Use the `opmnctl stopall` command to shut down the OPMN server as well as the OPMN-managed processes for the local Oracle Application Server instance. This request operates synchronously; it waits for the operation to complete before returning.

Shutting down the OPMN server is not necessary during normal operation. Shutting down the OPMN server prevents remote commands to OPMN from executing on the Oracle Application Server instance until OPMN is restarted.

The `opmnctl stopall` command should only be executed prior to shutting down OPMN and your computer. This request first tries to stop all OPMN-managed processes gracefully. Processes which will not stop gracefully will be forcefully shutdown. After stopping all managed processes, the OPMN daemon will shutdown itself.

The `opmnctl stopall` command should only be used when it is necessary to stop the OPMN daemon. Once started, the OPMN daemon should remain up until it is necessary to restart the computer or some other unforeseen administrative event occurs.

To stop all OPMN-managed processes without stopping the OPMN daemon, consider using the `opmnctl stopproc` command without any arguments.

To restart the OPMN daemon without restarting any OPMN-managed processes, consider using the `opmnctl reload` command. The `opmnctl reload` command is the appropriate command to use when the only goal is to restart the opmn daemon with a new configuration.

Use the `opmnctl stopproc` command if you want to stop all OPMN managed processes.

Use the `opmnctl reload` if you want OPMN to reread its configuration.

Enter one of the following commands to obtain additional information:

```
prompt > opmnctl usage stopall
```

or

```
prompt > opmnctl usage shutdown
```

### **1.8.1.5 opmnctl shutdown**

Syntax: `opmnctl shutdown`

Use the `opmnctl shutdown` command to shut down the OPMN server as well as the OPMN-managed processes for the local Oracle Application Server instance.

The `opmnctl shutdown` command quickly shutdowns the OPMN daemon and OPMN-managed processes for the local Oracle Application Server instance.

The `opmnctl shutdown` command is similar to the `opmnctl stopall` command but waits less time before initiating a forceful termination of OPMN-managed processes. After all of the OPMN-managed processes are stopped, the OPMN daemon will shutdown itself.

The `opmnctl shutdown` command should only be performed when it is necessary to stop the OPMN daemon. Once started, the OPMN daemon should remain up until it is necessary to restart the computer or some other unforeseen administrative event occurs.

To stop all OPMN-managed processes without stopping the OPMN daemon, consider using the `opmnctl stopproc` command without any arguments.

To restart the OPMN daemon without restarting any OPMN-managed processes, consider using the `opmnctl reload` command. The `opmnctl reload` command is the appropriate command to use when the objective is to restart the OPMN daemon with a new configuration.

On Microsoft Windows, you can also perform an `opmnctl shutdown` by stopping the Oracle<OracleHomename>ProcessManager service in the Windows services control panel.

Use the `opmnctl stopproc` command if you want to stop all OPMN managed processes.

Use the `opmnctl reload` if you want OPMN to reread its configuration.

Enter one of the following commands to obtain additional information:

```
prompt > opmnctl usage stopall
```

or

```
prompt > opmnctl usage shutdown
```

### 1.8.1.6 opmnctl reload

Syntax: `opmnctl [scope] reload`

Use this command to trigger the OPMN to reread its configuration files in the requested scope. This command restarts the OPMN server without restarting any Oracle Application Server processes managed by OPMN. The OPMN server for the Oracle Application Server instance must be up and running.

---

**Note:** On Microsoft Windows, you can highlight the Oracle<OracleHomename>ProcessManager in the services control panel and select **Restart**. The restart of the service is not equivalent to an `opmnctl reload`, however. This action is equivalent to an `opmnctl shutdown` followed by an `opmnctl startall`. It is a much slower operation than `opmnctl reload` because it restarts OPMN and all the processes managed by OPMN.

---

Enter the following command for additional detailed information:

```
prompt > opmnctl usage reload
```

**See Also:** [Section 1.8.3, "Command Definitions"](#)

## 1.8.2 Process Control Commands

The `opmnctl` process control commands enable you to start, stop, or restart single or multiple Oracle Application Server components. You can control an Oracle Application Server component at the <ias-component>, <process-set>, or <process-type> level.

This section describes the process control commands available with `opmnctl`. It includes the following process control commands:

- [Section 1.8.2.1, "opmnctl startproc, opmnctl restartproc and opmnctl stopproc"](#)

Output is not generated for the successful execution of an `opmnctl` process control command. Refer to [Appendix A, "OPMN Troubleshooting"](#) if you receive any error messages during `opmnctl` command execution.

### 1.8.2.1 `opmnctl startproc`, `opmnctl restartproc` and `opmnctl stopproc`

Syntax: `opmnctl [<scope>] startproc [<attr>=<value>...]`

`opmnctl [<scope>] restartproc [<attr>=<value>...]`

`opmnctl [<scope>] stopproc [<attr>=<value>...]`

Use these commands to start, restart, or stop OPMN-managed processes in the requested scope. The OPMN server for the Oracle Application Server instance must be up and running.

The following attributes and values can be used with the `startproc`, `stopproc`, and `restartproc` commands:

- `ias-component`, `process-type`, and `process-set`: The values for these attributes should be the same as the `id` value specified in the `opmn.xml` file. If no attribute is supplied, the command is applied to all OPMN-managed processes other than those that are configured in the `opmn.xml` file with `id-matching="true"`. To execute commands on components configured with `id-matching="true"`, it is necessary to specify the `ias-component` argument.
- `mode`: The `mode` attribute value can be either `sync` or `async`; the default value is `sync`. The `sync` value for `mode` causes the `opmnctl` command to operate synchronously and wait for the command to be executed completely before a return prompt is displayed. The timeout element can only be specified when the value of `mode` is `sync`. The value is specified in number of seconds. After the specified timeout expires, the operation is aborted for `startproc` but not for `restartproc` or `stopproc`. The `opmnctl` command prompt returns, the OPMN server continues to perform the `opmnctl restartproc` or `stopproc` command request until the operation is finished.  
  
The `async` value for `mode` causes the return prompt to be displayed immediately, while the OPMN server continues to perform the `opmnctl` command request until the operation is finished.
- `uniqueid`: This value is assigned by OPMN after starting up. You can use this value when you execute the `restartproc` and `stopproc` commands. You can obtain this value by entering the following command and obtaining the unique number for the Oracle Application Server component in the `uid` column of the generated output:

```
prompt > opmnctl status -l
```

Attribute names other than those listed may be specified for some types of Oracle Application Server processes managed by OPMN. Unique attribute name should be specific to each type of Oracle Application Server process.

Using the `opmnctl startproc`, `restartproc`, or `stopproc` commands with a specified scope and attributes enables control of specific processes in your enterprise. You can execute the `opmnctl startproc`, `restartproc`, or `stopproc` commands at the `<ias-component>`, `<process-type>` and the `<process-set>` level.

For example, the following command starts OracleAS Wireless at the `<process-set>` level:

```
prompt > opmnctl startproc ias-component=wireless process-type=alert_server
```

```
process-set=alert_instance_1
```

The following command restarts OC4J at the <process-type> level:

```
prompt > opmnctl restartproc ias-component=OC4J process-type=home
```

The following command stops Oracle HTTP Server at the <ias-component> level:

```
prompt > opmnctl stopproc ias-component=HTTP_Server
```

Enter one of the following commands to obtain additional information:

```
prompt > opmnctl usage startproc
```

or

```
prompt > opmnctl usage restartproc
```

or

```
prompt > opmnctl usage stopproc
```

#### See Also:

- [Section 1.8.3, "Command Definitions"](#)
- [Chapter 2, "Using OPMN"](#)

## 1.8.3 Command Definitions

opmnctl features command definitions that enable you to further define the action you would like to execute with OPMN.

This section describes the command definitions available with the opmnctl command. It includes the following sections:

- [Section 1.8.3.1, "Scope"](#)
- [Section 1.8.3.2, "Attributes"](#)
- [Section 1.8.3.3, "Verbose"](#)

### 1.8.3.1 Scope

Syntax: @instance[:instance[:instance...]]

@cluster[:clusname[:clusname...]]

@farm

The scope option specifies which Oracle Application Server instances the opmnctl command applies to. You can use the scope option for opmnctl commands for single or multiple Oracle Application Server instances, clusters and farms.

- **@instance:** If you do not specify a name after @instance option, the opmnctl command is applied to the local Oracle Application Server instance; local refers to the Oracle Application Server instance or cluster containing the OPMN server handling the request. The default is the local Oracle Application Server instance. If the @instance option is followed by Oracle Application Server instance names, the request will be routed to Oracle Application Server instances. To apply the command to one or more Oracle Application Server instances, specify @instance[:instance[:instance...]].
- **@cluster:** If you do not specify a name after @cluster option, the opmnctl command is applied to the local Oracle Application Server cluster. If @cluster is

followed by a set of 1 or more cluster names, the request will be routed to the all Oracle Application Server instances contained in the specified Oracle Application Server clusters. To apply the command to all Oracle Application Server instances within one or more Oracle Application Server clusters, specify `@cluster[:clusname[:clusname...]]`.

- **@farm:** To apply the command query to all Oracle Application Server instances in the Oracle Application Server farm.

For example, the following command starts OC4J on Oracle Application Server instance named "myInst2.foo.com":

```
prompt > opmnctl @instance:myInst2.foo.com startproc ias-component=HTTP_Server
```

**See Also:** [Chapter 2, "Using OPMN"](#)

### 1.8.3.2 Attributes

syntax: `<attribute>=<value>`

The `opmnctl` attributes enable you to apply process control operations to specific Oracle Application Server components. If no attributes are specified, all OPMN-managed processes are started, stopped, or restarted.

For example, the following command starts all Oracle Application Server processes configured for OracleAS Wireless:

```
prompt > opmnctl startproc ias-component=wireless
```

Refer to [Chapter 2, "Using OPMN"](#) for additional `opmnctl` command examples.

[Table 1–2](#) lists the attribute names and values that OPMN can use this command:

**Table 1–2** *opmnctl Attribute Names and Values*

Attribute Name	Attribute Values
ias-component	Value should be the same as the value for the <code>id</code> attribute for the <code>&lt;ias-component&gt;</code> element in the <code>opmn.xml</code> file.
process-type	Value should be the same as the value for the <code>id</code> attribute for the <code>&lt;process-type&gt;</code> element in the <code>opmn.xml</code> file.
process-set	Value should be the same as the value for the <code>id</code> attribute for the <code>&lt;process-set&gt;</code> element in the <code>opmn.xml</code> file.
mode	Value can either be "sync" or "async". The default value is "sync", meaning that this request operates synchronously, and waits for the operation to complete before returning. "async" indicates that the request returns immediately, while OPMN continues to perform the request until the operation finishes.
timeout	This can only be specified in "sync" mode. The value is in seconds. After this timeout expires, OPMN does not continue to perform the request for <code>startproc</code> operations. The request does continue for <code>restartproc</code> and <code>stopproc</code> operations.
uniqueid	This value is assigned by OPMN after starting up. You can use this value when you execute the <code>opmnctl restartproc</code> and <code>opmnctl stopproc</code> commands.

**See Also:** [Chapter 2, "Using OPMN"](#)



### 1.8.3.3 Verbose

Syntax: `opmnctl verbose command`

The `opmnctl verbose` option enables you to obtain detailed information about the command you are executing.

For example, the following command outputs the information shown in [Example 1-3](#):

```
prompt> opmnctl verbose startproc ias-component=HTTP_Server
```

#### **Example 1-3** *opmnctl verbose output*

```
HTTP/1.1 200 OK
Content-Length: 0
Content-Type: text/html
Response: Ping succeeded.
```

```
opmnctl: starting opmn managed processes...
```

```
HTTP/1.1 200 OK
Content-Length: 571
Content-Type: text/html
Response: 1 of 1 processes started.
```

```
<response>
<opmn id="jerichar-sun.us.oracle.com:6200" http-status="200" http-response="1 of 1
processes started.">
  <ias-instance id="M140801.jerichar-sun.us.oracle.com">
    <ias-component id="HTTP_Server">
      <process-type id="HTTP_Server">
        <process-set id="HTTP_Server">
          <process id="1954086921" pid="9355" status="Alive" index="1"
            log="/home/demoas/M140801/opmn/logs/HTTP_Server~1"
            operation="request" result="success">
          </process>
        </process-set>
      </process-type>
    </ias-component>
  </ias-instance>
</opmn>
</response>
```

## 1.8.4 Status Commands

The `opmnctl status` commands enable you to determine the status of OPMN-managed processes.

This section describes the command options available with the `opmnctl` command. It includes the following sections:

- [Section 1.8.4.1, "opmnctl status"](#)
- [Section 1.8.4.2, "opmnctl dmsdump"](#)
- [Section 1.8.4.3, "opmnctl ping"](#)

**See Also:**

- [Section 1.8.3.1, "Scope"](#)
- [Section 1.8.3.2, "Attributes"](#)
- [Section 1.8.4.1.1, "Options for the Status Command of opmnctl"](#)

**1.8.4.1 opmnctl status**

Syntax: `opmnctl [<scope>] status [<options>]`

The `status` command enables you to obtain information on the Oracle Application Server processes managed by OPMN.

The output is a text table. Each row in the table represents one Oracle Application Server process.

You can customize the status command in the following ways:

- Supply a scope to obtain status of processes running on other Oracle Application Server instances
- Change the information displayed about each Oracle Application Server process
- Remove the table headers from the output
- Change the field separator
- Change the record separator
- Change the width of individual columns
- Change the justification of the data in an individual column

Enter the `opmnctl usage status` command to obtain full details on how to use the status command.

[Example 1–4](#) shows the output after entering the `opmnctl status` command for the AppSrv1 instance on host `comp1` for the domain `yourcompany.com`:

**Example 1–4 opmnctl Status Output**

```
prompt > opmnctl status
```

```
Processes in Instance: AppSrv1.comp1.yourcompany.com
```

ias-component	process-type	pid	status
WebCache	WebCacheAdmin	29121	Alive
WebCache	WebCache	29120	Alive
OC4J	OC4J_Demos	N/A	Down
OC4J	home	29268	Init
dcm-daemon	dcm-daemon	29113	Alive
LogLoader	logloader	N/A	Down
HTTP_Server	HTTP_Server	29099	Alive

You can use the `opmnctl status` command with `<scope>` to obtain additional detailed information. For example, the following command gives you the status of every process of every component of every OracleAS Instance in an entire farm.:

```
prompt > opmnctl @farm status
```

**See Also:** [Section 1.8.3.1, "Scope"](#)

**1.8.4.1.1 Options for the Status Command of opmnctl** The following are the options you can specify for the <options> parameter:

- **-l**: Use this option to obtain the uniqueid (uid) value.

For example, the following command outputs the information shown in [Example 1-5](#):

```
prompt > opmnctl status -l
```

**Example 1-5 opmnctl status -l output**

```
Processes in Instance: j2eeuser.yourcompany.com
-----+-----+-----+-----+-----+-----+-----+-----
ias-component|process-type|pid   |status |uid       |memused|uptime   | ports
-----+-----+-----+-----+-----+-----+-----+-----
-OC4J        | home       | 5611 | Alive | 632225812|105008 |17:55:58 |jms:3701,
rmi: 3201,ajp:3000
```

The uid information enables you to stop or restart an individual Oracle Application Server process.

For example, the following command stops the home process-type:

```
prompt > opmnctl stopproc uniqueid=632225812
```

- **-fsep <string>**: Use this option to assign a field separator value for your opmnctl status output. The default value is |.
- **-rsep <string>**: Use this option to assign a record separator value for your opmnctl status output. The default value is \n.
- **-noheaders**: Use this option if you do not want a header displayed after you run the opmnctl status command.
- **-fmt <fmtlist>**: This is a single string containing one or more statistic identifiers connected together where each identifier has the following format: <statname>[<width>{<justification>}]. The default value is: %cmp18%prt18%pid5R%sta8.

[Table 1-3](#) lists the format string syntax for the <fmtlist> option:

**Table 1–3 Format String Syntax**

Format String Syntax	Description
<statname>	<p>This must be one of the following:</p> <ul style="list-style-type: none"> <li>■ clu: Oracle Application Server cluster name</li> <li>■ ins: Oracle Application Server instance name</li> <li>■ cmp: Oracle Application Server component ID</li> <li>■ prt: process-type ID</li> <li>■ prs: process-set ID</li> <li>■ idx: index of process in process-set</li> <li>■ pid: operating system process ID</li> <li>■ uid: OPMN uniqueid</li> <li>■ typ: name for this kind of process</li> <li>■ sta: process status</li> <li>■ stm: start time (ms)</li> <li>■ utm: up time (ms)</li> <li>■ cpu: cpu time (ms)</li> <li>■ mem: memory used (in KB)</li> <li>■ por: port list</li> </ul>
<width>	<p>Specifies the size for the field. Output shorter than this value receives padding according to the specified &lt;justification&gt;. Output longer than this value is truncated, and terminated with '~'.</p> <p>Default: width of each datum.</p>
<justification>	<p>Specifies the justification for the field. This enables you to justify output when it is less than the width. It is L, R, or C (left, right, or center justification).</p> <p>Default: L</p>

For example, the following command displays the output shown in [Example 1–6](#):

```
prompt> opmnctl status -noheaders -fsep @ -fmt %cmp%prt%pid%sta
```

**Example 1–6 opmnctl status -noheaders output**

```
OC4J@home@N/A@Down
dcm-daemon@dcm-daemon@13875@Alive
LogLoader@logloaderd@9800@Alive
HTTP_Server@HTTP_Server@13926@Alive
```

**See Also:** [Section 1.8.3, "Command Definitions"](#)

Enter the following command for additional detailed information:

```
prompt > opmnctl usage status
```

### 1.8.4.2 opmnctl dmsdump

Syntax: `opmnctl [<scope>] dmsdump`  
`[<attr>=<value> [&<attr>=<val>...]]`

The `opmnctl dmsdump` command enables you to print the Oracle Dynamic Monitoring Service (DMS) statistics for OPMN. You can obtain a printout of process control operations for specific Oracle Application Server components. If no attributes are specified, performance data for all OPMN-managed processes for your Oracle Application Server components are printed out.

DMS enables you to monitor a specific performance metric, a set of performance metrics, or all performance metrics. Options allow you to specify a reporting interval to report the requested metrics. For more information about DMS performance metric attributes and values refer to the *Oracle Application Server Performance Guide*.

The `<scope>` defines where the print request will be routed. The usage of `<scope>` with the `opmnctl dmsdump` command has the following characteristics:

- If none or only `@instance` is specified, the print request will be routed to the local Oracle Application Server instance only. If `@instance` is followed by other Oracle Application Server instance names, the request will be routed to the specified names.
- If `@cluster` is specified, the request will be routed to the local Oracle Application Server cluster. If `@cluster` is followed by names, the request will be routed to the Oracle Application Server cluster with the specified names. If `@cluster` is followed by other Oracle Application Server cluster names, the request will be routed to the specified names.
- If `@farm` is specified, the request will be routed to the whole farm.

Multiple `<attr>=<value>` pairs must be separated by an `&`. The following command:

```
prompt > opmnctl dmsdump "table=opmn_ons&format=xml"
```

will output the set of statistics that are gathered for ONS, including the ports that it listens on and the number of notifications it has processed. The output is in `.xml` format rather than text. If you want to review the output in text format do not include `&format=xml` on the command line.

For more information about DMS refer to the *Oracle Application Server Performance Guide*.

### 1.8.4.3 opmnctl ping

Syntax: `opmnctl ping [<max_retry>]`

The `opmnctl ping` command enables you to contact the local OPMN server to verify operation. `<max_retry>` specifies the maximum number of retry times. If `<max_retry>` is specified, the local OPMN is pinged every one second, until the command execution succeeds or `<max_retry>` is reached.

For example, the following command,

```
prompt > opmnctl ping 10
```

designates pinging of OPMN 10 times until the ping command succeeds

## 1.8.5 Help Commands

The `opmnctl help` commands enable you to obtain additional information regarding OPMN.

This section describes the help command options available with the `opmnctl` command. It includes the following sections:

- [Section 1.8.5.1, "opmnctl help"](#)
- [Section 1.8.5.2, "opmnctl usage"](#)
- [Section 1.8.5.3, "opmnctl validate"](#)

### 1.8.5.1 opmnctl help

Syntax: `opmnctl help`

Use this command to print a short syntax description of `opmnctl` commands.

[Example 1–7](#) shows the output from the `opmnctl help` command.

#### **Example 1–7 opmnctl help Output**

```
prompt > opmnctl help
```

```
usage: /ORACLE_HOME/bin/opmnctl [verbose] [<scope>] <command> [<options>]
```

```
verbose: print detailed execution message if available
```

```
Permitted <scope>/<command>/<options> combinations are:
```

scope	command	options
	start	- Start opmn
	startall	- Start opmn & all managed processes
	stopall	- Stop opmn & all managed processes
	shutdown	- Shutdown opmn & all managed processes
[<scope>]	startproc	[<attr>=<val>..] - Start opmn managed processes
[<scope>]	restartproc	[<attr>=<val>..] - Restart opmn managed processes
[<scope>]	stopproc	[<attr>=<val>..] - Stop opmn managed processes
[<scope>]	reload	- Trigger opmn to reread opmn.xml
[<scope>]	status	[<options>] - Get managed process status
[<scope>]	dmsdump	[<attr>=<val>&..] - Get DMS stats
	ping	[<max_retry>] - Ping local opmn
	validate	[<filename>] - Validate the given xml file
	help	- Print brief usage description
	usage	[<command>] - Print detailed usage description

### 1.8.5.2 opmnctl usage

Syntax: `opmnctl usage [<command>]`

The `usage` command displays help for all `opmnctl` commands, or only for the specified command.

The command can be one or more of the following:

- start
- startall
- startproc
- stopall
- stopproc
- restartproc
- reload
- shutdown

- ping
- status
- dmsdump
- help

For example, enter the following command to receive the output shown in [Example 1-8](#):

```
prompt > opmnctl usage stopall
```

#### **Example 1-8 opmnctl usage stopall output**

```
opmnctl stopall
```

Stop opmn daemon and opmn managed processes for local ias instance.

This request first tries to stop all opmn managed processes gracefully. Processes which will not stop gracefully will be forcefully shutdown. After stopping all managed processes, the opmn daemon will shutdown itself.

This request should only be performed when it is necessary to stop the opmn daemon. Once started, the opmn daemon should remain up until it is necessary to restart the computer or some other rare administrative event occurs.

To stop all opmn managed processes without stopping the opmn daemon, consider using the stopproc command without any arguments.

To restart the opmn daemon without restarting any managed processes, consider using the reload command. The reload command is the appropriate command to use when the only goal is to restart the opmn daemon with a new configuration.

This request operates synchronously and will wait for the operation to complete before returning.

#### **1.8.5.3 opmnctl validate**

Syntax: `opmnctl validate [<filename>]`

The `opmnctl validate` command validates the XML syntax of the `opmn.xml` file. The default `ORACLE_HOME/opmn/conf/opmn.xml` is validated if the filename parameter is not specified. The `<filename>` can be specified by either the relative or absolute path.

Only one file can be validated at a time. No output is generated for successful `opmn.xml` file validation.

## **1.9 Event Scripts**

You can configure OPMN to execute your own custom event scripts whenever a particular component starts, stops, or crashes. You can select from one or more of the following event types:

- **pre-start:** OPMN runs the pre-start script after any configured dependency checks have been performed and passed, and before the Oracle Application Server component starts. For example, the pre-start script can be used for site-specific initialization of external components.

- **pre-stop:** OPMN runs the pre-stop script before stopping a designated Oracle Application Server component. For example, the pre-stop script can be used for collecting Java Virtual Machine stack traces prior to stopping OC4J processes.
- **post-crash:** OPMN runs the post-crash script after the Oracle Application Server component has terminated unexpectedly. For example, a user could learn of component crashes by supplying a script or program to be executed at post-crash events which sends a notification to the administrator's pager.

**See Also:**

- [Section A.2.5, "Troubleshooting with Event Scripts"](#)
- [<event-scripts> in Chapter 3, "opmn.xml Common Configuration"](#)

## 1.10 Start Order Dependencies

Some Oracle Application Server components and services require that other components and services are up and running before starting. OPMN is configured at installation with default start order dependencies, which enables you to start all of the components in an instance in the proper order with a single command. Refer to the *Oracle Application Server Administrator's Guide* for more information on Oracle Application Server dependencies.

OPMN is configured with a set of dependencies but you can configure additional dependencies according to the environment

## 1.11 OPMN Log Files

The log files generated by OPMN provide important information that can help you identify and diagnose performance and configuration issues. The Application Server Control Console makes reviewing these log files easier by helping you locate and view Oracle Application Server component log files.

**See Also:**

- [Section A.2.1, "OPMN log Files"](#)
- *Oracle Application Server Administrator's Guide*

## 1.12 Security

The OPMN local listener port used by ONS clients and PM administrative processes do not use Secure Socket Layer (SSL) encryption for security, but rely on two other mechanisms to ensure authorized access to the OPMN server:

- OPMN binds the local listener port to the local host. Users on the local system can connect to this port and issue OPMN process control requests. Information requests are allowed on the OPMN request port, which is bound to the system IP. The request port does not have SSL encryption.
- When the OPMN server process first starts up and successfully binds to the local port, it creates a string of printable ASCII characters which it uses as a key for local connections. All connection attempts on the local port must include this key or the connection is closed by the OPMN server. The ASCII character string is written into the `ORACLE_HOME/opmn/conf/.formfactor` file. Processes that cannot access the `.formfactor` file are not permitted to interact with the OPMN server.



For security reasons, the OPMN server logs any attempts to connect to its local port with an invalid form factor key (a key that does not match the value written by this OPMN process into the `.formfactor` file).

In addition to attempted security violations, there are four common user errors that can cause this error to occur:

- The user attempts to run the OPMN client manually with the wrong user identification. Only the application server user can read the value from the `.formfactor` file, and so requests or processes run as the wrong user will not be able to provide the correct key to the OPMN server.
- The user is attempting to run an OPMN client from the wrong `ORACLE_HOME`. It is possible to have multiple `ORACLE_HOME` instances set up on the same system. If the other `ORACLE_HOME` instances have OPMN configured to use the same local port then the Oracle Application Server process request from the wrong `ORACLE_HOME` will read the wrong `.formfactor` file.
- The user has manually changed the local port configuration in the `opmn.xml` file and started a new OPMN server without first stopping the previous OPMN server. The new OPMN server will run, bind to the new port, and overwrite the `.formfactor` file. The previous OPMN server is now unreachable through the local port, and can only be shutdown through remote OPMN requests (if SSL and authentication are configured) or by manually stopping the previous OPMN server.
- The Oracle Application Server and the Oracle Database both use ONS. When these two products are installed onto the same host, an ONS port conflict arises since the default port values (local="6100" remote="6200") for ONS are the same for both the Oracle Database and Oracle Application Server.

ONS with the Oracle Database is only used for special configurations and therefore is typically never started. However, the database listener will attempt to connect to the Database ONS server but will end up connecting to the ONS server that was installed with Oracle Application Server. ONS (as part of OPMN) is always started whenever Oracle Application Server is started.

Because the Oracle Database is installed in a different `ORACLE_HOME` than that of Oracle Application Server, the Database ONS does not have access to the `.formfactor` file that was created when OPMN started up with the Oracle Application Server. As a result, the database listener attempts to connect to OPMN; the DB listener interprets it as a its standalone ONS) without a form factor string. Oracle Application Server OPMN logs an error similar to the following in the `ons.log` file:

```
04/11/15 18:43:32 [4] Local connection 0,127.0.0.1,6100 invalid form factor
```

This is expected OPMN behavior Oracle Application Server; preventing client access to the ONS server unless they possess the correct formfactor string.

To avoid having the Oracle Database listener contact the Oracle Application Server OPMN server, change the default local and remote port values for the ONS server that was installed with the Oracle Database. Alternatively, you can apply the latest Oracle Database patchset available on OTN:

<http://www.oracle.com/technology/products/>

### 1.12.1 Remote Security

OPMN supports remote requests to other OPMN servers in the same farm, but for security reasons all process control requests (start, restart and stop) are only enabled if SSL is enabled in the `opmn.xml` file and a wallet file is configured. If neither SSL nor a wallet file are configured, OPMN will reject any remote process control request with HTTP code 403.

The remote port used for remote administration must be SSL-enabled. The remote port should only be used for communication between multiple OPMN servers. Oracle Application Server components and Application Server Control Console transmit through the local port which is inaccessible to remote administration. All access control and authentication is controlled by going through Application Server Control Console.

**See Also:** *Oracle Application Server Security Guide*

## 1.13 DCM

DCM manages the configuration of `opmn.xml` and manages configurations among application server instances that are associated with a common Infrastructure (members of an Oracle Application Server farm). It enables Oracle Application Server cluster-wide deployment so you can deploy an application to an entire cluster, or make a single host or instance configuration change applicable across all instances in a cluster. The Application Server Control Console uses DCM to make configuration changes and to propagate configuration changes and deployed applications across the cluster.

All command line process control should be through the OPMN `opmnctl` command. The DCM `dcmctl` command should be used only for configuration related operations and application deployment.

**See Also:** *Distributed Configuration Management Administrator's Guide*

## 1.14 iHAT

The Oracle Application Server Hi-Av Tool (iHAT) provides a real time, graphical interface view of your enterprise. iHAT displays all Oracle Application Server processes managed by one or more OPMN servers including useful performance metrics about each process. The snapshot of the system is updated continuously at a configurable interval.

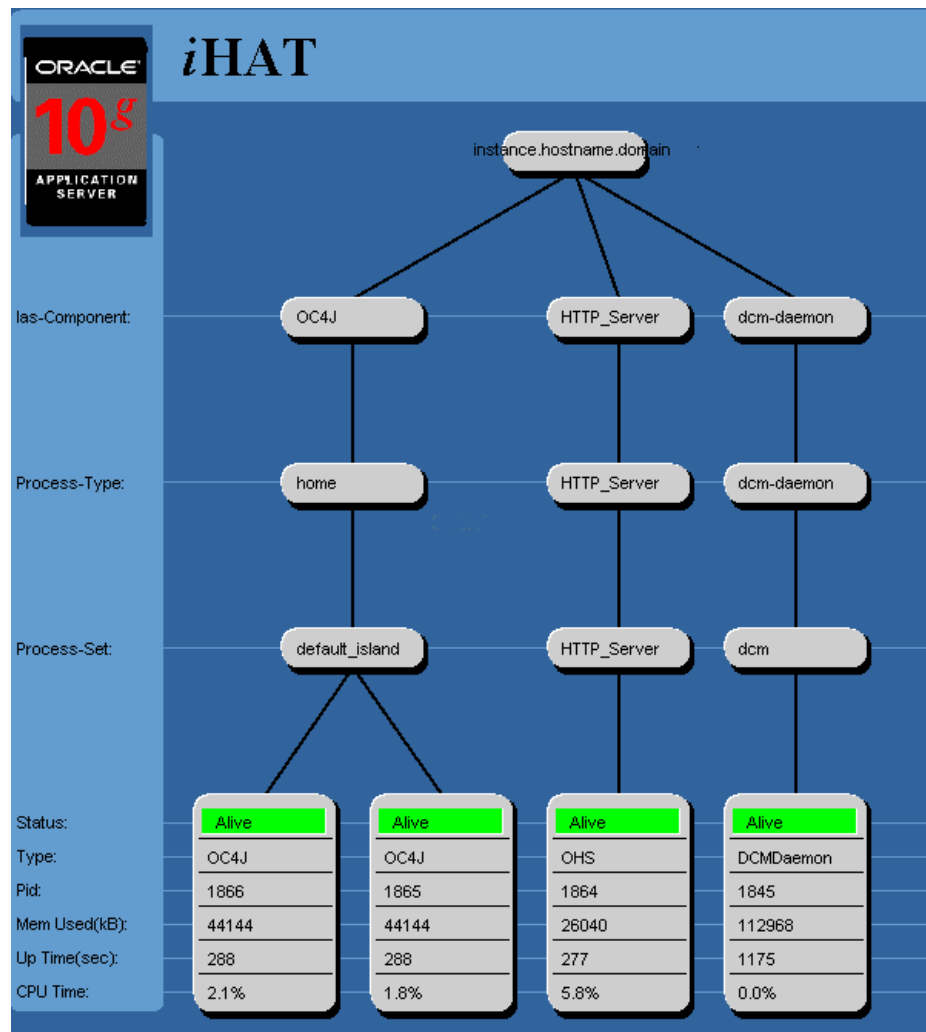
iHAT shows your Oracle Application Server instance with Oracle Application Server components. The `ias-component`, `process-set`, and `process-type` levels are displayed for each Oracle Application Server component. The child process display of the `process-type` shows the status, `process-type`, process identification (pid), memory usage, uptime, and CPU usage.

Figure 1–2 shows an example of the iHAT display for an enterprise with two Oracle Application Server instances.

To download iHAT, visit the OTN:

<http://www.oracle.com/technology/membership>

Figure 1-2 Example iHAT Display





---

## Using OPMN

This chapter provides command-line examples on how to use OPMN for Oracle Application Server. It features the following topics:

- [Section 2.1, "Starting OPMN"](#)
- [Section 2.2, "Starting and Stopping OPMN-Managed Processes for a Local Oracle Application Server Instance"](#)
- [Section 2.3, "Starting and Stopping all OPMN Managed Processes for a Remote Oracle Application Server Instance"](#)
- [Section 2.4, "Starting and Stopping an Oracle Application Server Component in a Local Oracle Application Server Instance"](#)
- [Section 2.5, "Starting and Stopping an Oracle Application Server Process Type in a Local Oracle Application Server Instance"](#)
- [Section 2.6, "Starting and Stopping a Multi-Oracle Application Server Instance Environment"](#)
- [Section 2.7, "Starting a Component on an Oracle Application Server Farm"](#)

### 2.1 Starting OPMN

OPMN does not depend on any other Oracle Application Server component being up and running before it can be started and used. The OPMN server should be started as soon as possible after turning on the host.

Use the following command to start OPMN without starting other Oracle Application Server components:

```
prompt > opmnctl start
```

### 2.2 Starting and Stopping OPMN-Managed Processes for a Local Oracle Application Server Instance

Use the following command to **start** OPMN-managed processes for a local Oracle Application Server instance:

```
prompt > opmnctl startproc
```

Use the following command to **stop** OPMN-managed processes for a local Oracle Application Server instance:

```
prompt > opmnctl stopproc
```

---

---

**Note:** Without arguments the `opmnctl startproc` and `opmnctl stopproc` commands start and stop all OPMN-managed processes.

---

---

## 2.3 Starting and Stopping all OPMN Managed Processes for a Remote Oracle Application Server Instance

Use the following command to **start** OPMN managed processes for a remote Oracle Application Server instance:

```
prompt > opmnctl @instance:oracleas2.foo.com startproc
```

Use the following command to **stop** OPMN managed processes for a remote Oracle Application Server instance:

```
prompt > opmnctl @instance:oracleas2.foo.com stopproc
```

## 2.4 Starting and Stopping an Oracle Application Server Component in a Local Oracle Application Server Instance

Use the following command to **start** Oracle Internet Directory in a local Oracle Application Server instance:

```
prompt > opmnctl startproc ias-component=OID
```

Use the following command to **stop** Oracle Internet Directory in a local Oracle Application Server instance:

```
prompt > opmnctl stopproc ias-component=OID
```

## 2.5 Starting and Stopping an Oracle Application Server Process Type in a Local Oracle Application Server Instance

Use the following command to **start** the `performance_server` process-type in a local Oracle Application Server instance:

```
prompt > opmnctl startproc ias-component=wireless process-type=performance_server
```

Use the following command to **stop** the `performance_server` process-type in a local Oracle Application Server instance:

```
prompt > opmnctl stopproc ias-component=wireless process-type=performance_server
```

## 2.6 Starting and Stopping a Multi-Oracle Application Server Instance Environment

Use the following command to **start** a multi-Oracle Application Server instance environment from local instance `oracleas1`:

```
prompt > opmnctl @instance:oracleas1.foo.com:oracleas2.bar.com startproc
```

This command starts all processes of all components on both instances specified with the <scope> argument. Notice that the local instance `oracleas1` is specified in the command.

Use the following command to **stop** a multi-Oracle Application Server instance environment from local instance `oracleas1`:

```
prompt > opmnctl @instance:oracleas1.foo.com:oracleas2.bar.com stopproc
```

This command stops all processes of all components on all two instances specified with the <scope> argument. Notice that the local instance `oracleas1` is specified in the command.

---

---

**Note:** You must also indicate your local Oracle Application Server instance when using the `@instance` <scope> with other Oracle Application Server instances. Commands with a <scope> argument only operates on the instances described by the <scope> argument. The command will only be applied to the local instance if it is described in the <scope> argument.

---

---

## 2.7 Starting a Component on an Oracle Application Server Farm

Use the following command to **start** the same Oracle Application Server component on multiple Oracle Application Server instances:

```
prompt > opmnctl @farm startproc ias-component=HTTP_Server
```

Use the following command to **stop** the same Oracle Application Server component on multiple Oracle Application Server instances:

```
prompt > opmnctl @farm stopproc ias-component=HTTP_Server
```





## opmn.xml Common Configuration

This chapter provides common configuration examples, and descriptions of elements and attributes for the OPMN `opmn.xml` file.

It contains the following topics:

- [Section 3.1, "Example of opmn.xml Elements and Attributes"](#)
- [Section 3.2, "opmn.xml Element and Attribute Descriptions"](#)

### 3.1 Example of opmn.xml Elements and Attributes

[Example 3–1](#) shows all possible elements and attributes that may appear in an `opmn.xml` file that are not specific to any Oracle Application Server component.

---

**Note:** OPMN will convert slashes in the path value string to be those of the directory path separator character for the system on which OPMN is running (for UNIX each `\` character is converted to `/`; for Windows each `/` is converted to `\`).

OPMN uses the `^` character as an escape character to disable slash conversion. `^/` on a Windows system will yield a `/` in the string. Specify two `^` characters if you need to specify the `^` character in the resultant string. For example, `^^` yields `^`.

---

#### *Example 3–1 Common Configuration Elements and Attributes*

```
<opmn>
  <notification-server>
    <ipaddr remote="ip" request="ip"/>
    <port local="port" remote="port" request="port"/>
    <log-file path="path" level="level" rotation-size="kBytes" rotation-hour="HOD"/>
    <ssl enabled="boolean" wallet-file="path" wallet-password="password"/>
  </notification-server>
  <process-manager insecure-remote-requests="boolean">
    <log-file path="path" level="level" rotation-size="kBytes" rotation-hour="HOD"/>
  </process-manager>
  <process-modules>
    <module path="path" tag="tag-id" status="state" cron="interval">
      <module-data>
        <category id="id">
          <data id="id" value="value" process-conversion="boolean"/>
        </category>
      </module-data>
    </module>
  </process-modules>
</opmn>
```

```
</process-modules>
  <ias-instance id="ias-instance-name" ORACLE_HOME="path">
    <environment>
      <variable id="id" value="value" append="boolean" process-conversion="boolean"/>
    </environment>
    <!-- module-data -->
    <ias-component id="component-id" id-matching="boolean" status="state">
      <!-- environment -->
      <!-- module-data -->
      <dependencies>
        <database db-connect-info="connect" infrastructure-key="key"
          timeout="depend-timeout" cache-timeout="cache-timeout"/>
        <OID address="address" timeout="depend-timeout" cache-timeout="cache-timeout">
          <ssl enabled="boolean" wallet-file="path" wallet-password="password">
        </OID>
        <OSSO host="hostname" port="port" URI="uri" timeout="depend-timeout"
          cache-timeout="cache-timeout">
          <ssl enabled="boolean" wallet-file="path" wallet-password="password">
        </OSSO>
        <managed-process ias-instance="ias-instance-id" ias-component="ias-component-id"
          process-type="process-type-id" process-set="process-set-id" autostart="boolean"
          autostop="boolean" timeout="depend-timeout" cache-timeout="cache-timeout"/>
      </dependencies>
      <process-type id="process-type-id" module-id="module-id" status="state"
        working-dir="path">
        <!--environment -->
        <!-- module-data -->
        <!-- dependencies -->
        <event-scripts>
          <pre-start path="path">
          <pre-stop path="path">
          <post-crash path="path">
        </event-scripts>
        <start timeout="timeout" retry="num"/>
        <stop timeout="timeout"/>
        <restart timeout="timeout" retry="num"/>
        <ping timeout="timeout" retry="num" interval="interval"/>
        <port id="id" range="range"/>
        <process-set id="process-set-id" restart-on-death="boolean" numprocs="num"
          minprocs="min" maxprocs="max" status="state" working-dir="path" parallel-requests=
            "boolean">
          <!-- environment -->
          <!-- module-data -->
          <!-- dependencies -->
          <!-- event script -->
          <!-- start -->
          <!-- stop -->
          <!-- restart -->
          <!-- ping -->
          <!-- port -->
        </process-set>
      </process-type>
    </ias-component>
  </ias-instance>
</process-manager>
</opmn>
```

## 3.2 opmn.xml Element and Attribute Descriptions

This section describes the elements and attributes in the `opmn.xml` file that are not specific to any Oracle Application Server component. This section also provides attribute descriptions of the elements.

### <opmn>

Required: true  
 Default: none  
 Parents: none  
 Attributes: none

Top-level element for `opmn.xml`

### <notification-server>

Required: true  
 Default: none  
 Parents: [<opmn>](#)  
 Attributes: none

Configures or, contains elements to configure the ONS portion of OPMN.

### <ipaddr>

Required: true  
 Default: none  
 Parents: [<notification-server>](#)  
 Attributes: `remote`, `request`

Specifies host information for ONS listener threads and host port bindings.

**`remote="ip"`**

Required: false  
 Default: none  
 Valid Values: IP address (in `###.###.###.###` format) or host name to which ONS will bind its remote port.

IP address or host name to which ONS will bind its remote port. The remote port is used for ONS to ONS communication. Notifications pass from ONS to ONS through the remote port, and OPMN uses ONS to route remote requests to other OPMNs through the remote port.

**`request="ip"`**

Required: false  
 Default: none  
 Valid Values: IP address (in `###.###.###.###` format) or host name to which ONS will bind its request port.

IP address or host name to which ONS will bind its remote port. This port can only be used for obtaining status information (DMS).

### <port>

Required: true  
 Default: none  
 Parents: [<notification-server>](#)

Attributes: `local`, `remote`, `request`

Configuration of the `port` information for ONS listener threads host and port bindings.

**`local="port"`**

Required: true

Default: none

Valid Values: A port number.

ONS local port value.

**`remote="port"`**

Required: false

Default: none

Valid Values: A port number.

ONS remote port value.

**`request="port"`**

Required: false

Default: none

Valid Values: A port number.

ONS request port value.

## **<log-file>**

Required: true

Default: none

Parents: [<notification-server>](#)

Attributes: `path`, `level`, `rotation-size`, `rotation-hour`

Configuration definitions for the ONS log mechanism.

**`path="path"`**

Required: true

Default: none

Valid Values: A file name path for the ONS log file.

The `path` attribute is a value string. All directories specified in the path must already exist, and OPMN must have read and write permissions for the directory in which the log file resides. `$ORACLE_HOME` may be used.

`ORACLE_HOME` is the root directory in which Oracle software is installed.

OPMN will convert slashes in the path value string to be those of the directory path separator character for the system on which OPMN is running (for UNIX each `\` character is converted to `/`; for Windows each `/` is converted to `\`).

OPMN uses the `^` character as an escape character to disable slash conversion. `^/` on a Windows system will yield a `/` in the string. Specify two `^` characters if you need to specify the `^` character in the resultant string. For example, `^^` yields `^`.

**`level="level"`**

Required: true

Default: none

Valid Values: An integer value of 0 through 9.

This value specifies the ONS logging level.

**0** No logging

**1** Fatal errors

**2** Errors

**3** Warnings

**4** Events worth noting

**5** Listener and resource management, `opmn.xml` parse information

**6** Connection management, internal notification routing, and worker thread management

**7** Notification parsing and processing, subscription addition and removal, and high level send queue checking

**8** Received notification content, detailed send queue processing, detailed subscription processing

**9** Sent notification content and detailed subscription processing

**`rotation-size="kBytes"`**

Required: false

Default: none

Valid Values: An integer.

The maximum size in kilobytes of the log file. When the log file reaches the configured size, the ONS logging mechanism will close the log file, rename it with an integer time stamp, and then create a new log file. This attribute may be used with `rotation-hour`.

**`rotation-hour="HOD"`**

Required: false

Default: none

Valid Values: An integer value of 0 through 23.

At the prescribed hour of the day, the ONS logging mechanism will close the log file, rename it with an integer time stamp, and then create a new log file. This attribute may be used with `rotation-size`.

## **<ssl>**

Required: false

Default: none

Parents: [<notification-server>](#)

Attributes: `enabled`, `wallet-file`, `wallet-password`

ONS to ONS security and authentication configuration.

**`enabled="boolean"`**

Required: true

Default: none

Valid Values: true or false

If the value is true, enables SSL for ONS.

**wallet-file="path"**

Required: false

Default: none

Valid Values: A path to an Oracle wallet.

Specify the Oracle wallet to use for authentication on ONS connections. `$ORACLE_HOME` may be used.

OPMN will convert slashes in the path value string to be those of the directory path separator character for the system on which OPMN is running (for UNIX each \ character is converted to /; for Windows each / is converted to \).

OPMN uses the ^ character as an escape character to disable slash conversion. ^/ on a Windows system will yield a / in the string. Specify two ^ characters if you need to specify the ^ character in the resultant string. For example, ^^ yields ^.

**wallet-password="password"**

Required: false

Default: none

Valid Values: A string for the wallet password.

The password string for the specified wallet.

### **<process-manager>**

Required: true

Default: none

Parents: [<opmn>](#)

Attributes: `insecure-remote-requests`

The `process-manager` contains the configuration definitions for the PM portion of OPMN.

**insecure-remote-requests="boolean"**

Required: false

Default: false

Valid Values: `true` or `false`

By default OPMN will only allow start, stop, restart, shutdown and reload requests rerouted from remote OPMNs if ONS SSL is enabled and a wallet file is configured for authentication.

---

---

**Note:** Setting this attribute to `true` overrides that security check and enables these requests to be issued remotely with no security features configured.

---

---

Setting this attribute to `true` is a major security risk and should only be done for testing purposes with all connected OPMNs behind a well secured fire wall or completely disconnected from any external network.

### **<log-file>**

Required: true

Default: none

Parents: [<process-manager>](#)

Attributes: path, level, rotation-size, rotation-hour

Configuration definitions for the PM log mechanism.

**path="path"**

Required: true

Default: none

Valid Values: A file name path for the PM log file.

All directories specified in the path must already exist, and OPMN must have read and write permissions for the directory in which the log file resides. *\$ORACLE\_HOME* may be used.

OPMN will convert slashes in the path value string to be those of the directory path separator character for the system on which OPMN is running (for UNIX each \ character is converted to /; for Windows each / is converted to \).

OPMN uses the ^ character as an escape character to disable slash conversion. ^/ on a Windows system will yield a / in the string. Specify two ^ characters if you need to specify the ^ character in the resultant string. For example, ^^ yields ^.

**level="level"**

Required: true

Default: none

Valid Values: An integer value of 0 through 9.

This value specifies the PM logging level.

**0** No logging

**1** Fatal errors

**2** Errors

**3** Warnings

**4** Events worth noting

**5** Internal and module initialization, process monitoring, and dependency check results

**6** Process resource management, http request and process job scheduling and execution, and dependency check details

**7** Http request parsing and resource management

**8** Http request response, and DMS statistics collection and requests

**9** Low level thread tracing

**rotation-size="kBytes"**

Required: false

Default: none

Valid Values: An integer.

The maximum size in kilobytes of the log file. When the log file reaches the configured size, the PM logging mechanism will close the log file, rename it with an integer time stamp, and then create a new log file. This attribute may be used with rotation-hour.

**rotation-hour="HOD"**

Required: false

Default: none

Valid Values: An integer value of 0 through 23.

At the prescribed hour of the day, the PM logging mechanism will close the log file, rename it with an integer time stamp, and then create a new log file. This attribute may be used with `rotation-size`.

### <process-modules>

Required: true

Default: none

Parents: [<process-manager>](#)

Attributes: none

The PM dynamically loads in a library for each specified process module. Each process module is designed to support a specific set of `process-type`, and is only required if those `process-type` are configured.

### <module>

Required: true

Default: none

Parents: [<process-modules>](#)

Attributes: `path`, `tag`, `status`, `cron`

A module is used to provide `process-type` specific support for the PM. Each module is implemented as a shared library which exports a set of standard functions and uses the PM process module API. A module must provide a list of the `process-types` it supports, and only one configured process module may list a specific `process-type`; no two modules can list the same `process-type`.

#### **path="path"**

Required: true

Default: none

Valid Values: A path for the module shared library.

The path must specify the shared library file, but if the library file has the standard system suffix (`.so` for UNIX and `.dll` for Windows), the suffix may be omitted and OPMN will automatically append it. `$ORACLE_HOME` may be used when specifying the path.

OPMN will convert slashes in the path value string to be those of the directory path separator character for the system on which OPMN is running (for UNIX each `\` character is converted to `/`; for Windows each `/` is converted to `\`).

OPMN uses the `^` character as an escape character to disable slash conversion. `^/` on a Windows system will yield a `/` in the string. Specify two `^` characters if you need to specify the `^` character in the resultant string. For example, `^^` yields `^`.

#### **tag="tag-id"**

Required: false

Default: The value specified by path.

Valid Values: A string uniquely identifying the module.

A module may report its tag value when logging errors to the PM log file or as part of the response to a request. While optional, it is a good idea to set this attribute to a meaningful value to help track any issues with process management.



**status="state"**

Required: false

Default: enabled

Valid Values: critical, enabled, or disabled

A module may be enabled, in which case PM loads in its shared library when it starts and calls the module's initialization functions, or disabled in which case the module entry is completely ignored. If the module process-types are configured in `opmn.xml` they must also be disabled. The critical state is the same as enabled, except that OPMN will terminate with a fatal error code if the module initialization fails.

**cron="interval"**

Required: false

Default: none

Valid Values: An integer.

Specify the interval in seconds between calls to the module's cron callback function. Configuring a cron interval for a module that does not support the cron callback is not allowed. Unless you have designed the module, you should neither add nor alter this attribute.

**<module-data>**

Required: false

Default: none

Parents: [<module>](#) , [<ias-instance>](#) , [<ias-component>](#) , [<dependencies>](#) , [<process-set>](#)

Attributes: none

The module-data blocks are used to define module specific name-value pairs that are meaningful only to a specific module. Each module-data block is organized into categories, which contain the name-value data pairs.

The module-data blocks can be defined for multiple elements within `opmn.xml`, and OPMN will create an aggregate module-data block at the process-set level that contains all values defined at or above it. If multiple definitions exist in this hierarchy with the same category id and data id, the value defined at the lowest level is used.

[Table 3-1](#) illustrates the module-data defined at each level in the hierarchy (with the highest level displayed at the top) and the resultant union at the process-set level of all of the module-data definitions:

**Table 3-1 module-data Hierarchy**

Module	Definition
ias-instance	<pre>&lt;category id="CatA"&gt;   &lt;data id= "DataAA" value="aaaa"/&gt; &lt;/category&gt;</pre>
ias-component	<pre>&lt;category id="CatA"&gt;   &lt;data id= "DataAB" value="abab"/&gt; &lt;/category&gt; &lt;category id="CatB"&gt;   &lt;data id= "DataBA" value="baba"/&gt; &lt;/category&gt;</pre>

**Table 3–1 (Cont.) module-data Hierarchy**

Module	Definition
module	<pre>&lt;category id="CatA"&gt;   &lt;data id= "DataAC" value="acac"/&gt; &lt;/category&gt;</pre>
process-type	<pre>&lt;category id="CatA"&gt;   &lt;data id= "DataAA" value="XXXX"/&gt; &lt;/category&gt;</pre>
process-set	<pre>&lt;category id="CatB"&gt;   &lt;data id= "DataBB" value="bbbb"/&gt; &lt;/category&gt;</pre>
RESULT	<pre>&lt;category id="CatA"&gt;   &lt;data id= "DataAA" value="XXXX"/&gt;   &lt;data id= "DataAB" value="abab"/&gt;   &lt;data id= "DataAC" value="acac"/&gt; &lt;/category&gt; &lt;category id="CatB"&gt;   &lt;data id= "DataBA" value="baba"/&gt;   &lt;data id= "DataBB" value="bbbb"/&gt; &lt;/category&gt;</pre>

**<category>**

Required: true

Default: none

Parents: [<module-data>](#)

Attributes: id

The `category` is an organizational level within a `module-data` block.

**id="id"**

Required: true

Default: none

Valid Values: A string.

This string identifies a data category. Each `category id` within a single `module-data` block must be unique, but multiple `module-data` blocks may contain the same `data category ids`, in which case the categories are considered to be related.

**<data>**

Required: true

Default: none

Parents: [<category>](#)

Attributes: id, value, process-conversion

A data name value definition within a `module-data category`.

**id="id"**

Required: true

Default: none

Valid Values: A string.

This string identifies a data element. Each `data id` within a single category must be unique, but multiple categories may contain the same data identifications. Data elements with the same identification as others, defined in different categories with the same identification are related.

**value="value"**

Required: true

Default: none

Valid Values: A string.

The value string associated with the data element `id`. Any environment variable defined anywhere within the scope of the `process-set` (any level at or above the `process-set`) in which the data value is defined (again, any level at or above the `process-set`) referenced within the value string as `$variable` or `%variable%` will be expanded to the variable value.

**process-conversion="boolean"**

Required: false

Default: true

Valid Values: true or false

By default OPMN converts slashes in the data value string to be those of the directory path separator character for the system on which OPMN is running (on UNIX each `'\'` character is converted to `'/'` and on Windows each `'/'` is converted to `'\'`). Set this attribute to `false` to disable conversion.

Note that if `process-conversion` is `true`, OPMN uses the `^` character as an escape character to disable process conversion for the following character, and so `^/` on a Windows system will yield a `/` in the string. Specify two `^` characters if you need to specify the `^` character in the resultant string: `^^` yields `^`.

### **<module-id>**

Required: true

Default: none

Parents: [<module>](#)

Attributes: `id`

The `module-id` name defines the type of process and associates the configuration with a process module.

This identifier is used by each `process-type` to specify which module supports it. A module may be configured with multiple `module-ids`.

**id="module-id"**

Required: true

Default: none

Valid Values: A string.

### **<ias-instance>**

Required: true

Default: none

Parents: [<process-manager>](#)

Attributes: `id`, `ORACLE_HOME`

The configuration definitions for an Oracle Application Server instance. Only one `ias-instance` is supported for each OPMN.

**id="ias-instance-name"**

Required: true

Default: none

Valid Values: A string.

This string should match the DCM configured `ias-instance` name in `dcm.conf`. OPMN itself ignores the value of this string and always uses the DCM configured value.

**ORACLE\_HOME="path"**

Required: true

Default: none

Valid Values: A path string.

This path must be the `$ORACLE_HOME` equivalent for this Oracle Application Server instance.

OPMN will convert slashes in the path value string to be those of the directory path separator character for the system on which OPMN is running (for UNIX each `\` character is converted to `/`; for Windows each `/` is converted to `\`).

OPMN uses the `^` character as an escape character to disable slash conversion. `^/` on a Windows system will yield a `/` in the string. Specify two `^` characters if you need to specify the `^` character in the resultant string. For example, `^^` yields `^`.

### **<environment>**

Required: true

Default: *Refer to the following paragraph.*

Parents: [<ias-instance>](#) , [<ias-component>](#) , [<dependencies>](#) , [<process-set>](#)

Attributes: none

Like `module-data` blocks, `environment` blocks can be defined for multiple elements within the `opmn.xml` file, and OPMN will create an aggregate `environment` block at the `process-set` level that contains all values defined at, or above it. If multiple definitions exist in this hierarchy with the same `id`, the value defined at the lowest level is used.

---

---

**Note:** OPMN sets the following default environment variables at the `ias-instance` level, with the values extracted either from the `ias-instance` configuration or from the OPMN run time environment:

UNIX: `ORACLE_HOME`, `ORACLE_NLS`, `OPMN_ENV_LC_ALL`,  
`OPMN_ENV_LANG`, `OPMN_ENV_NLS_LANG`, `SHELL`

Windows: `COMSPEC`, `SYSTEM_DRIVE`, and `SYSTEM_ROOT`

---

---

### **<variable>**

Required: true

Default: none

Parents: [<environment>](#)

Attributes: `id`, `value`, `append`, `process-conversion`

The environment variable name and value are defined with this element.

**id="name"**

Required: true  
 Default: none  
 Valid Values: A string.

The environment variable name. An `environment id` may be duplicated within an `environment block`, with the last defined value taking priority over earlier definitions. The same `environment id` may be defined within `environment blocks` for different elements, and the value defined at the lowest level will take priority over values defined at higher levels.

**value="value"**

Required: true  
 Default: none  
 Valid Values: A string.

The environment value. Environment variables referenced within the value string as `$variable` or `%variable%` will be expanded to the variable value. The same environment variable may reference itself to use a definition defined at a higher level, or earlier within this same `environment block`.

You may use the UNIX shell syntax for referencing an environment variable, `$variable` or `${variable}`, or the Windows format `%variable%`. Referenced variables that have not been defined remain in place as referenced, and so `value= "_notdefined_"` would remain unchanged if `notdefined` was not defined.

For example, the following environment block yields a value for `accumulate` of `"foobar"`.

```
<environment>
  <variable id="accumulate" value="foo">
  <variable id="accumulate" value="${accumulate}bar">
</environment>
```

**append="boolean"**

Required: false  
 Default: false  
 Valid Values: true or false

You can force OPMN to append the new environment variable value to the previously defined value, with the system library delimiter placed in between the two values (':' for UNIX and ';' for Windows) by specifying a value of `true` for this attribute. This is useful when assembling a value for a variable such as `CLASSPATH`.

For example, the following environment block yields a value for `CLASSPATH` of `"/foo:/bar"` on a UNIX system.

```
<environment>
  <variable id="CLASSPATH" value="/foo">
  <variable id="CLASSPATH" value="/bar" append="true">
</environment>
```

**process-conversion="boolean"**

Required: false  
 Default: true  
 Valid Values: true or false

OPMN converts slashes in the environment value string to be those of the directory path separator character for the system on which OPMN is running (for UNIX each \ character is converted to /; for Windows each / is converted to \).

If the process conversion is `true`, OPMN uses the `^` character as an escape character to disable slash conversion. `^/` on a Windows system will yield a `/` in the string. Specify two `^` characters if you need to specify the `^` character in the resultant string. For example, `^^` yields `^`.

### **<ias-component>**

Required: true

Default: none

Parents: [<ias-instance>](#)

Attributes: `id`, `id-matching`, `status`

An `ias-component` is a logical grouping of `process-type` for administrative purposes.

**`id="component-id"`**

Required: true

Default: none

Valid Values: a string

The `id` attribute uniquely identifies this `ias-component` within the `ias-instance`.

**`id-matching="boolean"`**

Required: false

Default: false

Valid Values: `true` or `false`

By default OPMN requests that do not specify `ias-components` match all configured `ias-components`, unless the `id-matching` attribute for a component is set to `true`, in which case the request must explicitly include the `ias-component id` in order to affect the `ias-component` or any `process-type` or `process-set` configured for that `ias-component`.

**`status="state"`**

Required: false

Default: `enabled`

Valid Values: `enabled` or `disabled`

An `ias-component` may be `enabled`, in which case OPMN parses all of its configured attributes and elements and enables requests to operate upon it, or `disabled`, in which case the `ias-component` entry is completely ignored.

### **<dependencies>**

Required: false

Default: none

Parents: [<ias-component>](#) , [<dependencies>](#) , [<process-set>](#)

Attributes: none

OPMN uses `dependencies` to determine if a process should be started or not. Like `module-data` and `environment` block, `dependencies` blocks can be defined for multiple elements within the `opmn.xml` file, and OPMN will create an aggregate dependency list at the `process-set` level that contains all dependencies defined at or

above it. If duplicate dependencies are defined at different levels, then duplicate checks on that dependency will be made before starting a process. OPMN will create an aggregate dependency list at the `process-set` level that contains all dependencies defined at or above it. If duplicate dependencies are defined at different levels, then duplicate checks on the dependency will be made before starting a process.

There are two primary types of dependencies: external and internal. External dependencies are for those resources not managed by OPMN. For example: OracleAS Metadata Repository and Application Server Control Console.

An external program is executed by OPMN to perform the check on the resource. Internal dependencies are for OPMN-managed processes (unit), which may include processes managed on a remote OPMN.

OPMN maintains a cache of dependency states which contains the last known state of each dependency, and the time it was last checked. A `cache-timeout` parameter for each dependency enables users to specify how long to use its state in the cache, or if it should be used at all. Similarly, a general timeout parameter for each dependency will determine how long OPMN will wait for a status update from that dependency before aborting the dependency check and the process start.

OPMN checks dependencies in the order in which they are declared. The traversal of this list of dependencies concludes either with the full sequence of successful checks, the dependency is available, or the first check failure, the dependency is not available, or the dependency check timed out.

## <database>

Required: false

Default: none

Parents: [<dependencies>](#)

Attributes: `db-connect-info`, `infrastructure-key`, `timeout`, `cache-timeout`

Specifies the database to check: either `db-connect-info` or `infrastructure-key` is used to identify the database.

**`db-connect-info="connect"`**

Required: true if `infrastructure-key` is not specified.

Default: none

Valid Values: A string

The string required to connect to the database.

**`infrastructure-key="key"`**

Required: true if `db-connect` is not specified.

Default: none

Valid Values: A string

The `infrastructure` key required to identify the database.

**`timeout="depend-timeout"`**

Required: false

Default: 1200

Valid Values: An integer

The `timeout` attribute specifies in seconds how long OPMN will wait for a dependency check to complete. If the check takes longer than the configured timeout, then OPMN will consider the check to have failed.

**`cache-timeout="cache-timeout"`**

Required: false

Default: 600

Valid Values: An integer

The `cache-timeout` attribute specifies how long in seconds OPMN will use the current "up" status for this dependency's entry in the cache. If the last successful dependency check was within the prescribed number of seconds from the current check, then the dependency check is instantly flagged as successful, otherwise another dependency check will be performed. Note that the `cache-timeout` is only for the last successful check of the dependency, and if the previous check failed, another access of the dependency will be performed for this check. A value of 0 indicates OPMN will always perform the check.

## <OID>

Required: false

Default: none

Parents: [<dependencies>](#)

Attributes: `address`, `infrastructure`, `timeout`, `cache-timeout`

Specifies the Oracle Internet Directory (OID) service to check either an address string for a specific Oracle Internet Directory, or that the OracleAS Infrastructure flag is set to `true` to use the default infrastructure Oracle Internet Directory.

**`address="address"`**

Required: true

Default: none

Valid Values: A string

The `address` string required to connect to Oracle Internet Directory.

**`infrastructure="boolean"`**

Required: true if `address` is not set.

Default: none

Valid Values: `true` or `false`

Use the default infrastructure Oracle Internet Directory for this Oracle Application Server instance.

**`timeout="depend-timeout"`**

Required: false

Default: 1200

Valid Values: An integer

The `timeout` attribute specifies in seconds how long OPMN will wait for a dependency check to complete. If the check takes longer than the configured timeout, then OPMN considers the check to have failed.

**`cache-timeout="cache-timeout"`**

Required: false

Default: 600



Valid Values: An integer

The `cache-timeout` attribute specifies how long in seconds OPMN will use the current "up" status for the dependency entry in the cache. If the last successful dependency check was within the prescribed number of seconds from the current check, then the dependency check is flagged as successful. Otherwise, OPMN performs another dependency check. The `cache-timeout` is only for the last successful check of the dependency. If the previous check failed, OPMN performs another access of the dependency check. A value of 0 indicates OPMN will always perform the check.

## <ssl>

Required: false

Default: none

Parents: <OID>

Attributes: `enabled`, `wallet-file`, `wallet-password`

The SSL information for the Oracle Internet Directory connection.

**`enabled="boolean"`**

Required: true

Default: none

Valid Values: true or false

To enable SSL on the Oracle Internet Directory connection, set this attribute to true.

**`wallet-file="path"`**

Required: false

Default: none

Valid Values: A path

The path to a wallet file for authentication of the Oracle Internet Directory connection. `$ORACLE_HOME` may be used.

OPMN will convert slashes in the `wallet-file` value string to be those of the directory path separator character for the system on which OPMN is running (on UNIX each \ character is converted to / and on Windows each / is converted to \).

OPMN uses the ^ character as an escape character to disable slash conversion for the following character, and so ^/ on a Windows system will yield a / in the string.

Specify two ^ characters if you need to specify the ^ character in the resultant string: ^^ yields ^.

**`wallet-password="password"`**

Required: false

Default: none

Valid Values: A string

The password for the specified wallet-file.

**`timeout="depend-timeout"`**

Required: false

Default: 1200

Valid Values: An integer

The `timeout` attribute specifies in seconds how long OPMN will wait for a dependency check to complete. If the check takes longer than the configured timeout, then OPMN considers the check to have failed.

**`cache-timeout="cache-timeout"`**

Required: false

Default: 600

Valid Values: An integer

The `cache-timeout` attribute specifies how long in seconds OPMN will use the current "up" status for the dependency entry in the cache. If the last successful dependency check was within the prescribed number of seconds from the current check, then the dependency check is flagged as successful. Otherwise, OPMN performs another dependency check. The `cache-timeout` is only for the last successful check of the dependency. If the previous check failed, OPMN performs another access of the dependency check. A value of 0 indicates OPMN will always perform the check.

## <OSSO>

Required: false

Default: none

Parents: [<dependencies>](#)

Attributes: `host`, `port`, `URI`, `timeout`, `cache-timeout`

Specifies the OracleAS Single Sign-On (OSSO) service to check.

**`host="hostname"`**

Required: true

Default: none

Valid Values: A string

The hostname for the OracleAS Single Sign-On connection.

**`port="port"`**

Required: true

Default: none

Valid Values: A port number

The port for the OracleAS Single Sign-On connection.

**`URI="uri"`**

Required: true

Default: none

Valid Values: A string

The URI for the OracleAS Single Sign-On connection.

## <ssl>

Required: false

Default: none

Parents: [<OSSO>](#)

Attributes: `enabled`, `wallet-file`, `wallet-password`

The SSL information for the OracleAS Single Sign-On connection.

**enabled="boolean"**

Required: true

Default: none

Valid Values: true or false

To enable SSL on the OracleAS Single Sign-On connection, set this attribute to `true`.

**wallet-file="path"**

Required: true

Default: none

Valid Values: A path

The path to a wallet file for authentication of the OracleAS Single Sign-On connection. `$ORACLE_HOME` may be used.

OPMN will convert slashes in the wallet-file value string to be those of the directory path separator character for the system on which OPMN is running (on UNIX each `\` character is converted to `/` and on Windows each `/` is converted to `\`).

OPMN uses the `^` character as an escape character to disable slash conversion for the following character, and so `^/` on a Windows system will yield a `/` in the string. Specify two `^` characters if you need to specify the `^` character in the resultant string: `^^` yields `^`.

**wallet-password="password"**

Required: false

Default: none

Valid Values: A string

The password for the specified wallet-file.

**timeout="depend-timeout"**

Required: false

Default: 1200

Valid Values: An integer

The `timeout` attribute specifies in seconds how long OPMN will wait for a dependency check to complete. If the check takes longer than the configured timeout, then OPMN considers the check to have failed.

**cache-timeout="cache-timeout"**

Required: false

Default: 600

Valid Values: An integer

The `cache-timeout` attribute specifies how long in seconds OPMN will use the current "up" status for the dependency entry in the cache. If the last successful dependency check was within the prescribed number of seconds from the current check, then the dependency check is flagged as successful. Otherwise, OPMN performs another dependency check. The `cache-timeout` is only for the last successful check of the dependency. If the previous check failed, OPMN performs another dependency check. A value of 0 indicates OPMN will always perform the check.

**<managed-process>**

Required: false

Default: none

Parents: [<dependencies>](#)Attributes: `ias-instance`, `ias-component`, `process-type`, `process-set`, `autostart`, `autostop`, `timeout`, `cache-timeout`

Specifies the managed process to check. A process for `process-type` or `process-set` does not start unless the specified dependency managed process is alive. Circular dependencies are detected and rejected for local managed processes, but not for remote managed processes; this may result in a dependency check deadlock, which times out.

**`ias-instance="ias-instance-id"`**

Required: false

Default: The `ias-instance` of the current `process-type` or `process-set`.

Valid Values: A string

The `ias-instance` for the managed process dependency. If the specified `ias-instance` is not managed by the current OPMN, it is assumed to be a remote managed process dependency.

**`ias-component="ias-component-id"`**

Required: true

Default: none

Valid Values: A string

The `ias-component` for the managed process dependency.

**`process-type="process-type-id"`**

Required: true

Default: none

Valid Values: A string

The `process-type-id` for the managed process dependency.

**`process-set="process-set-id"`**

Required: true

Default: none

Valid Values: A string

The `process-set-id` for the managed process dependency.

**`autostart="boolean"`**

Required: false

Default: false

Valid Values: true or false

If the managed process dependency is not running when the check is performed, attempt to start it.

**`autostop="boolean"`**

Required: false

Default: false

Valid Values: true or false

When the managed process dependency is stopped, then stop the managed process. The attribute is always `false` for remote managed process dependencies.

**`timeout="depend-timeout"`**

Required: false

Default: 1200

Valid Values: An integer

The `timeout` attribute specifies, in seconds, how long OPMN will wait for a dependency check to complete. If the check takes longer than the configured timeout, then OPMN considers the check to have failed.

**`cache-timeout="cache-timeout"`**

Required: false

Default: 600

Valid Values: An integer

This parameter is only used for a process managed by a remote OPMN. The `cache-timeout` attribute specifies how long in seconds OPMN will use the current "up" status for the dependency entry in the cache. If the last timeout dependency check was within the prescribed number of seconds from the current check, then the dependency check is instantly flagged as successful, otherwise OPMN performs another dependency check. Note that the `cache-timeout` is only for the last successful check of the dependency, and if the previous check failed, OPMN another access of the dependency will be performed for this check. A value of 0 indicates OPMN will always perform the check.

---

**Note:** The `cache-timeout` is only for the last successful check of the dependency, and if the previous check failed, OPMN will perform another dependency check.

---

## **<process-type>**

Required: true

Default: none

Parents: [<ias-component>](#)

Attributes: `id`, `module-id`, `status`, `working-dir`

A `process-type` is a grouping of `process-sets` that are supported by the same module.

**`id="process-type-id"`**

Required: true

Default: none

Valid Values: a string

The `id` attribute uniquely identifies this `process-type` within the `ias-component`.

**`module-id="module-id"`**

Required: true

Default: none

Valid Values: a string

The `module-id` attribute must map directly to the `module-id` element that supports this `process-type`.

**`status="state"`**

Required: false

Default: enabled

Valid Values: enabled or disabled

A `process-type` may be enabled, in which case OPMN parses all of its configured attributes and elements and enables requests to operate upon it, or disabled, in which case the `process-type` entry is completely ignored and treated as if it were not listed in `opmn.xml`.

**`working-dir="path"`**

Required: false

Default: None

Valid Values: A path

This path specifies the working directory set for managed processes created that belong to this `process-type`. If a `process-set` also defines a `working-dir` attribute, then that path takes precedence over the `process-type` path. `$ORACLE_HOME` may be used.

OPMN will convert slashes in the `working-dir` value string to be those of the directory path separator character for the system on which OPMN is running (on UNIX each `\` character is converted to `/` and on Windows each `/` is converted to `\`).

OPMN uses the `^` character as an escape character to disable slash conversion for the following character, and so `^/` on a Windows system will yield a `/` in the string. Specify two `^` characters if you need to specify the `^` character in the resultant string: `^^` yields `^`.

## <event-scripts>

Required: false

Default: none

Parents: `<process-type>`, `<process-set>`

Attributes: none

A configured `event script` is executed when a specific process related event has occurred. OPMN waits until the script completes or times out before proceeding with the next action for the process.

Table 3–2 shows event script arguments.

**Table 3–2 Event Script Arguments**

Option Name	Option Argument	Description
<code>-timeStamp</code>	<code>&lt;time&gt;</code>	An integer value for the current time on the system (in seconds).
<code>-instanceName</code>	<code>&lt;instance-name&gt;</code>	The <code>instance-name</code> of the managed process.
<code>-componentId</code>	<code>&lt;component-id&gt;</code>	The <code>component-id</code> of the managed process.
<code>-processType</code>	<code>&lt;process-type-id&gt;</code>	The <code>process-type</code> of the managed process.
<code>-processSet</code>	<code>&lt;process-set-id&gt;</code>	The <code>process-set</code> of the managed process.
<code>-processIndex</code>	<code>&lt;index&gt;</code>	The <code>process-index</code> of the managed process.

**Table 3–2 (Cont.) Event Script Arguments**

Option Name	Option Argument	Description
-stderr <sup>1</sup>	<path>	The path for the <code>stderr</code> file pointer of the process.
-stdout <sup>1</sup>	<path>	The path for the <code>stdout</code> file pointer of the process. Note: this argument will only be given for a pre-start script if the start is part of a process restart request.
-reason	<reason>	A string indicating the reason script was executed. The <code>http_request</code> indicates the process action is the result of the user http request to OPMN. The <code>non_http_request</code> indicates the process action was initiated by OPMN itself.
-pid <sup>2</sup>	<process-id>	The operating system integer value given for the <code>process-id</code> .
-startTime <sup>2</sup>	<time>	An integer value for the system start time of the process (in seconds).

<sup>1</sup> This argument will only be given for a pre-start script if the start is part of a process restart request. The pre-start event is triggered only prior to performing a start. A restart operation may be composed of a stop operation followed by a start operation. A start operation can occur as an operation all by itself or as a sub-operation of a restart.

<sup>2</sup> This argument is only available with pre-stop or post-crash event scripts.

**<pre-start>**

Required: false

Default: none

Parents: [<event-scripts>](#)

Attributes: path

OPMN runs the `pre-start` script after any configured dependency checks have been performed (and passed) and before the process is actually started. The timeout for this script is the timeout value configured for starting the process itself, and any time consumed by the execution of this script counts towards the process start timeout. If the script times out, the process will not be started and any associated http request will fail.

Be cautious when you execute any OPMN process requests such as `start`, `stop` or `restart` within an event script. These requests are serialized at the `process-set` level. If the script invokes a request on a `process-set` on which the current request (or another already queued request) is operating, then the script will hang until it times out.

**path="path"**

Required: true

Default: none

Valid Values: A path to the executable script.

The path must specify either an executable program for which OPMN has execute permission, or a script file for which OPMN has both read and executable permission. `$ORACLE_HOME` may be used.

OPMN will convert slashes in the path value string to be those of the directory path separator character for the system on which OPMN is running (on UNIX each `\` character is converted to `/` and on Windows each `/` is converted to `\`).

OPMN uses the ^ character as an escape character to disable slash conversion for the following character, and so ^/ on a Windows system will yield a / in the string. Specify two ^ characters if you need to specify the ^ character in the resultant string: ^^ yields ^.

#### <pre-stop>

Required: false

Default: none

Parents: [<event-scripts>](#)

Attributes: path

OPMN runs the specified script before stopping the associated process. The timeout for this script is the value configured for stopping the process itself. Any time consumed by the execution of this script counts towards the process stop timeout. If the script times out, any associated http request will fail. However, OPMN will proceed with stopping the process.

Be cautious when you execute any OPMN process requests such as start, stop or restart. These requests are serialized at the `process-set` level. If the script invokes a request on a `process-set` on which the current request (or another already queued request) is operating, then the script will hang until it times out.

**path="path"**

Required: true

Default: none

Valid Values: A path to the executable script.

The path must specify either an executable program for which OPMN has execute permission, or a script file for which OPMN has both read and executable permission. `$ORACLE_HOME` may be used.

OPMN will convert slashes in the path value string to be those of the directory path separator character for the system on which OPMN is running (on UNIX each \ character is converted to / and on Windows each / is converted to \).

OPMN uses the ^ character as an escape character to disable slash conversion for the following character, and so ^/ on a Windows system will yield a / in the string. Specify two ^ characters if you need to specify the ^ character in the resultant string: ^^ yields ^.

#### <post-crash>

Required: false

Default: none

Parents: [<event-scripts>](#)

Attributes: path

OPMN runs the specified script after the associated process has terminated unexpectedly. The timeout for this script is the timeout value configured for stopping the process itself. After the script has terminated OPMN schedules a replacement of the terminated process.

Be cautious when you execute any OPMN process requests such as start, stop or restart. These requests are serialized at the `process-set` level. If the script invokes a request on a `process-set` on which the current request (or another already queued request) is operating, then the script will hang until it times out.

**path="path"**



Required: true  
 Default: none  
 Valid Values: A path to the executable script.

The path must specify either an executable program for which OPMN has execute permission, or a script file for which OPMN has both read and executable permission. `$ORACLE_HOME` may be used.

OPMN will convert slashes in the path value string to be those of the directory path separator character for the system on which OPMN is running (on UNIX each `\` character is converted to `/` and on Windows each `/` is converted to `\`).

OPMN uses the `^` character as an escape character to disable slash conversion for the following character, and so `^/` on a Windows system will yield a `/` in the string. Specify two `^` characters if you need to specify the `^` character in the resultant string: `^^` yields `^`.

### <start>

Required: false  
 Default: Refer to the values in the following paragraphs.  
 Parents: [<process-type>](#) , [<process-set>](#)  
 Attributes: `timeout` , `retry`

The start parameters for managed processes.

**`timeout="timeout"`**

Required: false  
 Default: 120  
 Valid Values: An integer

The timeout value in seconds for the start of a managed process.

**`retry="num"`**

Required: false  
 Default: 0  
 Valid Values: An integer

The number of consecutive attempts that will be made to start the process for a single request.

### <stop>

Required: false  
 Default: Refer to the values in the following paragraphs.  
 Parents: [<process-type>](#) , [<process-set>](#)  
 Attributes: `timeout`

The stop parameters for managed processes.

**`timeout="timeout"`**

Required: false  
 Default: 60  
 Valid Values: An integer

The `timeout` value in seconds for the stopping a managed process.

**<restart>**

Required: false

Default: Refer to the values in the following paragraphs.

Parents: [<process-type>](#) , [<process-set>](#)

Attributes: `timeout`, `retry`

The `restart` parameters for managed processes.

**`timeout="timeout"`**

Required: false

Default: 180

Valid Values: An integer

The `timeout` value in seconds for the restart of a managed process.

**`retry="num"`**

Required: false

Default: 0

Valid Values: An integer

The number of consecutive attempts that will be made to restart the process for a single request.

**<ping>**

Required: false

Default: Refer to the values in the following paragraphs.

Parents: [<process-type>](#) , [<process-set>](#)

Attributes: `timeout`, `retry`, `interval`

The `ping` parameters for managed processes.

**`timeout="timeout"`**

Required: false

Default: 20

Valid Values: An integer

The `timeout` value in seconds for the ping of a managed process. Each module specifies a ping timeout.

**`retry="num"`**

Required: false

Default: 0

Valid Values: An integer

The number of consecutive ping failures that will be tolerated before the module declares the process unreachable and will restart it. Each module specifies ping retries.

**`interval="interval"`**

Required: false

Default: 20

Valid Values: An integer

The `interval`, in seconds, between each ping of a managed process.

**<port>**

Required: false  
 Default: none  
 Parents: [<process-type>](#) , [<process-set>](#)  
 Attributes: `id`, `range`

OPMN provides a port management mechanism for modules to use. Each module uses the ports configured with `id`.

**`id="id"`**

Required: true  
 Default: none  
 Valid Values: A string

The `id` attribute identifies the range of ports for the `process-type`. Each module has its own list of required or optional `port id`s.

**`range="range"`**

Required: true  
 Default: none  
 Valid Values: A port range

The `port range` specifies which ports to use for the `id`.

Upon request from a module for a port number from the `id`, OPMN checks if a port in the range has been bound on the local system, and if it has not, it returns that port number back to the module. Syntax of the `port range` is a comma separated list of individual port numbers or a low-high range specification.

Examples:

Specify ports 5555, 6666, 7777, 8888, and 9999:

`range="5555,6666,7777,8888,9999"`

Specify ports 4000 through 4250 (inclusive):

`range="4000-4250"`

Specify ports 7000 through 7049, 7775, 7785, and 8050 through 8099:

`range="7000-7049,7775,7785,8050-8099"`

**<process-set>**

Required: true  
 Default: none  
 Parents: [<process-type>](#)  
 Attributes: `id`, `restart-on-death`, `numprocs`, `status`, `working-dir`, `parallel-requests`

A `process-set` is the abstraction of a process within OPMN. All `module-data`, environment variables, and other configuration parameters are resolved into their final values at the `process-set` level.

**`id="process-set-id"`**

Required: true  
 Default: none

Valid Values: A string

The `id` attribute uniquely identifies this `process-set` within the `process-type`.

**`restart-on-death="boolean"`**

Required: false

Default: true

Valid Values: true or false

If a managed process terminates unexpectedly, that is, not stopped by a request, then OPMN will not automatically restart it. To enable automatic restarting of terminated managed processes set the attribute to true.

**`numprocs="num"`**

Required: true

Default: none

Valid Values: An integer

Specifies the number of processes for OPMN to start for this `process-set`.

**`minprocs="min"`**

Required: true unless `numprocs` is configured

Default: none

Valid Values: An integer

Specifies the default number of processes for OPMN to start for this process set. If `minprocs` is configured, then `maxprocs` must be set with a value greater than or equal to the value for `minprocs`. If `minprocs` and `maxprocs` are configured, a specific number of processes may be given in an OPMN request for this process set. This attribute may not be specified if `numprocs` has been configured.

**`maxprocs="max"`**

Required: true if `minprocs` is configured; otherwise false

Default: none

Valid Values: An integer

The `maxprocs` attribute must be specified if `minprocs` has been configured, but cannot be specified if `numprocs` has been configured.

**`status="state"`**

Required: false.

Default: enabled

Valid Values: enabled or disabled

A `process-set` may be enabled, in which case OPMN parses all of its configured attributes and elements and enables requests to operate upon it, or disabled, in which case the `process-set` entry is complete ignored and treated as if it were not even listed in `opmn.xml`.

**`working-dir="path"`**

Required: false.

Default: None

Valid Values: A path

This path specifies the working directory set for the managed processes created that belong to this `process-set`. `$ORACLE_HOME` may be used.

OPMN will convert slashes in the `working-dir` value string to be those of the directory path separator character for the system on which OPMN is running (on UNIX each `\` character is converted to `/` and on Windows each `/` is converted to `\`).

OPMN uses the `^` character as an escape character to disable slash conversion for the following character, and so `^/` on a Windows system will yield a `/` in the string. Specify two `^` characters if you need to specify the `^` character in the resultant string: `^^` yields `^`.

**`parallel-requests="boolean"`**

Required: false

Default: false

Valid Values: true or false

OPMN serializes requests at the process-set level, such that only one request can execute on a given process-set at a time: each subsequent request must wait until the previous request completes before it can execute. This default behavior is disabled for a process-set when `parallel-requests` is set to true.

---

**Note:** When this attribute is enabled OPMN performs **no** serialization on requests for the `process-set` at all, which means conflicting requests may be issued and execute at virtually the same time, thus leaving processes in the process-set in unpredictable states; therefore when `parallel-requests` is set to true you must verify that conflicting requests are not issued at the same time for the process-set (this includes requests with implicit wild-cards for matching process-sets).

---



---

# Configuring Oracle HTTP Server

This chapter describes Oracle HTTP Server configuration in the OPMN `opmn.xml` file. It features the following topics:

- [Section 4.1, "Oracle HTTP Server Process Module Configuration"](#)
- [Section 4.2, "Oracle HTTP Server Minimum Configuration"](#)
- [Section 4.3, "Oracle HTTP Server Complete Configuration"](#)
- [Section 4.4, "Oracle HTTP Server Attribute Descriptions"](#)
- [Section 4.5, "Oracle HTTP Server 2"](#)
- [Section 4.6, "Generic Apache \(UNIX only\)"](#)

---

**Note:**

- Only one Oracle HTTP Server can be configured for each `ORACLE_HOME`.
  - Oracle Application Server 10g Release 2 (10.1.2) comes with Oracle HTTP Server (a version of Apache 1.3). This version of Oracle Application Server cannot be configured to run Oracle HTTP Server 2 (a version of Apache 2).
  - To obtain Oracle HTTP Server 2 you must use the Oracle HTTP Server 2 standalone install. The Oracle HTTP Server 2 standalone installation is available on the companion CD. The Oracle HTTP Server 2 installation cannot be configured to run Oracle HTTP Server.
- 

## 4.1 Oracle HTTP Server Process Module Configuration

The following lines load and identify the Oracle HTTP Server process module. Management of Oracle HTTP Server processes by the process module are identified by the `module id`.

```
<module path="$ORACLE_HOME/opmn/lib/libopmnohs.so">  
  <module-id="OHS"/>  
</module>
```

## 4.2 Oracle HTTP Server Minimum Configuration

The following lines represent the minimum configuration for Oracle HTTP Server. Default values are assigned to all other configuration elements and attributes for Oracle HTTP Server.

```
<ias-component id="HTTP_Server">
  <process-type id="HTTP_Server" module-id="OHS">
    <process-set id="HTTP_Server" numprocs="1"/>
  </process-type>
</ias-component>
```

## 4.3 Oracle HTTP Server Complete Configuration

The following lines show a complete configuration for Oracle HTTP Server. It contains all possible configuration elements and attributes for Oracle HTTP Server.

```
<ias-component id="HTTP_Server" status="enabled" id-matching="false">
  <process-type id="HTTP_Server" module-id="OHS">
    <process-set id="HTTP_Server" restart-on-death="true" numprocs=1>
      <module-data>
        <category id="start-parameters">
          <data id="config-file" value="/myconfs/httpd.conf"/>
          <data id="start-mode" value="ssl-disabled"/>
          <data id="command-line" value="-D MyDefine"/>
        </category>
        <category id="ping-parameters">
          <data id="ping-url" value="/" />
        </category>
        <category id="restart-parameters">
          <data id="reverseping-timeout" value="345"/>
          <data id="no-reverseping-failed-ping-limit" value="3"/>
          <data id="reverseping-failed-ping-limit" value="6"/>
        </category>
      </module-data>
      <start timeout="300" retry="3"/>
      <stop timeout="300"/>
      <restart timeout="300"/>
      <ping timeout="30" interval="30"/>
    </process-set>
  </process-type>
</ias-component>
```

## 4.4 Oracle HTTP Server Attribute Descriptions

This section describes the attributes that are specific for Oracle HTTP Server.

The Oracle HTTP Server attributes are described with the following format:

- **Title:** This is the attribute name and value being defined. For example, `id="HTTP_Server"`.
- **Required:** This field defines whether or not the attribute is required in the component definition.
- **Default:** This defines the default value assigned to the attribute. The default value appears in the installed version of the `opmn.xml` file or is assigned internally if the attribute is not present.
- **Valid values:** If applicable, this field defines the valid values for the attribute. For example, `HTTP_Server`.



- **Path:** This field defines in which elements the attribute can appear. For example, `ias-component/process-type/process-set`

**id="HTTP\_Server"**

Required: true  
 Default: none  
 Valid values: HTTP\_Server  
 Path: `ias-component`  
 Path: `ias-component/process-type`  
 Path: `ias-component/process-type/process-set`

The `id` attribute is required and cannot be changed. The `id` must match the `targets.xml` entry or Application Server Control Console will not work.

**module-id="OHS"**

Required: true  
 Default: none  
 Valid values: OHS  
 Path: `ias-component/process-type`

The `module-id` attribute defines the type of process. It associates the configuration with a process module. The OHS `module-id` can be configured for managing Apache 1.3.

**numprocs=1**

Required: true  
 Default: none  
 Valid values: 1  
 Path: `ias-component/process-type/process-set`

The number of Oracle HTTP Server Instances to start. Only valid value is 1.

The `numprocs` attribute gives the number of Oracle HTTP Server instances to start. The only valid value is 1.

**id="start-parameters"**

Required: false  
 Default: none  
 Path: `ias-component/process-type/process-set/module-data/category`

The `start-parameters` category contains the parameters that are relevant for the startup of Oracle HTTP Server.

**id="config-file"**

Required: false  
 Default: `$ORACLE_HOME/Apache/Apache/conf/httpd.conf`  
 Valid values: any full path to an existing configuration file  
 Path: `ias-component/process-type/process-set/module-data/category/data`

The `config-file` `id` is an start command option which specifies the `httpd.conf` for starting Oracle HTTP Server. The `config-file` `id` is not supported on Microsoft Windows

**id="start-mode"**

Required: false  
Default: ssl-enabled  
Valid values: ssl-enabled/ssl-disabled  
Path: ias-component/process-type/process-set/module-data/  
category/data

This option specifies whether Oracle HTTP Server will be started with `ssl enabled`.

**id="command-line"**

Required: false  
Default: none  
Valid values: any valid command line options to Oracle HTTP Server  
Path: ias-component/process-type/process-set/module-data/  
category/data

This `id` option specifies extra command lines to append to the Oracle HTTP Server command line.

**id="ping-parameters"**

Required: false  
Default: none  
Path: ias-component/process-type/process-set/module-data/  
category  
The `ping parameters` category contains the parameters that configure how OPMN pings Oracle HTTP Server.

**id="ping-url"**

Required: false  
Default: /  
Valid values: the path portion of an url; for example: `http://127.0.0.1/<path>`  
Path: ias-component/process-type/process-set/module-data/  
category/data

The `ping-url` `id` specifies the URL at which OPMN pings Oracle HTTP Server.

**id="restart-parameters"**

Required: false  
Default: none  
Path: ias-component/process-type/process-set/module-data/  
category

The `restart-parameters` category is used for defining parameters that will be used in death-detection.

**id="reverseping-timeout"**

Required: false  
Default: 300 seconds  
Valid values: Any reasonable timeout value  
Path: ias-component/process-type/process-set/module-data/  
category/data

The `reverseping-timeout` value is the maximum allowable time between two notifications arriving from an Oracle HTTP Server process. As part of death-detection, the Oracle HTTP Server module performs forward pings on the Oracle HTTP Server process. In the event that forward pings start failing, the reverse pings are taken into account in death-detection and Oracle HTTP Server processes are restarted.

#### **id="no-reverseping-failed-ping-limit"**

Required: false

Default: 1

Valid values: Any reasonable value that reflects the tolerance that OPMN should have for failed forward pings when reverse pings are also failing. This tolerance is used by OPMN to determine when the process should be declared as unresponsive and replaced.

Path:

`ias-component/process-type/process-set/module-data/category/data`

This `id module data` element defines the tolerance for failed forward pings in the event that reverse pings are also not being received (within the timeout period specified by the `reverseping-timeout` data element). After the number of ping failures equals this limit, the process is deemed unresponsive and restarted by OPMN.

#### **id="reverseping-failed-ping-limit"**

Required: false

Default: 3

Valid values: Any reasonable value that reflects the tolerance that OPMN should have for failed forward pings when reverse pings are being received. This tolerance is used by OPMN to determine when the process should be declared as unresponsive and replaced.

Path:

`ias-component/process-type/process-set/module-data/category/data`

This `module data` element defines the tolerance for failed forward pings when reverse pings are succeeding. After the number of ping failures equals this limit, the process is deemed unresponsive and restarted by OPMN.

## **4.5 Oracle HTTP Server 2**

The Oracle HTTP Server 2 (OHS2) module ID is used to manage the version of Apache 2 shipped with Oracle Application Server 10g on the companion CD.

#### **module-id="OHS2"**

Required: true

Default: none

Valid values: OHS2

Path: `ias-component/process-type`

The `module-id` attribute defines the type of process. It associates the configuration with a process module.

#### **id="mpm"**

Required: false

Default: prefork

Valid values: worker/prefork

Path: ias-component/process-type/process-set/module-data/  
category/data

This option specifies what threading model OHS2 should use. This option is only valid for OHS2 and is only valid on Unix and Linux platforms.

Oracle does not support the perchild MPM. Microsoft Windows uses the mpm\_winnt MPM.

## 4.6 Generic Apache (UNIX only)

The Oracle HTTP Server process module can be configured to manage generic Apache processes. Follow these steps to configure the Oracle HTTP Server process module to manage generic Apache:

- Update the module definition to include the generic Apache module-id. The module definition should look similar to:

```
<module path="$ORACLE_HOME/opmn/lib/libopmnohs.so">
  <module-id id="OHS" />
  <module-id id="GENERIC_APACHE" />
</module>
```

- Alter your HTTP\_Server component to manage generic Apache. A GENERIC\_APACHE process has one required parameter specified as module-data. It is "apache-home" in the "start-parameters" module-data category. The "apache-home" specifies the installation directory of the generic Apache. The following example shows a generic Apache configuration:

```
<ias-component id="HTTP_Server">
  <process-type id="HTTP_Server" module-id="GENERIC_APACHE">
    <module-data>
      <category id="start-parameters">
        <data id="apache-home" value="/private1/apbuild/runapache_1.3.27"/>
      </category>
    </module-data>
    <process-set id="HTTP_Server" numprocs="1"/>
  </process-type>
</ias-component>
```

## Configuring OC4J

This chapter describes OC4J configuration in the OPMN `opmn.xml` file.

It features the following topics:

- [Section 5.1, "OC4J Process Module Configuration"](#)
- [Section 5.2, "OC4J Minimum Configuration"](#)
- [Section 5.3, "OC4J Complete Configuration"](#)
- [Section 5.4, "OC4J Attribute Descriptions"](#)

### 5.1 OC4J Process Module Configuration

The following lines load and identify the OC4J process module. Management of OC4J processes by the process module are identified by the `module id`.

```
<module path="$ORACLE_HOME/opmn/lib/libopmnoc4j.so">
  <module-id id="OC4J" />
</module>
```

### 5.2 OC4J Minimum Configuration

The following lines represent the minimum configuration for OC4J. Default values are assigned to all other configuration elements and attributes for OC4J.

```
<ias-component id="OC4J">
  <process-type id="home" module-id="OC4J">
    <port id="ajp" range="3301-3400" />
    <port id="rmi" range="3101-3200" />
    <port id="jms" range="3201-3300" />
    <process-set id="default-island" numprocs="1" />
  </process-type>
</ias-component>
```

### 5.3 OC4J Complete Configuration

The complete configuration example in this section showcases the attributes of OC4J configuration that you can control. It contains all possible configuration elements and attributes that can be used with this component.

```
<ias-component id="OC4J" status="enabled" id-matching="false">
  <environment>
    <variable id="LD_LIBRARY_PATH" value="$ORACLE_HOME/lib"
      append="true" />
  </environment>
```

```

<process-type id="home" module-id="OC4J">
  <port id="ajp" range="3301-3400" />
  <port id="rmi" range="3101-3200" />
  <port id="jms" range="3201-3300" />
<process-set id="default-island" restart-on-death="true" numprocs="1">
<module-data>
  <category id="start-parameters">
    <data id="java-options" value="-DTestVar=TestVal"/>
    <data id="oc4j-options" value=" "/>
    <data id="config-file" value="/my/config/dir/server.xml"/>
    <data id="java-bin" value="/my/javaloaction/jdk/bin/java"/>
  </category>
  <category id="stop-parameters">
    <data id="java-options" value="-DTestVar=TestVal"/>
  </category>
  <category id="restart-parameters">
    <data id="reverseping-timeout" value="345"/>
    <data id="no-reverseping-failed-ping-limit" value="3"/>
    <data id="reverseping-failed-ping-limit" value="6"/>
  </category>
  <category id="urlping-parameters">
    <data id="/j2ee/servlet/Spy" value="200"/>
  </category>
  <category id="security-parameters">
    <data id="wallet-file" value="file:/private/user/ssl_cert/client_cert"/>
    <data id="wallet-password" value="welcome1"/>
  </category>
</module-data>
  <start timeout="300" retry="3"/>
  <stop timeout="300"/>
  <restart timeout="300"/>
  <ping timeout="30" interval="30"/>
</process-set>
</process-type>
</ias-component>

```

## 5.4 OC4J Attribute Descriptions

This section describes the attributes that are specific for OC4J. This section also provides attribute descriptions of the attributes.

The OC4J attributes are described with the following format:

- **Title:** This is the attribute name and value being defined. For example, **id="OC4J"**.
- **Required:** This field defines whether or not the attribute is required in the component definition.
- **Default:** This defines the default value assigned to the attribute. The default value appears in the installed version of the `opmn.xml` file or is assigned internally if the attribute is not present.
- **Valid values:** If applicable, this field defines the valid values for the attribute. For example, OC4J.
- **Path:** This field defines in which elements the attribute can appear. For example, `ias-component`.

**id="OC4J"**

Required: true

Default: none  
Valid values: OC4J  
Path: `ias-component`

The `id` name is required and cannot be changed. The `id` must match the entry in the `targets.xml` file or Application Server Control Console will not work.

## **environment**

Required: false  
Default: none  
Path: `ias-component`

The `environment` element can be specified at multiple levels within the Oracle Application Server component. This concept is important for the OC4J module because OC4J processes can be part of other Oracle Application Server components. In those cases, the required environment may have to be specified at the `ias-component/process-type` level.

## **process-type**

Required: true  
Default: none  
Path: `ias-component`

For OC4J processes, the `process-type` element is administratively equivalent to an OC4J instance.

## **module-id="OC4J"**

Required: true  
Default: none  
Path: `ias-component/process-type`

The `module-id` associates the process with a module. For OC4J processes, this `id` has to match the `module-id` specified in the process module configuration for the OC4J module.

## **port**

Required: true  
Default: none  
Path: `ias-component/process-type`

The OC4J processes will not be started up unless port elements for `ajp` and `rmi` ports are configured. An OC4J process can be pinged at the `ajp` ports; therefore one of the ports has to be configured. If both type of ports are configured, then the `ajp` port is used for ping. In addition to the `ajp` and `rmi` ports, you can also configure other port types that will be passed in to the OC4J process at the command line during process startup.

## **id="ajp"**

Required: true  
Default: none  
Path: `ias-component/process-type/port`

A port element defining `ajp` port values is required.

**id="rmi"**

Required: true  
Default: none  
Path: `ias-component/process-type/port`

A port element defining rmi port values is required.

**id="jms"**

Required: true  
Default: none  
Path: `ias-component/process-type/port`

A port element defining jms port values is required.

**range**

Required: true  
Default: none  
Valid values: range of ports, individual port numbers or 0  
Path: `ias-component/process-type/port`

This attribute is used to specify valid port ranges, comma separated list of ports or a mix of both. For port selection by the operating system to select ports, specify 0 and the OC4J process will use a port provided by the system.

**<process-set>**

Required: true  
Default: none  
Path: `ias-component/process-type`

For OC4J processes, the `process-set` element is administratively equivalent to an OC4J island.

**id="start-parameters"**

Required: false  
Default: none  
Path: `ias-component/process-type/process-set/module-data/category`

The `start-parameters id` is a category that collects all of the parameters that are relevant for the startup of an OC4J process.

**id="java-options"**

Required: false  
Default: none  
Valid values: any options acceptable to Java  
Path: `ias-component/process-type/process-set/module-data/category/data`

OC4J requires that some `java-options` be passed to start and stop commands. These options are derived internally by OPMN, are not part of the `opmn.xml` configuration, and cannot be overridden. Additional `java-options` may be specified using this `module data` element.



**id="oc4j-options"**

Required: false

Default: none

Valid values: any options acceptable to the OC4J executable

Path: `ias-component/process-type/process-set/module-data/category/data`

OC4J processes require options to be passed in as part of the start or stop commands to function correctly. These options cannot be overridden. In addition to these options, other options can be passed in through this `module data` element. There is no default value for this data element.

**id="config-file"**

Required: false

Default: `$ORACLE_HOME/j2ee/<process-type id>/config/server.xml`

Valid values: any full path to an existing configuration file

Path: `ias-component/process-type/process-set/module-data/category/data`

The configuration file is an OC4J option in the start command. The default value for this data element is built from the `$ORACLE_HOME` variable and OC4J instance name (`process-type id`).

**id="java-bin"**

Required: false

Default: `$ORACLE_HOME/jdk/bin/java`

Valid values: Full path to `java.exe`

Path: `ias-component/process-type/process-set/module-data/category/data`

The default value is the complete path to Java that is available in the installation. You can specify alternate paths to the Java executable. However, a valid version of Java will have to be used for the process to start up and work correctly.

**id="stop-parameters"**

Required: false

Default: none

Path: `ias-component/process-type/process-set/module-data/category`

The `stop-parameters id` is a category that includes all the parameters that are relevant for stopping an OC4J process.

**id="restart-parameters"**

Required: false

Default: none

Path: `ias-component/process-type/process-set/module-data/category`

The `restart-parameters category` is used for defining parameters that will be used in death-detection.

**id="reverseping-timeout"**

Required: false

Default: 300 seconds

Valid values: Any reasonable timeout value

Path: `ias-component/process-type/process-set/module-data/category/data`

The `reverseping-timeout` value is the maximum allowable time between two notifications arriving from an OC4J process. As part of death-detection, the OC4J module performs forward pings on the process also. In the event that forward pings start failing, the reverse pings are taken into account in death-detection and restart.

**id="no-reverseping-failed-ping-limit"**

Required: false

Default: 1

Valid values: Any value that reflects the tolerance that OPMN should have for failed forward pings when reverse pings are also failing. This tolerance is used by OPMN to determine when the process should be declared as unresponsive and replaced.

Path: `ias-component/process-type/process-set/module-data/category/data`

This `module data` element defines the tolerance for failed forward pings in the event that reverse pings are also not being received (within the timeout period specified by `reverseping-timeout` data element). After the number of ping failures equals this limit, the process is deemed unresponsive and restarted by OPMN.

**id="reverseping-failed-ping-limit"**

Required: false

Default: 3

Valid values: Any reasonable value that reflects the tolerance that OPMN should have for failed forward pings when reverse pings are being received. This tolerance is used by OPMN to determine when the process should be declared as unresponsive and replaced.

Path: `ias-component/process-type/process-set/module-data/category/data`

This `module data` element defines the tolerance for failed forward pings when reverse pings are succeeding. After the number of ping failures equals this limit, the process is deemed unresponsive and restarted by OPMN.

**id="urlping-parameters"**

Required: false

Default: Not Applicable

Valid values: Not Applicable

Path: `ias-component/process-type/process-set/module-data/category`

The `"urlping-parameters"` id enables users to specify URLs for ping operations as part of OC4J process ping operations. The data under this category consists of the URL and a valid HTTP return code. AJP13 protocol is used to directly connect to the OC4J process and the HTTP return code is validated against the configured code. If there are multiple URLs configured, failure in pinging any one of them will be considered a ping failure and the process will be restarted after the ping failures limit is exceeded.

**id="/j2ee/servlet/Spy"**

Required: false  
 Default: Not Applicable  
 Valid values: Any valid URL on the OC4J process.  
 Path: ias-component/process-type/process-set/module-data/category/data

This is the URL in the OC4J process that will be pinged.

**value="200"**

Required: false  
 Default: Not Applicable  
 Valid values: Any valid HTTP return code.  
 Path: ias-component/process-type/process-set/module-data/category/data

The following is the HTTP code that results from ping operations to the configured URL.

```
<category id="security-parameters">
  <data id="wallet-file" value="file:/private/user/ssl_cert/client_cert"/>
  <data id="wallet-password" value="welcome1"/>
</category>
```

**id="security-parameters"**

Required: false  
 Default: Not Applicable  
 Valid values: Not Applicable  
 Path: ias-component/process-type/process-set/module-data/category/

The OC4J process module can perform pings over SSL. The "security-parameters" id is a category that enables users to specify the wallet file and password for such communication.

**id="wallet-file"**

Required: false  
 Default: Not Applicable  
 Valid values: Not Applicable  
 Path: ias-component/process-type/process-set/module-data/category/data

The data id whose value is the path to the wallet file (not including the filename).

**value="file:/private/user/ssl\_cert/client\_cert"**

Required: false  
 Default: Not applicable  
 Valid values: Path to a wallet file (not including the filename).  
 Path: ias-component/process-type/process-set/module-data/category/data

The path to the wallet file (not including the filename). The data in the wallet file is used in SSL authentication during ping.

**id="wallet-password"**

Required: false

Default: Not applicable

Valid values: Not applicable

Path: ias-component/process-type/process-set/module-data/  
category/data

The data id that specifies the wallet password.

**value="welcome1"**

Required: false

Default: Not applicable

Valid values: The valid wallet password.

Path: ias-component/process-type/process-set/module-data/  
category/data

This value specifies the password for the wallet.

# Configuring Oracle Application Server Web Cache

This chapter describes Oracle Application Server Web Cache (OracleAS Web Cache) configuration in the OPMN `opmn.xml` file.

It features the following topics:

- [Section 6.1, "OracleAS Web Cache Process Module Configuration"](#)
- [Section 6.2, "OracleAS Web Cache Minimum Configuration"](#)
- [Section 6.3, "OracleAS Web Cache Complete Configuration"](#)
- [Section 6.4, "OracleAS Web Cache Attribute Descriptions"](#)

## 6.1 OracleAS Web Cache Process Module Configuration

The following lines load and identify the OracleAS Web Cache process module. Management of OracleAS Web Cache processes by the process module are identified by the module id.

```
<module path="$ORACLE_HOME/opmn/lib/libopmnwc.so">
  <module-id id="WebCache" />
  <module-id id="WebCacheAdmin" />
</module>
```

## 6.2 OracleAS Web Cache Minimum Configuration

The following lines represent the minimum configuration for OracleAS Web Cache. Default values are assigned to all other configuration elements and attributes for OracleAS Web Cache.

```
<ias-component id="WebCache">
  <environment>
    <variable id="LD_LIBRARY_PATH" value="$ORACLE_HOME/lib:$ORACLE_HOME/opmn/lib" append="true"/>
  </environment>
  <process-type id="WebCache" module-id="WebCache">
    <process-set id="WebCache" numprocs="1"/>
  </process-type>
  <process-type id="WebCacheAdmin" module-id="WebCache-admin">
    <process-set id="WebCacheAdmin" restart-on-death="false" numprocs="1"/>
  </process-type>
</ias-component>
```

## 6.3 OracleAS Web Cache Complete Configuration

The following example represents the complete configuration for OracleAS Web Cache. It contains all possible configuration elements and attributes that can be used with OracleAS Web Cache.

```
<ias-component id="WebCache">
  <environment>
    <variable id="LD_LIBRARY_PATH" value="$ORACLE_HOME/lib:$ORACLE_HOME/opmn/lib" append="true"/>
  </environment>
  <process-type id="WebCache" module-id="WebCache">
    <process-set id="WebCache" restart-on-death="true" numprocs="1"/>
    <module-data>
      <category id=start-parameters>
        <data id="command-line" value="Option1 Option2"/>
      </category>
    </module-data>
  </process-set>
</process-type>
  <process-type id="WebCacheAdmin" module-id="WebCache"-admin">
    <process-set id="WebCacheAdmin" restart-on-death="false" numprocs="1"/>
    <module-data>
      <category id=start-parameters>
        <data id="command-line" value="Option1 Option2"/>
      </category>
    </module-data>
  </process-set>
</process-type>
</ias-component>
```

## 6.4 OracleAS Web Cache Attribute Descriptions

This section describes the attributes that are specific for OracleAS Web Cache.

The OracleAS Web Cache attributes are described with the following format:

- **Title:** This is the attribute name and value being defined. For example, `id="WebCache"`.
- **Required:** This field defines whether or not the attribute is required in the component definition.
- **Default:** This defines the default value assigned to the attribute. The default value appears in the installed version of the `opmn.xml` file or is assigned internally if the attribute is not present.
- **Valid values:** If applicable, this field defines the valid values for the attribute. For example, `WebCache`.
- **Path:** This field defines in which elements the attribute can appear. For example, `ias-component`.

### `id="WebCache"`

Required: true

Default: none

Valid values: `WebCache`

Path: `ias-component`

Path: `ias-component/process-type`

Path: `ias-component/process-type/process-set`

This `id` is required and cannot be changed. The `id` must match the `targets.xml` entry or Application Server Control Console will not work.

**id="WebCacheAdmin"**

Required: true  
Default: none  
Valid values: WebCacheAdmin  
Path: `ias-component/process-type`  
Path: `ias-component/process-type/process-set`

The `id` name is required and cannot be changed. The `id` must match the `targets.xml` entry or elements and attributes will not work.

**environment**

Required: true  
Default: none  
Path: `ias-component`

The environment segment must add `$ORACLE_HOME/lib` and `$ORACLE_HOME/opmn/lib` to the `LD_LIBRARY_PATH`.

**module-id="WebCache"**

Required: true  
Default: none  
Valid values: WebCache, WebCache-admin  
Path: `ias-component/process-type`

The `module id` defines whether the child `process-set` element defines a WebCache server process or a WebCache administrative process. At least one WebCache server process must be defined in the WebCache component.

**restart-on-death**

Required: true  
Default: true  
Valid values: true or false  
Path: `ias-component/process-type/process-set`

The `restart-on-death` attribute defines whether the process will be restarted if it exits unexpectedly. By default after Oracle Application Server installation, this attribute is set to `true` for the OracleAS Web Cache server, and `false` for the OracleAS Web Cache administrative process.

**numprocs="1"**

Required: true  
Default: none  
Valid values: 1  
Path: `ias-component/process-type/process-set`

This attribute gives the number of OracleAS Web Cache processes started for the `process-set`. The value must be set to 1.

**id=start-parameters**

Required: false

Default: none

Path: `ias-component/process-type/process-set/module-data/category`

The `start-parameters` category contains the parameters that are relevant for the startup of OracleAS Web Cache or OracleAS Web Cache Administrator.

**id ="command-line"**

Required: true

Default: none

Valid values: 1

Path: `ias-component/process-type/process-set`

The options specified for the value of `command-line` will be added to the `command-line` for OracleAS Web Cache or OracleAS Web Cache Administrator.



# Configuring Oracle Internet Directory

This chapter describes Oracle Internet Directory configuration in the OPMN `opmn.xml` file.

It features the following topics:

- [Section 7.1, "Oracle Internet Directory Process Module Configuration"](#)
- [Section 7.2, "Oracle Internet Directory Minimum Configuration"](#)
- [Section 7.3, "Oracle Internet Directory Complete Configuration"](#)
- [Section 7.4, "Oracle Internet Directory Attribute Descriptions"](#)

## 7.1 Oracle Internet Directory Process Module Configuration

The following lines load and identify the Oracle Internet Directory process module. Management of Oracle Internet Directory processes by the process module are identified by the `module id`.

```
<module path="$ORACLE_HOME/opmn/lib/libopmnoid.so">
  <module-id id="OID" />
</module>
```

## 7.2 Oracle Internet Directory Minimum Configuration

The following lines represent the minimum configuration for Oracle Internet Directory. Default values are assigned to all other configuration elements and attributes for Oracle Internet Directory.

```
<ias-component id="OID">
  <process-type id="OID" module-id="OID">
    <process-set id="OID" numprocs="1">
      <dependencies>
        <database db-connect-info="IASDB.US.ORACLE.COM" />
      </dependencies>
      <module-data>
        <category id="oidctl-parameters">
          <data id="connect" value="IASDB.US.ORACLE.COM" />
          <data id="startoidldapd" value="true" />
        </category>
        <category id="oidmon-parameters">
          <data id="connect" value="IASDB.US.ORACLE.COM" />
        </category>
      </module-data>
    </process-set>
  </process-type>
```

```
</ias-component>
```

## 7.3 Oracle Internet Directory Complete Configuration

The following example represents the complete configuration for Oracle Internet Directory. It contains all possible configuration elements and attributes that can be used with Oracle Internet Directory.

```
<ias-component id="OID">
  <process-type id="OID" module-id="OID" working-dir="my_working_dir">
    <environment>
      <variable id="TNS_ADMIN" value="/private/user/" />
    </environment>
    <process-set id="OID" numprocs="1">
      <dependencies>
        <database db-connect-info="IASDB.US.ORACLE.COM" />
      </dependencies>
      <module-data>
        <category id="oidmon-parameters">
          <data id="connect" value="IASDB.US.ORACLE.COM" />
          <data id="host" value="virtual_host_name" />
          <data id="sleep" value="20" />
          <data id="run-oidctl" value="true" />
        </category>
        <category id="oidctl-parameters">
          <data id="connect" value="IASDB.US.ORACLE.COM" />
          <data id="instance" value="1" />
          <data id="configset" value="1" />
          <data id="host" value="12.12.12.12" />
          <data id="flags" value="-p 389" />
          <data id="startoidldapd" value="true" />
          <data id="startoidrepld" value="true" />
          <data id="startodisrv" value="true" />
        </category>
      </module-data>
    </process-set>
  </process-type>
</ias-component>
```

## 7.4 Oracle Internet Directory Attribute Descriptions

This section describes the attributes that are specific for Oracle Internet Directory. This section also provides attribute descriptions of the attributes.

The Oracle Internet Directory attributes are described with the following format:

- **Title:** This is the attribute name and value being defined. For example, `id="OID"`.
- **Required:** This field defines whether or not the attribute is required in the component definition.
- **Default:** This defines the default value assigned to the attribute. The default value appears in the installed version of the `opmn.xml` file or is assigned internally if the attribute is not present.
- **Valid values:** If applicable, this field defines the valid values for the attribute. For example, `OID`.
- **Path:** This field defines in which elements the attribute can appear. For example, `ias-component`.

**id="OID"**

Required: true  
 Default: none  
 Valid values: OID  
 Path: `ias-component`  
 Path: `ias-component/process-type`  
 Path: `ias-component/process-type/process-set`

This `id` name is required and cannot be changed. The name must match the entry `targets.xml` or Application Server Control Console will not work.

**module-id="OID"**

Required: true  
 Default: none  
 Path: `ias-component/process-type`

The `module-id` name defines the type of process and associates this configuration with a process module.

**working-dir="my\_working\_dir"**

Required: false  
 Default: `$ORACLE_HOME`  
 Path: `ias-component/process-type`

The `working-dir` attribute specifies where the `oidmon` process will run.

**id="TNS\_ADMIN"**

Required: false  
 Default: `$ORACLE_HOME/network/admin`  
 Path: `ias-component/process-type/environment`

Oracle Internet Directory requires the connection to the database by specifying a connect string. The string taken by Oracle Internet Directory is an alias. At runtime, the string is resolved using the `tnsnames.ora` file in the `$TNS_ADMIN` directory. By default, `$TNS_ADMIN` points to `$ORACLE_HOME/network/admin`. If `tnsnames.ora` exists under a different directory, the environment must be set to the different directory.

**numprocs="1"**

Required: true  
 Default: none  
 Valid values: 1  
 Path: `ias-component/process-type/process-set`

This attribute gives the number of `oidmon` instances started for this `process-set`. There should be at most one instance running in one Oracle Application Server instance.

**db-connect-info="IASDB.US.ORACLE.COM"**

Required: true  
 Default: none  
 Valid values: The database connect string  
 Path: `ias-component/process-type/process-set`

Oracle Internet Directory depends on the backend database. If the backend database is not running, `oidmon` and other Oracle Internet Directory server processes cannot start.

**id="oidmon-parameters"**

Required: false

Default: none

Path: `ias-component/process-type/process-set/module-data/category`

The `id` category specifies the startup parameters for `oidmon`.

**id="connect"**

Required: true

Default: none

Valid values: The database connect string alias

Path: `ias-component/process-type/process-set/module-data/category/data`

The `id` data must be configured to start `oidmon`. The following command starts `oidmon`:

```
prompt> $ORACLE_HOME/bin/oidmon connect=<value> -opmnuid <uid> start
```

**id="host"**

Required: false

Default: none

Valid values: The virtual host name

Path: `ias-component/process-type/process-set/module-data/category/data`

The `host` data is configured by default at installation time. The following command starts `oidmon`:

```
prompt> $ORACLE_HOME/bin/oidmon connect=<value> host=<value> -opmnuid <uid> start
```

**id="sleep"**

Required: false

Default: 10 seconds

Valid values: The time interval in seconds at which `oidmon` monitors Oracle Internet Directory servers

Path: `ias-component/process-type/process-set/module-data/category/data`

If the `sleep` data is configured, the following command starts `oidmon`:

```
prompt> $ORACLE_HOME/bin/oidmon connect=<connect_value> sleep=<sleep_value> -opmnuid <uid> start
```

**id="run-oidctl"**

Required: false

Default: true

Valid values: true or false

Path: `ias-component/process-type/process-set/module-data/`

category/data

OPMN manages `oidmon`. However, when starting and stopping `oidmon`, OPMN also attempts to start `oidldapd` through `oidctl` by default.

If you do not want to start any process associated with `oidctl`, set this value to `false`.

#### **id="instance"**

Required: false

Default: 1

Valid values: The numerical value of the instance ( $>0$  and  $\leq 1000$ ) to be started and stopped.

Path: `ias-component/process-type/process-set/module-data/category/data`

The `instance` value is required for using `oidctl`. If this `id` is not configured, the default value is 1. You can overwrite this default option by specifying a value.

#### **id="configset"**

Required: false

Default: 0

Valid values: The numerical value of the configuration set ( $\geq 0$  and  $\leq 1000$ )

Path: `ias-component/process-type/process-set/module-data/category/data`

The `configset id` is only needed to start `oidldapd`. If you want to start `oidrepld` or `oidsrv`, the assigned data is ignored. If `configset` is not configured, the default value is 0.

#### **id="flags"**

Required: false

Default: none

Path: `ias-component/process-type/process-set/module-data/category/data`

If `flags` is configured, it will overwrite the settings in the `configset` at runtime.

#### **id="startoidldapd"**

Required: false

Default: true

Valid values: true or false

Path: `ias-component/process-type/process-set/module-data/category/data`

If the `id` data is configured, the default start command for the server is:

```
prompt> $ORACLE_HOME/bin/oidctl connect=<connect_value> server=oidldapd instance=1
configset=0 start
```

#### **id="startoidrepld"**

Required: false

Default: false

Valid values: true or false

Path: `ias-component/process-type/process-set/module-data/`

category/data

If the `id` data is configured, the default start command for the server is:

```
prompt> $ORACLE_HOME/bin/oidctl connect=<connect_value> server=oidrepld instance=1
configset=0 start
```

---

---

**Note:** This server must be configured before it can be brought up manually or by OPMN after installation.

---

---

#### **id="startodisrv"**

Required: false

Default: false

Valid values: true or false

Path: ias-component/process-type/process-set/module-data/  
category/data

If the `id` data is configured, the default start command for the server is:

```
prompt> $ORACLE_HOME/bin/oidctl connect=<connect_value> server=odisrv instance=1
configset=0 start
```

---

# Configuring Oracle Application Server Port Tunnel

This chapter describes Oracle Application Server Port Tunnel (OracleAS Port Tunnel) configuration in the OPMN `opmn.xml` file.

It features the following topics:

- [Section 8.1, "OracleAS Port Tunnel Process Module Configuration"](#)
- [Section 8.2, "OracleAS Port Tunnel Minimum Configuration"](#)
- [Section 8.3, "OracleAS Port Tunnel Complete Configuration"](#)
- [Section 8.4, "OracleAS Port Tunnel Attribute Descriptions"](#)

**See Also:** *Oracle HTTP Server Administrator's Guide*

## 8.1 OracleAS Port Tunnel Process Module Configuration

The following lines load and identify the OracleAS Port Tunnel process module. Management of OracleAS Port Tunnel processes by the process module are identified by the `module id`.

```
<module path="$ORACLE_HOME/opmn/lib/libopmniaspt.so">
  <module-id id="IASPT" />
</module>
```

## 8.2 OracleAS Port Tunnel Minimum Configuration

The following lines represent the minimum configuration for OracleAS Port Tunnel. Default values are assigned to all other configuration elements and attributes for OracleAS Port Tunnel.

```
<ias-component id="IASPT">
  <process-type id="IASPT" module-id="IASPT">
    <process-set id="IASPT" numprocs="1"/>
  </process-type>
</ias-component>
```

## 8.3 OracleAS Port Tunnel Complete Configuration

The following example represents the complete configuration for OracleAS Port Tunnel. It contains all possible configuration elements and attributes that can be used with OracleAS Port Tunnel.

```
<module path="$ORACLE_HOME/opmn/lib/libopmniaspt.so">
  <module-id id="IASPT" />
</module>
<ias-component id="IASPT" status="enabled" id-matching="false">
  <process-type id="IASPT" module-id="IASPT">
    <port id="ajp" range="6701-6703" />
    <process-set id="IASPT" restart-on-death="true" id="ajp" />
  </process-type>
</ias-component>
```

## 8.4 OracleAS Port Tunnel Attribute Descriptions

This section describes the attributes that are specific for OracleAS Port Tunnel.

The OracleAS Port Tunnel attributes are described with the following format:

- **Title:** This is the attribute name and value being defined. For example, `id="IASPT"`.
- **Required:** This field defines whether or not the attribute is required in the component definition.
- **Default:** This defines the default value assigned to the attribute. The default value appears in the installed version of the `opmn.xml` file or is assigned internally if the attribute is not present.
- **Valid values:** If applicable, this field defines the valid values for the attribute. For example, `IASPT`.
- **Path:** This field defines in which elements the attribute can appear. For example, `ias-component`.

### **id="IASPT"**

Required: true  
Default: none  
Valid values: IASPT  
Path: ias-component  
Path: ias-component/process-type  
Path: ias-component/process-set

The `id` name is required and cannot be changed. The `id` name must match the entry in the `targets.xml` file.

### **module-id="IASPT"**

Required: true  
Default: none  
Path: ias-component/process-type

The `module-id` name defines the type of process and associates the configuration with a process module.

### **id="ajp"**

Required: false  
Default: none  
Valid values: ajp  
Path: ias-component/process-type/port



The `id` value should be used together with `range` in `port` property to specify the `ajp` ports to be used by the OracleAS Port Tunnel server. If the `id` is specified, the `port` number configured in the `iaspt.conf` file is overwritten.

**`range="6701-6703"`**

Required: false

Default: none

Valid values: Any single port or a range of ports

Path: `ias-component/process-type/port`

The `range` value should be used together with `ajp` in `port` property to specify the `ajp` ports to be used by OracleAS Port Tunnel servers.

**`numprocs="3"`**

Required: true

Default: none

Valid values: Any number

Path: `ias-component/process-type/process-set`

This attribute tells how many OracleAS Port Tunnel server processes to be started. The `ajp` range should be configured in the `port` property if the value is 1. If the value is greater than 1, `ajp` range has to be configured to specify enough ports for each OracleAS Port Tunnel server process. Typically, the value is 1 port for each process.



# Configuring Oracle Application Server Wireless

This chapter describes Oracle Application Server Wireless (OracleAS Wireless) configuration in the OPMN `opmn.xml` file.

It features the following topics:

- [Section 9.1, "OracleAS Wireless Process Module Configuration"](#)
- [Section 9.2, "OracleAS Wireless Minimum Configuration"](#)
- [Section 9.3, "OracleAS Wireless Complete Configuration"](#)
- [Section 9.4, "OracleAS Wireless Attribute Descriptions"](#)

## 9.1 OracleAS Wireless Process Module Configuration

The following lines load and identify the OracleAS Wireless process module. Management of OracleAS Wireless processes by the process module are identified by the module id.

```
<module path="$ORACLE_HOME/opmn/lib/libopmnwireless.so">
  <module-id="notificationeventcollector"/>
  <module-id="datafeeder"/>
  <module-id="performance"/>
  <module-id="messaging"/>
  <module-id="notification"/>
  <module-id="location"/>
  <module-id="telnetserver"/>
  <module-id="telnetdispatcher"/>
  <module-id="edgeserver"/>
</module>
```

## 9.2 OracleAS Wireless Minimum Configuration

The following lines represent the minimum configuration for OracleAS Wireless. Default values are assigned to all other configuration elements and attributes for OracleAS Wireless.

```
<ias-component id="wireless" status="enabled">
  <environment>
    <variable id="WIRELESS_HOME" value="$ORACLE_HOME/wireless" append="false"/>
    <variable id="WIRELESS_LIB" value="$ORACLE_HOME/wireless/lib" append="false"/>
    <variable id="JAVA13_HOME" value="$ORACLE_HOME/jdk" append="false"/>
    <variable id="JAVA_HOME" value="$ORACLE_HOME/jdk" append="false"/>
  </environment>
</ias-component>
```

```
<variable id="IAS_LIB" value="$ORACLE_HOME/lib" append="false"/>
<variable id="LD_LIBRARY_PATH" value="$ORACLE_HOME/wireless/lib" append="true" />
<variable id="CLASSPATH" value="." append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jdk/jre/lib/rt.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jdk/lib/tools.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jdk/jre/lib/ext/activation.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jdk/jre/lib/ext/mail.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jdbc/lib/classes12.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jdbc/lib/nls_charset12.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/server/classes" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/server/classes/messages"
  append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/wireless.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/wireless_tools.jar"
  append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/wireless_modules.jar"
  append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/wireless_drm.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/xserver.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/mwa.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/jabberbeans.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/bcel.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/pop3.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/xmlrpc.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/batik-css.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/batik-dom.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/batik-ext.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/batik-svg-dom.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/batik-util.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/panama_modules.zip"
  append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/panama_modules_commerce.zip"
  append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/panama_modules_infra.zip"
  append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/panama_modules_location.zip"
  append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/panama_modules_pim.zip"
  append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/mod_pim_webdav_httpclient.jar"
  append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/studio.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/client.zip" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/server.zip" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/diagnostics/lib/ojdl.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/webcache/jlib/jawc.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/portal/jlib/pdkjava.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/portal/jlib/ptlshare.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/lib/xmlparserv2.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/lib/xschema.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/lib/xforms.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/lib/servlet.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/lib/classgen.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/ohw.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/ui2.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/share.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/jdev-cm.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/fndctx.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/wfapi.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/regexp.jar" append="true"/>
```

```

<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/javax-ssl-1_2.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/jssl-1_2.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/ojmisc.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/emPid.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/netcfg.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/providerutil.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/repository.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/jndi.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/ldap.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/ldapjclnt9.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/ord/jlib/jai_codec.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/ord/jlib/jai_core.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/ord/jlib/jpec_codec.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/ord/jlib/ordimimg.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/ord/jlib/ordim.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/ord/jlib/ordhttp.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/datadirect/sun/lib/ldap/ldapbp.jar"
  append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/sso/lib/ssosdk902.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/sso/lib/ossoreg.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/lbs/lib/sdoapi.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/lbs/lib/sdovis.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/dcm/lib/dcm.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/j2ee/home/lib/http_client.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/j2ee/home/jazn.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/j2ee/home/jaas.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/soap/lib/soap.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/soap/lib/wsdl.jar" append="true"/>
</environment>
<dependencies>
  <OID infrastructure="true" timeout="720" />
  <database infrastructure-key="wireless" timeout="720" />
</dependencies>
<process-type id="datafeeder_server" module-id="datafeeder">
  <stop timeout="300"/>
  <restart timeout="480"/>
  <process-set id="datafeeder_instance_1" numprocs="1" status="disabled"/>
</process-type>
<process-type id="performance_server" module-id="performance">
  <start timeout="420"/>
  <stop timeout="420"/>
  <process-set id="performancemonitor1" numprocs="1">
    <start timeout="420"/>
    <stop timeout="420"/>
    <restart timeout="420"/>
  </process-set>
</process-type>
<process-type id="messaging_server" module-id="messaging">
  <start timeout="420"/>
  <stop timeout="420"/>
  <process-set id="messagingserver1" numprocs="1">
    <start timeout="420"/>
    <stop timeout="420"/>
    <restart timeout="420"/>
  </process-set>
</process-type>
<process-type id="notificationeventcollector_server" module-id="notificationeventcollector">
  <start timeout="420"/>
  <stop timeout="420"/>
  <process-set id="notificationeventcollector1" numprocs="1" status="disabled">

```

```
<start timeout="420"/>
<stop timeout="420"/>
<restart timeout="420"/>
</process-set>
</process-type>
<process-type id="notification_server" module-id="notification">
  <module-data>
    <category id="start-parameters">
      <data id="java-parameters" value="-Xms64M -Xmx256M"/>
    </category>
  </module-data>
  <start timeout="1320"/>
  <stop timeout="420"/>
  <process-set id="notificationengine1" numprocs="1" status="disabled">
    <start timeout="1320"/>
    <stop timeout="420"/>
    <restart timeout="420"/>
  </process-set>
</process-type>
<process-type id="location_server" module-id="location">
  <start timeout="420"/>
  <stop timeout="420"/>
  <process-set id="locationeventserver1" numprocs="1" status="disabled">
    <start timeout="420"/>
    <stop timeout="420"/>
    <restart timeout="420"/>
  </process-set>
</process-type>
<process-type id="telnet_server" module-id="telnetserver">
  <start timeout="420"/>
  <stop timeout="420"/>
  <process-set id="telnet_server1" numprocs="1">
    <start timeout="420"/>
    <stop timeout="420"/>
    <restart timeout="420"/>
  </process-set>
</process-type>
<process-type id="telnet_dispatcher" module-id="telnetdispatcher">
  <start timeout="420"/>
  <stop timeout="420"/>
  <process-set id="telnet_dispatcher1" numprocs="1" status="disabled">
    <start timeout="420"/>
    <stop timeout="420"/>
    <restart timeout="420"/>
  </process-set>
</process-type>
<process-type id="edge_server" module-id="edgeserver">
  <start timeout="420"/>
  <stop timeout="420"/>
  <process-set id="edgeserver1" numprocs="1" status="disabled">
    <start timeout="420"/>
    <stop timeout="420"/>
    <restart timeout="420"/>
  </process-set>
</process-type>
</ias-component>
```

## 9.3 OracleAS Wireless Complete Configuration

The following lines show a complete configuration for OracleAS Wireless. It contains all possible configuration elements and attributes for OracleAS Wireless.

```
<ias-component id="wireless" status="enabled">
  <environment>
    <variable id="WIRELESS_HOME" value="$ORACLE_HOME/wireless" append="false"/>
    <variable id="WIRELESS_LIB" value="$ORACLE_HOME/wireless/lib" append="false"/>
    <variable id="JAVA13_HOME" value="$ORACLE_HOME/jdk" append="false"/>
    <variable id="JAVA_HOME" value="$ORACLE_HOME/jdk" append="false"/>
    <variable id="IAS_LIB" value="$ORACLE_HOME/lib" append="false"/>
    <variable id="LD_LIBRARY_PATH" value="$ORACLE_HOME/wireless/lib" append="true" />
    <variable id="CLASSPATH" value="." append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/jdk/jre/lib/rt.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/jdk/lib/tools.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/jdk/jre/lib/ext/activation.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/jdk/jre/lib/ext/mail.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/jdbc/lib/classes12.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/jdbc/lib/nls_charset12.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/server/classes" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/server/classes/messages"
      append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/wireless.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/wireless_tools.jar" append=
      "true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/wireless_modules.jar" append=
      "true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/wireless_drm.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/xserver.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/mwa.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/jabberbeans.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/bcel.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/pop3.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/xmlrpc.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/batik-css.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/batik-dom.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/batik-ext.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/batik-svg-dom.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/batik-util.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/panama_modules.zip" append=
      "true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/panama_modules_commerce.zip"
      append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/panama_modules_infra.zip" append=
      "true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/panama_modules_location.zip"
      append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/panama_modules_pim.zip" append=
      "true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/mod_pim_webdav_httpclient.jar"
      append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/studio.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/client.zip" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/server.zip" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/diagnostics/lib/ojdl.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/webcache/jlib/jawc.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/portal/jlib/pdkjava.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/portal/jlib/ptlshare.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/lib/xmlparserv2.jar" append="true"/>
    <variable id="CLASSPATH" value="$ORACLE_HOME/lib/xschema.jar" append="true"/>
  </environment>
</ias-component>
```

```

<variable id="CLASSPATH" value="$ORACLE_HOME/lib/xforms.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/lib/servlet.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/lib/classgen.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/ohw.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/uix2.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/share.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/jdev-cm.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/fndctx.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/wfapi.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/regexp.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/javax-ssl-1_2.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/jssl-1_2.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/ojmisc.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/emPid.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/netcfg.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/providerutil.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/repository.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/jndi.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/ldap.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/jlib/ldapjclnt9.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/ord/jlib/jai_codec.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/ord/jlib/jai_core.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/ord/jlib/jpec_codec.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/ord/jlib/ordimimg.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/ord/jlib/ordim.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/ord/jlib/ordhttp.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/datadirect/sun/lib/ldap/ldapbp.jar" append="
true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/sso/lib/ssosdk902.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/sso/lib/ossoreg.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/lbs/lib/sdoapi.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/lbs/lib/sdovis.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/dcm/lib/dcm.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/j2ee/home/lib/http_client.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/j2ee/home/jazn.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/j2ee/home/jaas.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/soap/lib/soap.jar" append="true"/>
<variable id="CLASSPATH" value="$ORACLE_HOME/soap/lib/wsdl.jar" append="true"/>
</environment>
<dependencies>
  <OID infrastructure="true" timeout="720" />
  <database infrastructure-key="wireless" timeout="720" />
</dependencies>
<process-type id="datafeeder_server" module-id="datafeeder">
  <module-data>
    <category id="start-parameters">
      <data id="java-bin" value="/my/path/to/java/exec"/>
      <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
-Dyet_another_definition=yet_another_value"/>
      <data id="class-name" value="MyJavaStartClass"/>
      <data id="application-parameters" value="my_application_parameters"/>
    </category>
    <category id="stop-parameters">
      <data id="java-bin" value="/my/path/to/java/exec"/>
      <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
-Dyet_another_definition=yet_another_value"/>
      <data id="class-name" value="MyJavaStopClass"/>
      <data id="application-parameters" value="my_application_parameters"/>
    </category>
  </module-data>

```



```

<stop timeout="300"/>
<restart timeout="480"/>
<process-set id="datafeeder_instance_1" numprocs="1" status="disabled"/>
</process-type>
<process-type id="performance_server" module-id="performance">
  <module-data>
    <category id="start-parameters">
      <data id="java-bin" value="/my/path/to/java/exec"/>
      <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
        -Dyet_another_definition=yet_another_value"/>
      <data id="class-name" value="MyJavaStartClass"/>
      <data id="application-parameters" value="my_application_parameters"/>
    </category>
    <category id="stop-parameters">
      <data id="java-bin" value="/my/path/to/java/exec"/>
      <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
        -Dyet_another_definition=yet_another_value"/>
      <data id="class-name" value="MyJavaStopClass"/>
      <data id="application-parameters" value="my_application_parameters"/>
    </category>
  </module-data>
  <start timeout="420"/>
  <stop timeout="420"/>
  <process-set id="performancemonitor1" numprocs="1">
    <start timeout="420"/>
    <stop timeout="420"/>
    <restart timeout="420"/>
  </process-set>
</process-type>
<process-type id="messaging_server" module-id="messaging">
  <module-data>
    <category id="start-parameters">
      <data id="java-bin" value="/my/path/to/java/exec"/>
      <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
        -Dyet_another_definition=yet_another_value"/>
      <data id="class-name" value="MyJavaStartClass"/>
      <data id="application-parameters" value="my_application_parameters"/>
    </category>
    <category id="stop-parameters">
      <data id="java-bin" value="/my/path/to/java/exec"/>
      <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
        -Dyet_another_definition=yet_another_value"/>
      <data id="class-name" value="MyJavaStopClass"/>
      <data id="application-parameters" value="my_application_parameters"/>
    </category>
  </module-data>
  <start timeout="420"/>
  <stop timeout="420"/>
  <process-set id="messagingserver1" numprocs="1">
    <start timeout="420"/>
    <stop timeout="420"/>
    <restart timeout="420"/>
  </process-set>
</process-type>
<process-type id="notificationeventcollector_server" module-id="notificationeventcollector">
  <module-data>
    <category id="start-parameters">
      <data id="java-bin" value="/my/path/to/java/exec"/>
      <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
        -Dyet_another_definition=yet_another_value"/>
    </category>
  </module-data>
  <start timeout="420"/>
  <stop timeout="420"/>
  <process-set id="notificationeventcollector1" numprocs="1">
    <start timeout="420"/>
    <stop timeout="420"/>
    <restart timeout="420"/>
  </process-set>
</process-type>

```

```

    <data id="class-name" value="MyJavaStartClass"/>
    <data id="application-parameters" value="my_application_parameters"/>
</category>
<category id="stop-parameters">
    <data id="java-bin" value="/my/path/to/java/exec"/>
    <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
    -Dyet_another_definition=yet_another_value"/>
    <data id="class-name" value="MyJavaStopClass"/>
    <data id="application-parameters" value="my_application_parameters"/>
</category>
</module-data>
<start timeout="420"/>
<stop timeout="420"/>
<process-set id="notificationeventcollector1" numprocs="1" status="disabled">
    <start timeout="420"/>
    <stop timeout="420"/>
    <restart timeout="420"/>
</process-set>
</process-type>
<process-type id="notification_server" module-id="notification">
    <module-data>
        <category id="start-parameters">
            <data id="java-bin" value="/my/path/to/java/exec"/>
            <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
            -Dyet_another_definition=yet_another_value"/>
            <data id="class-name" value="MyJavaStartClass"/>
            <data id="application-parameters" value="my_application_parameters"/>
        </category>
        <category id="stop-parameters">
            <data id="java-bin" value="/my/path/to/java/exec"/>
            <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
            -Dyet_another_definition=yet_another_value"/>
            <data id="class-name" value="MyJavaStopClass"/>
            <data id="application-parameters" value="my_application_parameters"/>
        </category>
    </module-data>
    <start timeout="1320"/>
    <stop timeout="420"/>
    <process-set id="notificationengine1" numprocs="1" status="disabled">
        <start timeout="1320"/>
        <stop timeout="420"/>
        <restart timeout="420"/>
    </process-set>
</process-type>
<process-type id="location_server" module-id="location">
    <module-data>
        <category id="start-parameters">
            <data id="java-bin" value="/my/path/to/java/exec"/>
            <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
            -Dyet_another_definition=yet_another_value"/>
            <data id="class-name" value="MyJavaStartClass"/>
            <data id="application-parameters" value="my_application_parameters"/>
        </category>
        <category id="stop-parameters">
            <data id="java-bin" value="/my/path/to/java/exec"/>
            <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
            -Dyet_another_definition=yet_another_value"/>
            <data id="class-name" value="MyJavaStopClass"/>
            <data id="application-parameters" value="my_application_parameters"/>
        </category>
    </module-data>
    <start timeout="1320"/>
    <stop timeout="420"/>
    <process-set id="locationengine1" numprocs="1" status="disabled">
        <start timeout="1320"/>
        <stop timeout="420"/>
        <restart timeout="420"/>
    </process-set>
</process-type>

```

```

</module-data>
<start timeout="420"/>
<stop timeout="420"/>
<process-set id="locationeventserver1" numprocs="1" status="disabled">
  <start timeout="420"/>
  <stop timeout="420"/>
  <restart timeout="420"/>
</process-set>
</process-type>
<process-type id="telnet_server" module-id="telnetserver">
  <module-data>
    <category id="start-parameters">
      <data id="java-bin" value="/my/path/to/java/exec"/>
      <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
        -Dyet_another_definition=yet_another_value"/>
      <data id="class-name" value="MyJavaStartClass"/>
      <data id="application-parameters" value="my_application_parameters"/>
    </category>
    <category id="stop-parameters">
      <data id="java-bin" value="/my/path/to/java/exec"/>
      <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
        -Dyet_another_definition=yet_another_value"/>
      <data id="class-name" value="MyJavaStopClass"/>
      <data id="application-parameters" value="my_application_parameters"/>
    </category>
  </module-data>
  <start timeout="420"/>
  <stop timeout="420"/>
  <process-set id="telnet_server1" numprocs="1">
    <start timeout="420"/>
    <stop timeout="420"/>
    <restart timeout="420"/>
  </process-set>
</process-type>
<process-type id="telnet_dispatcher" module-id="telnetdispatcher">
  <module-data>
    <category id="start-parameters">
      <data id="java-bin" value="/my/path/to/java/exec"/>
      <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
        -Dyet_another_definition=yet_another_value"/>
      <data id="class-name" value="MyJavaStartClass"/>
      <data id="application-parameters" value="my_application_parameters"/>
    </category>
    <category id="stop-parameters">
      <data id="java-bin" value="/my/path/to/java/exec"/>
      <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
        -Dyet_another_definition=yet_another_value"/>
      <data id="class-name" value="MyJavaStopClass"/>
      <data id="application-parameters" value="my_application_parameters"/>
    </category>
  </module-data>
  <start timeout="420"/>
  <stop timeout="420"/>
  <process-set id="telnet_dispatcher1" numprocs="1" status="disabled">
    <start timeout="420"/>
    <stop timeout="420"/>
    <restart timeout="420"/>
  </process-set>
</process-type>
<process-type id="edge_server" module-id="edgeserver">

```

```
<module-data>
  <category id="start-parameters">
    <data id="java-bin" value="/my/path/to/java/exec"/>
    <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
      -Dyet_another_definition=yet_another_value"/>
    <data id="class-name" value="MyJavaStartClass"/>
    <data id="application-parameters" value="my_application_parameters"/>
  </category>
  <category id="stop-parameters">
    <data id="java-bin" value="/my/path/to/java/exec"/>
    <data id="java-parameters" value="-Doracle.wireless.instance=my_messaging_instance_name
      -Dyet_another_definition=yet_another_value"/>
    <data id="class-name" value="MyJavaStopClass"/>
    <data id="application-parameters" value="my_application_parameters"/>
  </category>
</module-data>
<start timeout="420"/>
<stop timeout="420"/>
<process-set id="edgeserver1" numprocs="1" status="disabled">
  <start timeout="420"/>
  <stop timeout="420"/>
  <restart timeout="420"/>
</process-set>
</process-type>
</ias-component>
```

## 9.4 OracleAS Wireless Attribute Descriptions

This section describes the attributes that are specific for OracleAS Wireless.

The OracleAS Wireless attributes are described with the following format:

- **Title:** This is the attribute name and value being defined. For example, **id="wireless"**.
- **Required:** This field defines whether or not the attribute is required in the component definition.
- **Default:** This defines the default value assigned to the attribute. The default value appears in the installed version of the `opmn.xml` file or is assigned internally if the attribute is not present.
- **Valid values:** If applicable, this field defines the valid values for the attribute. For example, `wireless`.
- **Path:** This field defines in which elements the attribute can appear. For example, `ias-component`.

### **id="wireless"**

Required: true  
Default: none  
Valid values: wireless  
Path: ias-component

This `id` is required and cannot be changed. The name must match the `targets.xml` entry or elements and attributes will not work.

**module-id="notificationeventcollector"**

Required: true  
Default: none  
Path: `ias-component/process-type`

This `module-id` defines the type of process and associates this configuration with a process module. In [Section 9.1](#), the nine OracleAS Wireless server types are associated with OracleAS Wireless process module.

**module-id="datafeeder"**

Required: true  
Default: none  
Path: `ias-component/process-type`

This `module-id` defines the type of process and associates this configuration with a process module. In [Section 9.1](#), the nine OracleAS Wireless server types are associated with OracleAS Wireless process module.

**module-id="performance"**

Required: true  
Default: none  
Path: `ias-component/process-type`

This `module-id` defines the type of process and associates this configuration with a process module. In [Section 9.1](#), the nine OracleAS Wireless server types are associated with OracleAS Wireless process module.

**module-id="messaging"**

Required: true  
Default: none  
Path: `ias-component/process-type`

This `module-id` defines the type of process and associates this configuration with a process module. In [Section 9.1](#), the nine OracleAS Wireless server types are associated with OracleAS Wireless process module.

**module-id="notification"**

Required: true  
Default: none  
Path: `ias-component/process-type`

This `module-id` defines the type of process and associates this configuration with a process module. In [Section 9.1](#), the nine OracleAS Wireless server types are associated with OracleAS Wireless process module.

**module-id="location"**

Required: true  
Default: none  
Path: `ias-component/process-type`

This `module-id` defines the type of process and associates this configuration with a process module. In [Section 9.1](#), the nine OracleAS Wireless server types are associated with OracleAS Wireless process module.

**module-id="telnetserver"**

Required: true  
Default: none  
Path: ias-component/process-type

This module-id defines the type of process and associates this configuration with a process module. In [Section 9.1](#), the nine OracleAS Wireless server types are associated with OracleAS Wireless process module.

**module-id="telnetdispatcher"**

Required: true  
Default: none  
Path: ias-component/process-type

This module-id defines the type of process and associates this configuration with a process module. In [Section 9.1](#), the nine OracleAS Wireless server types are associated with OracleAS Wireless process module.

**module-id="edgeserver"**

Required: true  
Default: none  
Path: ias-component/process-type

This module-id defines the type of process and associates this configuration with a process module. In [Section 9.1](#), the nine OracleAS Wireless server types are associated with OracleAS Wireless process module.

**numprocs="1"**

Required: true  
Default: none  
Valid values: 1  
Path: ias-component/process-type/process-set

This attribute gives the number of OracleAS Wireless instances started for this process-set. The OracleAS Wireless module does not permit more than one instance with the same process-set. To start more than one processes for the same server type, put one process-set entry with one unique process-set id for each process.

For example:

```
<process-type id="datafeeder_server" module-id="datafeeder">  
  <process-set id="datafeeder_instance_1" numprocs="1"/>  
  <process-set id="datafeeder_instance_2" numprocs="1"/>  
</process-type>
```

Do not create the following:

```
<process-type id="datafeeder_server" module-id="datafeeder">  
  <process-set id="datafeeder_instance_1" numprocs="2"/>  
</process-type>
```

**id="start-parameters"**

Required: false  
Default: none  
Path: ias-component/process-type/process-set/module-data/

### category

This `id` indicates the configuration block for `start-parameters` category. This category contains data related to starting the module.

#### **id="java-bin"**

Required: false

Default: `$ORACLE_HOME/jdk/bin/java`

Valid values: Full path to `java.exe`

Path: `ias-component/process-type/process-set/module-data/category/data`

You can specify alternate paths to the Java executable. However, a valid version of Java will have to be used for the process to start up and work correctly.

#### **id="java-parameters"**

Required: false

Default: `-Doracle.wireless.instance=<id specified in process set>`

`-Doracle.opmn.uid=<opmn assigned unique id>`

Valid values: Any valid Java parameters

Path: `ias-component/process-type/process-set/module-data/category/data`

This `id` enables you to specify any valid Java parameters. All specified Java parameters will be appended to the default values. OPMN derives some java parameters internally and additional parameters can be added with this attribute. If you specify `"-Doracle.wireless.instance=<value>"` or `"-Doracle.opmn.uid=<value>"`, it will replace the default value.

#### **id="class-name"**

Required: false

Default: `oracle.panama.sysmgmt.server.PanamaServer` for start and `oracle.panama.sysmgmt.client.manager` for stop

Valid values: Any valid Java class names

Path: `ias-component/process-type/process-set/module-data/category/data`

This `id` enables you to specify a Java class name other than the default ones to start or stop a OracleAS Wireless process. Oracle recommends not changing this parameter; it is for internal use only.

#### **id="application-parameters"**

Required: false

Default: none

Valid values: any valid Java application parameters

Path: `ias-component/process-type/process-set/module-data/category/data`

This `id` enables you to put more application parameters in addition to the default one. The default one tells which OracleAS Wireless `process-type` to be started.

#### **id="stop-parameters"**

Required: false

Default: none

Path: `ias-component/process-type/process-set/module-data/category`

This `id` indicates the configuration block for `stop-parameters` category. In this category, some stop related module data can be specified.

**See Also:** `id="start-parameters"`

**start timeout="<seconds>"**

Required: false

Default: 120

Valid values: A timeout value in seconds

Path: `ias-component/process-type/process-set/module-data/category`

This parameter notifies OPMN how long it should wait for the process to start. The default value is 180 seconds. However, when starting OracleAS Wireless processes, database access is required and it may take up to 5 minutes to signal a process to start.

**stop timeout="<seconds>"**

Required: false

Default: 60

Valid values: A timeout value in seconds

Path: `ias-component/process-type/process-set/module-data/category`

This parameter notifies OPMN how long it should wait for the process to terminate. The default value is 180 seconds. However, when stopping OracleAS Wireless processes, database access is required and it may take up to 5 minutes to signal a process to terminate.

**restart timeout="<seconds>"**

Required: false

Default: 180

Valid values: A timeout value in seconds

Path: `ias-component/process-type/process-set/module-data/category`

This parameter notifies OPMN how long it should wait for a process to restart. The default value is 180 seconds. However, a restart action is actually a stop and start, so the total value should be:

180 (for start)

+ 300 (for stop)

=480 seconds.



---

# Configuring Oracle Application Server Reports Services

This chapter describes Oracle Application Server Reports Services (OracleAS Reports Services) configuration in the OPMN `opmn.xml` file. The OracleAS Reports Services module manages the Reports Server and Reports Bridge processes.

It features the following topics:

- [Section 10.1, "Introduction"](#)
- [Section 10.2, "OracleAS Reports Services Process Module Configuration"](#)
- [Section 10.3, "Reports Server Minimum Configuration"](#)
- [Section 10.4, "Reports Server Full Configuration"](#)
- [Section 10.7, "OracleAS Reports Services Attribute Descriptions"](#)

## 10.1 Introduction

The following sections describe the processes that the OracleAS Reports Services module manages:

- [Section 10.1.1, "Reports Server"](#)
- [Section 10.1.2, "Reports Bridge"](#)

### 10.1.1 Reports Server

The Reports Server can run in two modes:

- in-process
- standalone

The in-process server runs inside an OC4J instance and is managed by the OC4J module. In this mode, the OracleAS Reports Services module does not have to be configured. The OracleAS Reports Services server configuration is part of the OC4J instance configuration and is handled by the OC4J module.

The standalone OracleAS Reports Services server is managed by the OracleAS Reports Services module. The OracleAS Reports Services process module must be configured as shown in the [Section 10.2, "OracleAS Reports Services Process Module Configuration"](#).

## 10.1.2 Reports Bridge

Reports Bridge is a service used by OracleAS Reports Services clients to locate a Reports Server running in a remote subnet.

## 10.2 OracleAS Reports Services Process Module Configuration

The following lines load and identify the OracleAS Reports Services process module. Management of OracleAS Reports Services processes by the process module are identified by the module id.

```
<module path="$ORACLE_HOME/opmn/lib/libopmnreports.so">
  <module-id id="ReportsServices" />
  <module-id id="ReportsBridgeServices" />
</module>
```

## 10.3 Reports Server Minimum Configuration

This section describes the minimum configuration for Reports Server. It features the following topics:

- [Section 10.3.1, "In-process Reports Server Minimum Configuration"](#)
- [Section 10.3.2, "Standalone Reports Server Minimum Configuration"](#)

### 10.3.1 In-process Reports Server Minimum Configuration

The following example shows the minimum configuration required for the in-process Reports Server server. The lines should be placed within the configuration for the appropriate OC4J instance. The configuration directives are handled by the OC4J module.

```
<ias-component ....>
  <process-type ....>
    <module-data>
      <category id="urlping-parameters">
        <data id="/reports/rwservlet/pingserver?start=auto" value="200"/>
      </category>
    </module-data>
  </process-type>
</ias-component>
```

### 10.3.2 Standalone Reports Server Minimum Configuration

The following example represents the minimum configuration for a standalone Reports Server component. Reasonable default values are assigned to all other configuration elements and attributes that can be used with this component.

```
<ias-component id="ReportsServer1">
  <process-type id="ReportsServer1" module-id="ReportsServices">
    <process-set id="ReportsServer1" numprocs="1">
      <environment>
        <variable id="PATH" value="Put Path_value from env here" append="true"/>
        <variable id="LD_LIBRARY_PATH" value="Put LD_Library_Path_value from env here"
          append="true"/>
      </environment>
    </process-set>
  </process-type>
```

```
</ias-component>
```

## 10.4 Reports Server Full Configuration

The following example represents the full configuration for Reports Server. It contains all possible configuration elements and attributes that can be used with this component. This section features the following topics:

- [Section 10.4.1, "In-process Reports Server Full Configuration"](#)
- [Section 10.4.2, "Standalone Reports Server Full Configuration"](#)

### 10.4.1 In-process Reports Server Full Configuration

The following example shows the full configuration required for the in-process Reports Server. The lines should be placed within the configuration for the appropriate OC4J instance. The configuration directives are handled by the OC4J module.

```
<ias-component ....>
  <process-type ....>
    <module-data>
      <category id="urlping-parameters">
        <data id="/reports/rwservlet/pingserver?start=auto" value="200"/>
      </category>
    </module-data>
    <process-set .../>
  </process-type>
</ias-component>
```

### 10.4.2 Standalone Reports Server Full Configuration

The following example shows the full configuration required for the standalone Reports Server.

```
<ias-component id="ReportsServer1" status="enabled" id-matching="false">
  <process-type id="ReportsServer1" module-id="ReportsServices">
    <process-set id="ReportsServer1" restart-on-death="true" numprocs="1">
      <environment>
        <variable id="PATH" value="Put Path_value here" append="true"/>
        <variable id="LD_LIBRARY_PATH" value="Put LD_Library_Path_value here" append="true">
      </environment>
      <module-data>
        <category id="general-parameters">
          <data id="batch" value="yes"/>
        </category>
        <category id="restart-parameters">
          <data id="reverseping-timeout" value="300"/>
        </category>
      </module-data>
      <start timeout="300" retry="3"/>
      <stop timeout="300"/>
      <restart timeout="300"/>
      <ping timeout="30" interval="30"/>
    </process-set>
  </process-type>
</ias-component>
```

## 10.5 Reports Bridge Minimum Configuration

The following example represents the minimum configuration for a standalone Reports Bridge component. Reasonable default values are assigned to all other configuration elements and attributes that can be used with this component.

```
<ias-component id="RWBridge" status="enabled" id-matching="false">
  <process-type id="ReportsBridge" module-id="ReportsBridgeServices">
    <process-set id="RWBridge" restart-on-death="true" numprocs="1">
      <environment>
        <variable id="PATH" value="$ORACLE_HOME/jdk/jre/bin" append="true"/>
        <variable id="PATH" value="Put Path_value from env here" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/jlib/zrclient.jar" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/reports/jlib/rwrun.jar" append="true"/>
      </environment>
    </process-set>
  </process-type>
</ias-component>
```

## 10.6 Reports Bridge Full Configuration

The following example represents the full configuration for a Reports Bridge component. Reasonable default values are assigned to all other configuration elements and attributes that can be used with this component.

```
<ias-component id="RWBridge" status="enabled" id-matching="false">
  <process-type id="ReportsBridge" module-id="ReportsBridgeServices">
    <process-set id="RWBridge" restart-on-death="true" numprocs="1">
      <environment>
        <variable id="PATH" value="$ORACLE_HOME/jdk/jre/bin" append="true"/>
        <variable id="PATH" value="Put Path_value here" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/jlib/zrclient.jar" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/reports/jlib/rwrun.jar" append="true"/>
      </environment>
      <module-data>
        <category id="restart-parameters">
          <data id="reverseping-timeout" value="120"/>
        </category>
        <category id="start-parameters">
          <data id="jvm-options" value="-Xms128mb -Xmx256mb"/>
          <data id="bridge-options" value="Test"/>
        </category>
        <category id="stop-parameters">
          <data id="jvm-options" value="-Xms128mb -Xmx256mb"/>
          <data id="bridge-options" value="Test"/>
        </category>
      </module-data>
      <start timeout="120" retry="3"/>
      <stop timeout="120"/>
      <restart timeout="120" retry="0"/>
    </process-set>
  </process-type>
</ias-component>
```

## 10.7 OracleAS Reports Services Attribute Descriptions

This section describes the attributes that are specific for OracleAS Reports Services.

The OracleAS Reports Services attributes are described with the following format:

- **Title:** This is the attribute name and value being defined. For example, `id="ReportsServer1"`.
- **Required:** This field defines whether or not the attribute is required in the component definition.
- **Default:** This defines the default value assigned to the attribute. The default value appears in the installed version of the `opmn.xml` file or is assigned internally if the attribute is not present.
- **Valid values:** This field defines the valid values for the attribute. For example, `ReportsServer1`.
- **Path:** This field defines in which elements the attribute can appear. For example, `ias-component`.

#### **id="ReportsServer1"**

Required: true  
 Default: none  
 Valid values: `ReportsServer1`  
 Path: `ias-component`  
 Path: `ias-component/process-type`  
 Path: `ias-component/process-type/process-set`

This `id` is required and cannot be changed. The name must match the `targets.xml` entry or elements and attributes will not work.

#### **id="RWBridge"**

Required: true  
 Default: none  
 Valid values: A string  
 Path: `ias-component`

A string can be specified as the name of the bridge.

#### **process-type id="ReportsBridge"**

Required: true  
 Default: none  
 Valid values: `ReportsBridge`  
 Path: `ias-component/process-type`

This `id` is required and cannot be changed.

#### **process-set id="RWBridge"**

Required: true  
 Default: none  
 Valid values: `ReportsBridge`  
 Path: `ias-component/process-type/process-set`

A string can be specified as the name of the bridge. The name must have the same value as specified in the `id` for the `ias-component`.

#### **module-id="ReportsServices"**

Required: true

Default: ReportServices  
Valid values: ReportServices  
Path: ias-component/process-type

This module-id defines the type of process and associates this configuration with the OracleAS Reports Services process module. This module-id is specified when the OracleAS Reports Services module is used to control the Reports Server.

**module-id="ReportsBridgeServices"**

Required: true  
Default: none  
Valid values: ReportsBridgeServices  
Path: ias-component/process-type

This id defines the type of process and associates the configuration with the OracleAS Reports Services process module. This id is used when the OracleAS Reports Services module is used to control the Reports Bridge services.

**numprocs="1"**

Required: true  
Default: none  
Valid values: 1  
Path: ias-component/process-type/process-set

This attribute gives the number of OracleAS Reports Services servers started for the process-set. Its value has to be 1 because the process-set id identifies the OracleAS Reports Services server.

**id="PATH"**

Required: true  
Default: none  
Valid values: none  
Path: ias-component/process-type/process-set/environment

This is the PATH environment variable set for the OracleAS Reports Services process.

**value="Put Path\_value here"**

Required: true  
Default: none  
Valid values: The path value has to include the directory containing the uname command.  
Path: ias-component/process-type/process-set/environment/variable

This is the value of the PATH environment variable set for the OracleAS Reports Services process. This variable has to be set for the start script to find "uname".

**id="general-parameters"**

Required: false  
Default: none  
Path: ias-component/process-type/process-set/module-data/category

This category collects all the data that is common to the various process management tasks.

**id="batch"**

Required: false  
 Default: none  
 Valid values: yes, no  
 Path: `ias-component/process-type/process-set/module-data/category/data`

This `id` is used to construct the command line for starting and stopping of the OracleAS Reports Services server. The information is passed through to OracleAS Reports Services, without interpretation or use by OPMN. If it is not configured, this argument is not passed in.

**See Also:** *Oracle Application Server Reports Services Publishing Reports to the Web*

**id="restart-parameters"**

Required: false  
 Default: none  
 Path: `ias-component/process-type/process-set/module-data/category`

The `restart-parameters` category for defines parameters used in death detection.

**id="reverseping-timeout"**

Required: false  
 Default: 300  
 Valid values: Any reasonable timeout value specified in seconds.  
 Path: `ias-component/process-type/process-set/module-data/category/data`

If OPMN does not receive a notification within the last `reverseping-timeout` period, it restarts the process.

**id="urlping-parameters"**

Required: false  
 Default: none  
 Path: `ias-component/process-type/process-set/module-data/category`

This `id` is a category that collects all the URLs that will be pinged by the OC4J module. The protocol used for ping is AJPv1.3.

**id="/reports/rwservlet/pingserver?start=auto"**

Required: false  
 Default: none  
 Valid values: `/reports/rwservlet/pingserver?start=auto`  
 Path: `ias-component/process-type/process-set/module-data/category/data`

This data `id` specifies the URL that will be pinged by the OC4J module. In the context of the in-process OracleAS Reports Services server, pingging this URL allows OPMN to

determine if the OracleAS Reports Services server application is responsive or not. If the application is unresponsive, OPMN restarts the corresponding OC4J process.

**value="200"**

Required: false

Default: none

Valid values: 200

Path: ias-component/process-type/process-set/module-data/category/data

The value specifies a valid HTTP code that is expected in response to the ping request. If the response HTTP code matches the value configured here, OPMN considers the application to be valid. Otherwise, OPMN restarts the OC4J process.

**id="start-parameters"**

Required: false

Default: none

Path: ias-component/process-type/process-set/module-data/category

The `start-parameters` category is used for defining parameters that will be used in starting Reports Bridge.

**id="stop-parameters"**

Required: false

Default: none

Path: ias-component/process-type/process-set/module-data/category

The `stop-parameters` category is used for defining parameters that will be used in stopping the Reports Bridge.

**id="jvm-options"**

Required: false

Default: none

Valid values: Valid Java Virtual Machine (JVM) options

Path: ias-component/process-type/process-set/module-data/category/data

The `jvm-options` are supplied to the JVM while starting or stopping the Reports Bridge.

**id="bridge-options"**

Required: false

Default: none

Path: ias-component/process-type/process-set/module-data/category/data

The `bridge-options` are passed on to the Reports Bridge during starting and stopping. These can be any custom command line options that are supported for starting and stopping the Reports Bridge.

**id="CLASSPATH"**

Required: true



Default: none

Path: `ias-component/process-type/process-set/environment/`

The CLASSPATH value must point to `$ORACLE_HOME/jlib/zrclient.jar` and `$ORACLE_HOME/reports/jlib/rwrun.jar`. The classes required for running the Reports Bridge are available in these .jar files.



# Configuring Oracle Business Intelligence Discoverer

This chapter describes Oracle Business Intelligence Discoverer (OracleBI Discoverer) configuration in the OPMN `opmn.xml` file.

It features the following topics:

- [Section 11.1, "OracleBI Discoverer Process Module Configuration"](#)
- [Section 11.2, "OracleBI Discoverer Minimum Configuration"](#)
- [Section 11.3, "OracleBI Discoverer Complete Configuration"](#)
- [Section 11.4, "OracleBI Discoverer Attribute Descriptions"](#)

## 11.1 OracleBI Discoverer Process Module Configuration

The following lines load and identify the OracleBI Discoverer process module. Management of OracleBI Discoverer processes by the process module are identified by the module id.

```
<module path="$ORACLE_HOME/opmn/lib/libopmndisco.so">
  <module-id id="Disco_ServicesStatus"/>
  <module-id id="Disco_SessionServer"/>
  <module-id id="Disco_PreferenceServer"/>
</module>
```

## 11.2 OracleBI Discoverer Minimum Configuration

The following lines represent the minimum configuration for OracleBI Discoverer. Default values are assigned to all other configuration elements and attributes for OracleBI Discoverer.

```
<ias-component id="Discoverer">
  <environment>
    <variable id="PREFERENCE_PORT" value="16001"/>
    <variable id="DISCO_DIR" value="$ORACLE_HOME/discoverer"/>
    <variable id="LD_LIBRARY_PATH" value="$DISCO_DIR/lib:$ORACLE_HOME/lib" append="true"/>
  </environment>
  <process-type id="SessionServer" module-id="Disco_SessionServer" >
    <port id="ses" range="3801-3950"/>
    <process-set id="SessionServer" minprocs="0" maxprocs=150 restart-on-death="false"
      parallel-requests="true"/>
  </process-type>
  <process-type id="ServicesStatus" module-id="Disco_ServicesStatus">
```

```

    <process-set id="ServicesStatus" numprocs="1"/>
  </process-type>
  <process-type id="PreferenceServer" module-id="Disco_PreferenceServer">
    <process-set id="PreferenceServer" numprocs="1"/>
  </process-type>
</ias-component>

```

## 11.3 OracleBI Discoverer Complete Configuration

The following lines show a complete configuration for OracleBI Discoverer. It contains all possible configuration elements and attributes for OracleBI Discoverer.

```

<ias-component id="Discoverer">
  <environment>
    <variable id="DISCO_DIR" value="$ORACLE_HOME/discoverer"/>
    <variable id="PREFERENCE_PORT" value="16001"/>
    <variable id="LD_LIBRARY_PATH" value="$DISCO_DIR/lib:$ORACLE_HOME/lib" append="true"/>
  </environment>
  <process-type id="SessionServer" module-id="Disco_SessionServer">
    <ping interval="40"/>
    <port id="ses" range="3801-3950"/>
    <process-set id="SessionServer" minprocs="0" maxprocs="150" restart-on-death="false"
      parallel-requests="true"/>
    <module-data>
      <category id="start-parameters">
        <data id="dis51ws-options" value="dis51ws-options_value"/>
        <data id="dis51ws-extra-options" value="dis51ws-extra-options_value"/>
      </category>
    </module-data>
  </process-type>
  <process-type id="ServicesStatus" module-id="Disco_ServicesStatus">
    <process-set id="ServicesStatus" numprocs="1"/>
  </process-type>
  <process-type id="PreferenceServer" module-id="Disco_PreferenceServer" >
    <process-set id="PreferenceServer" numprocs="1"/>
    <module-data>
      <category id="start-parameters">
        <data id="dis51pr-options" value="dis51pr-options_value"/>
        <data id="dis51pr-extra-options" value="dis51pr-extra-options_value"/>
      </category>
    </module-data>
  </process-type>
</ias-component>

```

## 11.4 OracleBI Discoverer Attribute Descriptions

This section describes the attributes that are specific for OracleBI Discoverer.

The OracleBI Discoverer attributes are described with the following format:

- **Title:** This is the attribute name and value being defined. For example, **id="Discoverer"**.
- **Required:** This field defines whether or not the attribute is required in the component definition.
- **Default:** This defines the default value assigned to the attribute. The default value appears in the installed version of the `opmn.xml` file or is assigned internally if the attribute is not present.

- **Valid values:** If applicable, this field defines the valid values for the attribute. For example, `Discoverer`.
- **Path:** This field defines in which elements the attribute can appear. For example, `ias-component`.

**id="Discoverer"**

Required: true  
 Default: none  
 Valid values: `Discover`  
 Path: `ias-component`

This `id` is required and cannot be changed. The name must match the `targets.xml` entry or Application Server Control Console will not work.

**id="DISCO\_DIR"**

Required: true  
 Default: none  
 Path: `ias-component/environment`

This environment variable is required for OPMN to locate the binaries and libraries of OracleBI Discoverer components, such as `dis51pr`.

**id="PREFERENCE\_PORT"**

Required: true  
 Default: none  
 Path: `ias-component/environment`

This environment variable defines the port for the preference server.

**id="LD\_LIBRARY\_PATH"**

Required: true  
 Default: none  
 Path: `ias-component/environment/variable`

This environment variable is required for OracleBI Discoverer binaries to find the libraries.

The following directories are required:

- `$DISCO_DIR/lib`
- `$ORACLE_HOME/lib`

**id="SessionServer"**

Required: true  
 Default: none  
 Valid values: `SessionServer`  
 Path: `ias-component/process-type`

This `id` is required and cannot be changed. The name must match the `targets.xml` entry or Application Server Control Console will not work.

**module-id="Disco\_SessionServer"**

Required: true

Default: none

Path: `ias-component/process-type`

This `id` defines the type of process and associates this configuration with a process module.

**interval="40"**

Required: false

Default: 20 seconds

Valid values:  $\geq 0$

Path: `ias-component/process-type/ping`

Please refer to the common configuration for a full list of attributes for the `ping` element. The code example in [Section 11.3, "OracleBI Discoverer Complete Configuration"](#) shows a ping interval to ping a process every 40 seconds. Oracle recommends a 40 second ping interval for Session servers.

**id="ses"**

Required: false

Default: "ses"

Valid values: "ses"

Path: `ias-component/process-type/port`

Please refer to the common configuration for a full list of attributes for "port" element. "ses" is the port id under which the range of ports for the Session server is specified.

**range="3801-3950"**

Required: false

Default: 3801-3950

Valid values:  $\geq 0$

Path: `ias-component/process-type/port`

Please refer to the common configuration for a full list of attributes for "port" element. When configuring ports for the Session servers make sure there are enough ports; the number of ports should be equal to `maxprocs`.

**minprocs="0"**

Required: true

Default: none

Valid values: 1

Path: `ias-component/process-type/process-set`

This attribute gives the number of Session server instances started for this `process-set`. This should be set to zero so OPMN does not attempt to start the Session servers when a `startproc` command is executed on OracleBI Discoverer.

**maxprocs="150"**

Required: true

Default: none

Valid values: 1

Path: `ias-component/process-type/process-set`

This attribute gives the number of Session server instances started for this process-set. This should be configured such that the maximum number of processes does not exhaust computer resources.

**restart-on-death="false"**

Required: true  
 Default: true  
 Valid values: true, false  
 Path: `ias-component/process-type/process-set`

This attribute notifies OPMN whether to attempt to restart a process after it stops. For Session servers this should always be set to "false".

**parallel-requests="true"**

Required: false  
 Default: false  
 Valid values: true, false  
 Path: `ias-component/process-type/process-set`

This attribute notifies OPMN on how to handle incoming requests for this process-set. If set to true the requests will be handled in parallel. If set to false the requests will be handled serially.

**id="start-parameters"**

Required: false  
 Default: none  
 Path: `ias-component/process-type/process-set/module-data/category`

This is the category to specify the startup parameters for Session servers.

**id="dis51ws-options"**

Required: false  
 Default: *Refer to the following paragraph.*  
 Path: `ias-component/process-type/process-set/module-data/category/data`

The default command line options for `$DISCO_DIR/bin/dis51ws` are:

```
prompt> "-opmn_ping_time <ping interval> -port <port> -preferencePort $PREFERENCE_
PORT -uid <unique identifier>"
```

You can override these default options by specifying valid command line options for this element. The unique identifier (`-uid`) is added at the end of the command line. Make sure you include the `-uid` at the end of your specified options.

**id="dis51ws-extra-options"**

Required: false  
 Default: *Refer to the following paragraph.*  
 Path: `ias-component/process-type/process-set/module-data/category/data`

The default command line options for `$DISCO_DIR/bin/dis51ws` are:

```
prompt> "-opmn_ping_time <ping interval> -port <port> -preferencePort $PREFERENCE_
```

```
PORT -uid <unique identifier>"
```

You can add to default options by specifying valid command line options for this element. If `dis51ws-options` is specified this element will be ignored.

**id="ServicesStatus"**

Required: true  
Default: none  
Valid values: ServicesStatus  
Path: `ias-component/process-type`

This `id` is required and cannot be changed. The name must match the `targets.xml` entry or Application Server Control Console will not work.

**module-id="Disco\_ServicesStatus"**

Required: true  
Default: none  
Valid values: ServicesStatus  
Path: `ias-component/process-type`

This name defines the type of process and associates this configuration with a process module.

**numprocs="1"**

Required: true  
Default: none  
Valid values: 1  
Path: `ias-component/process-type/process-set`

This attribute gives the number of ServicesStatus instances started for this `process-set`. There should be at most one ServicesStatus instance running in one Oracle Application Server instance.

**id="PreferenceServer"**

Required: true  
Default: none  
Valid values: Preference Server  
Path: `ias-component/process-type`

This `id` is required and cannot be changed. The name must match the `targets.xml` entry or Application Server Control Console will not work.

**module-id="Disco\_PreferenceServer"**

Required: true  
Default: none  
Path: `ias-component/process-type`

This name defines the type of process and associates this configuration with a process module.

**numprocs="1"**

Required: false  
Default: none



Path: `ias-component/process-type/process-set`

This attribute gives the number of Preference server instances started for this process set. There should be at most one instance running in one Oracle Application Server instance.

**id="start-parameters"**

Required: false

Default: none

Path: `ias-component/process-type/process-set/module-data/category`

This is the category to specify the ping parameters for `dis51pr`.

**id="dis51pr-options"**

Required: false

Default: *Refer to the following paragraph.*

Path: `ias-component/process-type/process-set/module-data/category/data`

The default command line option for `$DISCO_DIR/bin/dis51pr` is:

```
prompt> "-ORBEndpoint iiop://:$PREFERENCE_PORT -U <unique identifier>"
```

You can override the default option by specifying valid command line options for this element. The unique identifier (`-U`) is added to the end of the command line. Make sure to include `-U` at the end of your command option.

**id="dis51pr-extra-options"**

Required: false

Default: *Refer to the following paragraph.*

Path: `ias-component/process-type/process-set/module-data/category/data`

The default command line option for `$DISCO_DIR/bin/dis51pr` is:

```
prompt> "-ORBEndpoint iiop://:$PREFERENCE_PORT -U <unique identifier>"
```

You can override the default option by specifying valid command line options for this element. The unique identifier (`-U`) is added to the end of the command line. Make sure to include `-U` at the end of your command option.



## Configuring Log Loader

This chapter describes Log Loader configuration in the OPMN `opmn.xml` file.

It features the following topics:

- [Section 12.1, "Log Loader Process Module Configuration"](#)
- [Section 12.2, "Log Loader Minimum Configuration"](#)
- [Section 12.3, "Log Loader Complete Configuration"](#)
- [Section 12.4, "Log Loader Attribute Descriptions"](#)

### 12.1 Log Loader Process Module Configuration

The following lines load and identify the Log Loader process module. Management of Log Loader processes by the process module are identified by the `module id`.

```
<module path="$ORACLE_HOME/opmn/lib/liblogloader.so">
  <module-id id="LOGLDR"/>
</module>
```

### 12.2 Log Loader Minimum Configuration

The following lines represent the minimum configuration for Log Loader. Default values are assigned to all other configuration elements and attributes for Log Loader.

```
<ias-component id="LogLoader" status="enabled" id-matching="true">
  <process-type id="logloaderd" module-id="LOGLDR" working-dir="$ORACLE_HOME">
    <environment>
      <variable id="CLASSPATH" value="$ORACLE_HOME/diagnostics/lib/ojdl.jar" append="true"/>
      <variable id="CLASSPATH" value="$ORACLE_HOME/lib/xmlparserv2.jar" append="true"/>
      <variable id="CLASSPATH" value="$ORACLE_HOME/jdbc/lib/classes12.jar" append="true"/>
      <variable id="CLASSPATH" value="$ORACLE_HOME/jdbc/lib/nls_charset12.jar" append="true"/>
      <variable id="CLASSPATH" value="$ORACLE_HOME/opmn/lib/ons.jar" append="true"/>
      <variable id="CLASSPATH" value="$ORACLE_HOME/opmn/lib/optic.jar" append="true"/>
      <variable id="CLASSPATH" value="$ORACLE_HOME/jlib/oraclepki.jar" append="true"/>
    </environment>
    <process-set id="logloaderd" restart-on-death="true" numprocs="1">
      <module-data>
        <category id="start-parameters">
          <data id="java-parameters" value="-server -Doracle.home=$ORACLE_HOME
            -Djava.security.policy=$ORACLE_HOME/diagnostics/bin/java.policy"/>
        </category>
        <category id="stop-parameters">
          <data id="java-parameters" value="-Doracle.home=$ORACLE_HOME -Djava.security.policy
```

```

        =$ORACLE_HOME/diagnostics/bin/java.policy"/>
    </category>
</module-data>
</process-set>
</process-type>
</ias-component>

```

## 12.3 Log Loader Complete Configuration

The following lines show the complete configuration for Log Loader. It contains all possible configuration elements and attributes for Log Loader.

```

<ias-component id="LogLoader" status="enabled" id-matching="true">
  <process-type id="logloaderd" module-id="LOGLDR" working-dir="$ORACLE_HOME">
    <environment>
      <variable id="CLASSPATH" value="$ORACLE_HOME/diagnostics/lib/ojdl.jar" append="true"/>
      <variable id="CLASSPATH" value="$ORACLE_HOME/lib/xmlparserv2.jar" append="true"/>
      <variable id="CLASSPATH" value="$ORACLE_HOME/jdbc/lib/classes12.jar" append="true"/>
      <variable id="CLASSPATH" value="$ORACLE_HOME/jdbc/lib/nls_charset12.jar" append="true"/>
      <variable id="CLASSPATH" value="$ORACLE_HOME/opmn/lib/ons.jar" append="true"/>
      <variable id="CLASSPATH" value="$ORACLE_HOME/opmn/lib/optic.jar" append="true"/>
      <variable id="CLASSPATH" value="$ORACLE_HOME/jlib/oraclepki.jar" append="true"/>
    </environment>
    <process-set id="logloaderd" restart-on-death="true" numprocs="1">
      <module-data>
        <category id="start-parameters">
          <data id="java-bin" value="$ORACLE_HOME/jdk/bin/java"/>
          <data id="java-parameters" value="-server-Doracle.home= $ORACLE_HOME
            -Djava.security.policy=$ORACLE_HOME/diagnostics/bin/java.policy"/>
          <data id="class-name" value="oracle.core.ojdl.loader.LogLoaderDaemon"/>
          <data id="application-parameters" value="-start -config $ORACLE_HOME/diagnostics/
            config/logloader.xml"/>
        </category>
        <category id="stop-parameters">
          <data id="java-bin" value="$ORACLE_HOME/jdk/bin/java"/>
          <data id="java-parameters" value="-Doracle.home=$ORACLE_HOME
            -Djava.security.policy=$ORACLE_HOME/diagnostics/ bin/java.policy"/>
          <data id="class-name" value="oracle.core.ojdl.loader.LogLoaderDaemon"/>
          <data id="application-parameters" value="-stop"/>
        </category>
      </module-data>
    </process-set>
  </process-type>
</ias-component>

```

## 12.4 Log Loader Attribute Descriptions

This section describes the attributes that are specific for Log Loader. This section also provides attribute descriptions of the attributes.

The Log Loader attributes are described with the following format:

- **Title:** This is the attribute name and value being defined. For example, **id="LogLoader"**.
- **Required:** This field defines whether or not the attribute is required in the component definition.

- **Default:** This defines the default value assigned to the attribute. The default value appears in the installed version of the `opmn.xml` file or is assigned internally if the attribute is not present.
- **Valid values:** If applicable, this field defines the valid values for the attribute. For example, `LogLoader`.
- **Path:** This field defines in which elements the attribute can appear. For example, `ias-component`.

**id="LogLoader"**

Required: true  
Default: none  
Valid values: `LogLoader`  
Path: `ias-component`

This `id` is required and cannot be changed.

**id="logloaderd"**

Required: true  
Default: none  
Valid values: `logloaderd`  
Path: `ias-component/process-type`  
Path: `ias-component/process-type/process-set`

This `id` is required and cannot be changed.

**module-id="LOGLDR"**

Required: true  
Default: none  
Path: `ias-component/process-type`

This name defines the type of process and associates this configuration with a process module.

**numprocs="1"**

Required: true  
Default: none  
Valid values: 1  
Path: `ias-component/process-type/process-set`

This attribute gives the number of LogLoader server instances started for this `process-set`. The value should always be 1.

**id="start-parameters"**

Required: true  
Default: none  
Path: `ias-component/process-type/process-set/module-data/category`

This indicates the configuration block for `start-parameters` category. This category specifies the Java-parameters data must be specified.

**id="java-bin"**

Required: false

Default: `$ORACLE_HOME/jdk/bin/java`

Valid values: Any full file path to a Java executable file

Path: `ias-component/process-type/process-set/module-data/category/data`

You can specify alternate paths to the Java executable.

**id="java-parameters"**

Required: true

Default: None

Valid values: Any valid Java parameters

Path: `ias-component/process-type/process-set/module-data/category/data`

The parameters `-Doracle.home` and `-Djava.security.policy` are required and must be defined in this section.

**id="class-name"**

Required: false

Default: `oracle.core.ojdl.loader.LogLoaderDaemon`

Valid values: a Java class name

Path: `ias-component/process-type/process-set/module-data/category/data`

This field enables you to specify a Java class name other than the default to start or stop the LogLoader server process. This `id` is for internal use and must not be changed by the user.

**id="application-parameters"**

Required: false

Default: To start: `-start -config $ORACLE_HOME/diagnostics/config/logloader.xml`.

To stop: `-stop -config $ORACLE_HOME/diagnostics/config/logloader.xml`

Valid values: any valid Java application parameters that are recognized by the class specified in "class-name".

Path: `ias-component/process-type/process-set/module-data/category/data`

This field enables you to set application parameters. Any parameters defined here will replace the default values.

**id="stop-parameters"**

Required: false

Default: none

Path: `ias-component/process-type/process-set/module-data/category`

This indicates the configuration block for `stop-parameters` category. In this category, some stop related module data can be specified.

## Configuring DCM Daemon

This chapter describes DCM Daemon configuration in the OPMN `opmn.xml` file.

It features the following topics:

- [Section 13.1, "DCM Daemon Process Module Configuration"](#)
- [Section 13.2, "DCM Daemon Minimum Configuration"](#)
- [Section 13.3, "DCM Daemon Complete Configuration"](#)
- [Section 13.4, "DCM Daemon Attribute Descriptions"](#)

### 13.1 DCM Daemon Process Module Configuration

The following lines load and identify the DCM Daemon process module. Management of DCM Daemon processes by the process module are identified by the `module id`.

```
<module path="$ORACLE_HOME/opmn/lib/libopmndcmdaemon">
  <module-id id="DCMDaemon"/>
</module>
```

### 13.2 DCM Daemon Minimum Configuration

The following lines represent the minimum configuration for the DCM Daemon. Default values are assigned to all other configuration elements and attributes for DCM Daemon.

```
<ias-component id="dcm-daemon" status="enabled" id-matching="true">
  <process-type id="dcm-daemon" module-id="DCMDaemon">
    <process-set id="dcm" numprocs="1">
      <module-data>
        <category id="start-parameters">
          <data id="jar-file" value="$ORACLE_HOME/dcm/lib/dcm.jar"/>
          <data id="application-parameters" value="daemon -logdir $ORACLE_HOME/dcm/logs/
            daemon_logs -o $ORACLE_HOME"/>
        </category>
      </module-data>
    </process-set>
  </process-type>
</ias-component>
```

### 13.3 DCM Daemon Complete Configuration

The following lines show a complete configuration for the DCM Daemon. It contains all possible configuration elements and attributes for the DCM Daemon.

```
<ias-component id="dcm-daemon" status="enabled" id-matching="true">
  <process-type id="dcm-daemon" module-id="DCMDaemon">
    <process-set id="dcm" numprocs="1">
      <module-data>
        <category id="start-parameters">
          <data id="java-bin" value="$ORACLE_HOME/jdk/bin/java" />
          <data id="java-parameters" value="-Doracle.ias.sysmgmt.logging.loglevel=ERROR"/>
          <data id="jar-file" value="$ORACLE_HOME/dcm/lib/dcm.jar" />
          <data id="application-parameters" value="daemon -logdir $ORACLE_HOME/dcm/logs
            /daemon_logs -o $ORACLE_HOME"/>
        </category>
      </module-data>
    </process-set>
  </process-type>
</ias-component>
```

## 13.4 DCM Daemon Attribute Descriptions

This section describes the attributes that are specific for the DCM Daemon. This section also provides attribute descriptions of the attributes.

The DCM Daemon attributes are described with the following format:

- **Title:** This is the attribute name and value being defined. For example, **id="dcm-daemon"**.
- **Required:** This field defines whether or not the attribute is required in the component definition.
- **Default:** This defines the default value assigned to the attribute. The default value appears in the installed version of the `opmn.xml` file or is assigned internally if the attribute is not present.
- **Valid values:** If applicable, this field defines the valid values for the attribute. For example, `dcm-daemon`.
- **Path:** This field defines in which elements the attribute can appear. For example, `ias-component`.

### **id="dcm-daemon"**

Required: true  
Default: none  
Valid values: `dcm-daemon`  
Path: `ias-component`  
Path: `ias-component/process-type`

This `id` is required and cannot be changed.

### **module-id="DCMDaemon"**

Required: true  
Default: none  
Path: `ias-component/process-type`

This `id` defines the type of process and associates this configuration with a process module. This name is required and cannot be changed.

### **numprocs="1"**

Required: true



Default: none

Valid values: 1

Path: `ias-component/process-type/process-set`

This attribute gives the number of DCM Daemon instances started for the `process-set`. The value should always be 1.

#### **id="start-parameters"**

Required: true

Default: none

Path: `ias-component/process-type/process-set/module-data/category`

This indicates the configuration block for the `start-parameters` category. In this category, at a minimum, the Java-parameters data must be specified.

#### **id="java-bin"**

Required: false

Default: `$ORACLE_HOME/jdk/bin/java`

Valid values: Full path to `java.exe`

Path: `ias-component/process-type/process-set/module-data/category/data`

You can specify alternate paths to the Java Interpreter (executable).

#### **id="java-parameters"**

Required: false

Default: `-Doracle.ias.sysmgmt.logging.loglevel=ERROR`

Valid values: Any valid Java parameters

Path: `ias-component/process-type/process-set/module-data/category/data`

This field enables you to specify any valid Java parameters. All specified Java parameters overwrite the default values. If the `loglevel` is not specified, the `loglevel` for the DCM daemon is set at the `ERROR` level.

#### **id="jar-file"**

Required: true

Default: `$ORACLE_HOME/dcm/lib/dcm.jar`

Valid values: full path to valid `dcm.jar`

Path: `ias-component/process-type/process-set/module-data/category`

This field specifies a JAR file for starting and stopping the DCM daemon. This `id` is required and cannot be changed.

#### **id="application-parameters"**

Required: true

Default: `daemon -logdir $ORACLE_HOME/dcm/logs/daemon_logs -o $ORACLE_HOME`

Valid values: any valid Java application parameters that are recognized by the main-class specified in `dcm.jars manifest.mf`

Path: `ias-component/process-type/process-set/module-data/category/data`

This field enables you to set application parameters. Parameters defined here replace the default values. In order to start the DCM Daemon, it is mandatory to supply `daemon`, `-logdir`, and `-o` parameters.

## Configuring Oracle Application Server Integration Business Activity Monitoring

This chapter describes Oracle Application Server Integration Business Activity Monitoring (OracleAS Integration BAM) configuration in the OPMN `opmn.xml` file.

It features the following topics:

- [Section 14.1, "OracleAS Integration BAM Process Module Configuration"](#)
- [Section 14.2, "OracleAS Integration BAM Minimum Configuration"](#)
- [Section 14.3, "OracleAS Integration BAM Complete Configuration"](#)
- [Section 14.4, "OracleAS Integration BAM Attribute Descriptions"](#)

### 14.1 OracleAS Integration BAM Process Module Configuration

The following lines load and identify the process module. Management of OracleAS Integration BAM processes by the process module are identified by the `module id`.

```
<module path="$ORACLE_HOME/opmn/lib/libopmnbam.so">
  <module-id id="MonitorFramework"/>
  <module-id id="BAMEngine"/>
</module>
```

### 14.2 OracleAS Integration BAM Minimum Configuration

The following lines represent the minimum configuration for OracleAS Integration BAM. Default values are assigned to all other configuration elements and attributes for OracleAS Integration BAM.

```
<ias-component id="BAM" status="enabled">
  <process-type id="engine" module-id="BAMEngine" >
    <environment>
      <variable id="BAM_PROPERTIES_FILE" value="$ORACLE_HOME/work/BAMProperties/
        bam.properties"/>
      <variable id="TNS_INSTANCE" value="bamhost"/>
      <variable id="LD_LIBRARY_PATH" value="/usr/local/lib:/usr/lib:$ORACLE_HOME/lib32"
        append="true"/>
    </environment>
    <process-set id="BAMEngine" numprocs="1">
      <module-data>
        <category id="start-parameters">
          <data id="bin-path" value="$ORACLE_HOME/bam/bin/wsd"/>
        </category>
```

```

        </module-data>
    </process-set>
</process-type>
<process-type id="monitor" module-id="MonitorFramework">
    <environment>
        <variable id="CLASSPATH" value="$ORACLE_HOME/bam/config" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/aqapi.jar" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/bam.jar" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/jdbc/lib/classes12.jar" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/dms.jar" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/ip.jar" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/b2b.jar" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/oai904.jar" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/sqlj/lib/runtime12ee.jar" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/sqlj/lib/translator.jar" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/xmlcomp.jar" append="true"/>
        <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/xmlparserv2_10g.jar" append="true"/>
    </environment>
    <process-set id="MonitorFramework" numprocs="1"/>
</process-type>
</ias-component>

```

## 14.3 OracleAS Integration BAM Complete Configuration

The following lines show a complete configuration for OracleAS Integration BAM. It contains all possible configuration elements and attributes for OracleAS Integration BAM.

```

<ias-component id= BAM status="enabled">
    <process-type id="engine" module-id="BAMEngine">
        <environment>
            <variable id="BAM_PROPERTIES_FILE" value= "$ORACLE_HOME/work/BAMProperties
                /bam.properties"/>
            <variable id="TNS_INSTANCE" value="bamhost"/>
            <variable id="LD_LIBRARY_PATH" value= "/usr/local/lib:/usr/lib:$ORACLE_HOME/lib32"
                append="true"/>
        </environment>
        <process-set id="BAMEngine" numprocs="1">
            <module-data>
                <category id="start-parameters">
                    <data id="bin_path" value="$ORACLE_HOME/bam/bin/wsd"/>
                </category>
            </module-data>
        </process-set>
    </process-type>
    <process-type id="monitor" module id="Monitor Framework">
        <environment>
            <variable id="CLASSPATH" value="$ORACLE_HOME/bam/config" append="true"/>
            <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/aqapi.jar" append="true"/>
            <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/bam.jar" append="true"/>
            <variable id="CLASSPATH" value="$ORACLE_HOME/jdbc/lib/classes12.jar" append="true"/>
            <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/dms.jar" append="true"/>
            <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/ip.jar" append="true"/>
            <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/b2b.jar" append="true"/>
            <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/oai904.jar" append="true"/>
            <variable id="CLASSPATH" value="$ORACLE_HOME/sqlj/lib/runtime12ee.jar" append="true"/>
            <variable id="CLASSPATH" value="$ORACLE_HOME/sqlj/lib/translator.jar" append="true"/>
            <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/xmlcomp.jar" append="true"/>
            <variable id="CLASSPATH" value="$ORACLE_HOME/bam/lib/xmlparserv2_10g.jar" append="true"/>
        </environment>
    </process-type>

```

```

</environment>
  <module-data>
    <category id="start-parameters">
      <data id="java-bin" value="/my/path/to/java/exec"/>
      <data id="java-parameters" value="-ms8m -mx512m"/>
      <data id="class-name" value="MyJavaStartClass"/>
      <data id="application-parameters" value="MyApplicationParameters"/>
    </category>
    <category id="stop-parameters">
      <data id="java-bin" value="/my/path/to/java/exec"/>
      <data id="java-parameters" value="-ms8m -mx512m"/>
      <data id="class-name" value="MyJavaStopClass"/>
      <data id="application-parameters" value="MyApplicationParameters"/>
      <data id="stop-option" value="MyStopOption"/>
    </category>
  </module-data>
  <process-set id="MonitorFramework" numprocs="1"/>
  <stop timeout="MyStopTimeout">
</process-type>
</ias-component>

```

## 14.4 OracleAS Integration BAM Attribute Descriptions

This section describes the attributes that are specific for OracleAS Integration BAM.

The OracleAS Integration BAM attributes are described with the following format:

- **Title:** This is the attribute name and value being defined. For example, **id="BAM"**.
- **Required:** This field defines whether or not the attribute is required in the component definition.
- **Default:** This defines the default value assigned to the attribute. The default value appears in the installed version of the `opmn.xml` file or is assigned internally if the attribute is not present.
- **Valid values:** This field defines the valid values for the attribute. For example, BAM.
- **Path:** This field defines in which elements the attribute can appear. For example, `ias-component`.

### id= BAM

Required: true  
 Default: none  
 Valid values: BAM  
 Path: `ias-component`

This `id` is required and cannot be changed. The `id` name must match the `targets.xml` entry or Application Server Control Console will not work.

### id="BAM\_PROPERTIES\_FILE"

Required: true  
 Default: none  
 Path: `ias-component/environment`

This environment variable defines the full path to the `bam.properties` file.

**module-id="BAMEngine"**

Required: true  
 Default: none  
 Valid values: BAMEngine  
 Path: ias-component

This id is required and should not be changed to anything else.

**id="TNS\_INSTANCE"**

Required: true  
 Valid values: Unique string name  
 Path: ias-component/environment

This environmental variable is required for OracleAS Integration BAM services.

**id="LD\_LIBRARY\_PATH"**

Required: true  
 Default: none  
 Path: ias-component/environment

This environment variable is required by OracleAS Integration BAM binaries to find their respective libraries.

**id="bin\_path"**

Required: true  
 Default: none  
 Path: ias-component/process-type/module-data/category  
 Path: ias-component/process-type/process-set/module-data/category

This element must contain the full path (including the executable) to the process that starts the OracleAS Integration BAM engine.

**category id="start-parameters"**

Required: false  
 Default: none  
 Path: ias-component/module-data/category  
 Path: ias-component/process-type/module-data/category  
 Path: ias-component/process-type/process-set/module-data/category

This element indicates the configuration block for the start-parameters category. In this category, data associated with the stop module can be specified.

**category id="stop-parameters"**

Required: false  
 Default: none  
 Path: ias-component/module-data/category  
 Path: ias-component/process-type/module-data/category  
 Path: ias-component/process-type/process-set/module-data/category

This element indicates the configuration block for the stop-parameters category. In this category, data associated with the stop module can be specified.

**module id="Monitor Framework"**

Required: true  
 Default: none  
 Valid values: MonitorFramework  
 Path: ias-component/process-type

This id defines the type of process and associates the configuration with a process module.

**data id="java-bin"**

Required: false  
 Default: \$ORACLE\_HOME/jdk/bin/java  
 Valid values: Any full file path to a java executable file  
 Path: ias-component/module-data/category/data  
 Path: ias-component/process-type/module-data/category/data  
 Path: ias-component/process-type/process-set/module-data/category/data

You can specify alternate paths to the java executable. However, a valid version of java will have to be used for the process to start up and work correctly. Oracle recommends JDK 1.4.1 or above.

**data id="java-parameters"**

Required: false  
 Default: none  
 Valid values: "-ms8m -mx512m" or any other valid java parameters  
 Path: ias-component/module-data/category/data  
 Path: ias-component/process-type/module-data/category/data  
 Path: ias-component/process-type/process-set/module-data/category/data

You can specify any valid java parameters in the command line or replace existing default parameters using the default parameters list.

**data id="class-name"**

Required: false  
 Default: oracle.tip.bam.runtime.ProcessManager  
 Valid values: A valid java class name to start/stop BAM MonitorFramework  
 Path: ias-component/module-data/category/data  
 Path: ias-component/process-type/module-data/category/data  
 Path: ias-component/process-type/process-set/module-data/category/data

You can specify the java class name which will be used for starting or stopping the OracleAS Integration BAM monitor framework process.

**data id="application-parameters"**

Required: false  
 Default: "MonitorFramework start" or "MonitorFramework stop"  
 Valid values: Valid application parameters  
 Path: ias-component/module-data/category/data  
 Path: ias-component/process-type/module-data/category/data  
 Path: ias-component/process-type/process-set/module-data/category/data

data

You can specify valid application parameters which will be used for starting or stopping the OracleAS Integration BAM monitor framework processes.

**data id="stop-option"**

Required: false

Default: none

Valid values: force

Path: ias-component/module-data/category/data

Path: ias-component/process-type/module-data/category/data

Path: ias-component/process-type/process-set/module-data/category/data

You can specify a valid stop option to force MonitorFramework to shutdown immediately without sending events to the repository.

**process-set id="mf\_instance1"**

Required: true

Default: none

Valid values: A process instance name, for example "MonitorFramework".

Path: ias-component/process-type/process-set

This parameter is to tell the process instance name for this process-set. This instance name will be passed into OracleAS Integration BAM management class for starting or stopping Monitor Framework.

**numprocs="1"**

Required: true

Default: none

Valid values: 1

Path: ias-component/process-type/process-set

This attribute specifies the number of the instances started for this process set. OracleAS Integration BAM currently only enables you to start one instance with the same process type.

**stop timeout="MyStopTimeout"**

Required: false

Default: 60

Valid values: Any valid timeout in seconds.

Path: ias-component/process-type/

This parameter is to give a timeout in seconds before opmn hard kill the Monitor Framework process. A longer timeout enables Monitor Framework to flush the buffered events into repository.



---

## Configuring Custom Process

This chapter describes custom process configuration in the OPMN `opmn.xml` file.

It features the following topics:

- [Section 15.1, "Custom Process Module Configuration"](#)
- [Section 15.2, "Custom Process Minimum Configuration"](#)
- [Section 15.3, "Custom Process Complete Configuration"](#)
- [Section 15.4, "Custom Process Attribute Descriptions"](#)

### 15.1 Custom Process Module Configuration

The following lines load and identify the custom process module. Management of custom processes by the process module are identified by the `module id`.

```
<module path="$ORACLE_HOME/opmn/lib/libopmncustom.so">
  <module-id id="CUSTOM" />
</module>
```

### 15.2 Custom Process Minimum Configuration

The following lines represent the minimum configuration for a custom process. Default values are assigned to all other configuration elements and attributes for the custom process.

```
<ias-component id="Custom">
  <process-type id="Custom" module-id="CUSTOM">
    <process-set id="Custom" numprocs="1">
      <module-data>
        <category id="start-parameters">
          <data id="start-executable" value="Your start executable here" />
        </category>
      </module-data>
    </process-set>
  </process-type>
</ias-component>
```

### 15.3 Custom Process Complete Configuration

[Example 15-1](#) show a complete configuration for a custom process. It contains all possible configuration elements and attributes for a custom process.

A custom process can be part of any other Oracle Application Server component. In such cases, the `process-type` element in [Example 15–1](#) must be part of the component configuration.

### 15.3.1 Ping

The custom module provides the framework for pinging a custom process in one of two ways:

- HTTP ping
- script ping

The type of ping can be configured by specifying the appropriate data in the `ping-parameters` category. The sample Oracle Application Server configuration example [Example 15–1](#) shows a custom process using HTTP ping. [Example 15–2](#) is an example of script ping that you can substitute into the component configuration.

#### **Example 15–1 Custom Process Complete Configuration**

```
<ias-component id="Custom" status="enabled" id-matching="false">
  <environment>
    <variable id="TEST_ENV_VARIABLE" value="/your/test/value"
      append="false"/>
  </environment>
  <process-type id="Custom" module-id="CUSTOM">
    <process-set id="Custom" restart-on-death="true" numprocs="1">
      <module-data>
        <category id="start-parameters">
          <data id="start-executable" value="Your start executable here" />
          <data id="start-args" value="Your start args here" />
        </category>
        <category id="stop-parameters">
          <data id="stop-executable" value="Your stop executable here" />
          <data id="stop-args" value="Your stop args here" />
        </category>
        <category id="restart-parameters">
          <data id="restart-executable" value="Your restart executable here"/>
          <data id="restart-args" value="Your restart args here" />
        </category>
        <category id="ping-parameters">
          <data id="ping-type" value="http" />
          <data id="ping-url" value="/your/ping/url" />
          <data id="ping-host" value="abc.company.com" />
          <data id="ping-port" value="7777" />
          <data id="ping-limit" value="3" />
          <data id="ping-timeout" value="300" />
        </category>
        <category id="ready-parameters">
          <data id="use-ping-for-ready" value="false" />
        </category>
      </module-data>
    </process-set>
  </process-type>
</ias-component>
```

Pinging with a script can be configured as shown in [Example 15–2](#).

#### **Example 15–2 Ping Type Script**

```
<category id="ping-parameters">
```

```

<data id="ping-type" value="script" />
<data id="script-executable" value="Ping executable here" />
<data id="script-args" value="Ping arguments here " />
</category>

```

You can use ping (when available) for determining the readiness of a process. This indicates that OPMN needs confirmation that a managed process has started successfully after creation. Processes can inform OPMN of their ready status in various ways. The custom module enables these processes to communicate readiness through ping. If you configure ping for a custom process, you can also use this mechanism to determine if the process is ready. You can choose not to configure any mechanism for determining readiness in which case the custom module just assumes that the process started successfully.

---

**Note:** The ready ping, if configured, is created soon after the process is created. If the process takes a while to initialize and respond to pings, then using ping for determining readiness is not appropriate. This is because if the process does not respond to the "ready ping", OPMN will determine that the process did not start correctly and stop it.

---

## 15.4 Custom Process Attribute Descriptions

This section describes the attributes that are specific for a custom process. This section also provides attribute descriptions of the attributes.

The custom process attributes are described with the following format:

- **Title:** This is the attribute name and value being defined. For example, `id="Custom"`.
- **Required:** This field defines whether or not the attribute is required in the component definition.
- **Default:** This defines the default value assigned to the attribute. The default value appears in the installed version of the `opmn.xml` file or is assigned internally if the attribute is not present.
- **Valid values:** If applicable, this field defines the valid values for the attribute. For example, `custom`.
- **Path:** This field defines in which elements the attribute can appear. For example, `ias-component`.

### `id="Custom"`

Required: true

Default: none

Valid values: Any `id` of your choice

Path: `ias-component`

Path: `ias-component/process-type`

Path: `ias-component/process-type/process-set`

This `id` is required and can be any name you choose. The `id` cannot be a duplicate of existing names.

### `module-id="CUSTOM"`

Required: true

Default: none

Valid values: The same as the `module-id` specified in [Section 15.1, "Custom Process Module Configuration"](#).

Path: `ias-component/process-type`

The `module-id` associates the process with a module. For Custom processes, this `id` has to match the `module-id` specified in Process Module Configuration for the Custom module.

#### **id="start-parameters"**

Required: true

Default: none

Path: `ias-component/process-type/process-set/module-data/category`

The `start-parameters` category contains child elements specifying the start executable and start arguments.

#### **id="start-executable"**

Required: true

Default: none

Valid values: a valid executable to run

Path:

`ias-component/process-type/process-set/module-data/category/data`

This data element specifies the name of the executable to be started.

#### **id="start-args"**

Required: false

Default: none

Valid values: Valid arguments to the executable specified by `start-executable` data element.

Path:

`ias-component/process-type/process-set/module-data/category/data`

The value of this data element should be a string containing all the arguments for the start executable. Multiple data elements with this `id` should not be specified.

#### **id="stop-parameters"**

Required: false

Default: none

Path: `ias-component/process-type/process-set/module-data/category`

The `stop-parameters` category contains child elements specifying the stop executable and stop arguments. If this category is not configured, OPMN stops the process with the kill command.

#### **id="stop-executable"**

Required: false

Default: none

Path:

ias-component/process-type/process-set/module-data/category/  
data

This data element specifies the name of the executable to be used for stopping the process.

**id="stop-args"**

Required: false

Default: none

Path:

ias-component/process-type/process-set/module-data/category/  
data

The value of this data element should be a string containing all the arguments to the stop executable. Multiple data elements with this `id` should not be specified.

**id="restart-parameters"**

Required: false

Default: none

Path: ias-component/process-type/process-set/module-data/  
category

The `restart-parameters` category contains child elements specifying the restart executable and restart arguments. This category needs to be configured if the process has an explicit restart command. In the absence of a restart command, a stop followed by the start command will be executed whenever the process needs to be restarted.

When restart data is specified, OPMN assumes that the process ID of the process remains the same after a restart. If there is no explicit restart command available for the process, a stop followed by a start is issued. In this scenario, a process ID change is acceptable.

**id="restart-executable"**

Required: false

Default: none

Valid values: A valid restart executable name

Path:

ias-component/process-type/process-set/module-data/category/  
data

This data element specifies the name of the executable to be used for restarting the process.

**id="restart-args"**

Required: false

Default: none

Valid values: valid arguments to the restart executable

Path:

ias-component/process-type/process-set/module-data/category/  
data

The value of this data element should be a string containing all the arguments to the restart executable. Multiple data elements with this `id` should not be specified.

**id="ping-parameters"**

Required: false

Default: none

Path: `ias-component/process-type/process-set/module-data/category`

Custom processes that are pinged through the HTTP protocol must specify this category. This module data category consists of all the data required to perform such a ping.

**id="ping-type"**

Required: false

Default: none

Valid values: `http`, `script`

Path:

`ias-component/process-type/process-set/module-data/category/data`

Custom processes that wish to be pinged have to specify this module data.

**See Also:**

- [Example 15-1, "Custom Process Complete Configuration"](#)
- [Example 15-2, "Ping Type Script"](#)

**id="ping-url"**

Required: false

Default: `/`

Valid values: Any valid URL

Path:

`ias-component/process-type/process-set/module-data/category/data`

This data element is used to specify the URL at which the process will be pinged. The listed parameters are used for HTTP pings.

**id="ping-host"**

Required: false

Default: none

Valid values: A valid hostname to which a custom process is bound.

Path:

`ias-component/process-type/process-set/module-data/category/data`

This data element is used to specify the host name to which a custom process is bound. If this data is not specified, pinging will not be performed. If an invalid hostname is specified, the process-set will be disabled.

**id="ping-port"**

Required: false

Default: none

Valid values: A valid port at which a custom process is listening for HTTP requests

Path:

`ias-component/process-type/process-set/module-data/category/`

data

The port at which a custom process is listening. If this data is not specified, pinging will not be performed. If an invalid port is specified, the `process-set` will be disabled.

**id="ping-limit"**

Required: false

Default: 3

Valid values: Any reasonable value that reflects the tolerance that OPMN should have for failed pings. This tolerance is used by OPMN to determine when the process should be declared unresponsive and restarted.

Path: `ias-component/process-type/process-set/module-data/category/data`

This `module data` element defines the tolerance for failed pings. After the number of ping failures reaches this limit, the process is deemed unresponsive and restarted by OPMN.

**id="ping-timeout"**

Required: false

Default: 300 seconds

Valid values: Any reasonable timeout value

Path:

`ias-component/process-type/process-set/module-data/category/data`

The timeout value specified with this data element is used as the maximum time OPMN will wait for a ping response. If a response is not obtained within the timeout period, the ping attempt will be considered a failure.

**id="script-executable"**

Required: false

Default: none

Valid values: A valid script executable

Path: `ias-component/process-type/process-set/module-data/category/data`

This data element specifies the name of the executable to be used for pinging the process. An exit value of 0 from this executable is considered success. All other values indicate a ping failure.

**id="script-args"**

Required: false

Default: none

Valid values: valid arguments to the ping executable

Path: `ias-component/process-type/process-set/module-data/category/data`

The value of this data element should be a string containing all the arguments to the ping executable. Multiple data elements with this id should not be specified.

**id="ready-parameters"**

Required: false

Default: none

Path: `ias-component/process-type/process-set/module-data/category`

The `module data category` to indicate if pinging should be used to determine that a custom process started successfully.

**id="use-ping-for-ready"**

Required: false

Default: false

Valid values: `true` or `false`

Path: `ias-component/process-type/process-set/module-data/category/data`

The value of this data element determines if pinging should be used to determine if a process is available.



---

# OPMN Troubleshooting

This chapter describes some troubleshooting tips for OPMN. It features the following topics:

- [Section A.1, "Problems and Solutions"](#)
- [Section A.2, "Diagnosing OPMN Problems"](#)
- [Section A.3, "Need More Help?"](#)

## A.1 Problems and Solutions

This section describes some of the common problems encountered when using OPMN. It features the following topics:

- [Section A.1.1, "Oracle Application Server Process Does Not Start"](#)
- [Section A.1.2, "Determining if Oracle Application Server Processes are Dying or Unresponsive"](#)
- [Section A.1.3, "opmnctl Command Execution Times Out"](#)
- [Section A.1.4, "Oracle Application Server Component Automatically Restarted by OPMN"](#)
- [Section A.1.5, "Unexpected opmnctl start Behavior"](#)
- [Section A.1.6, "Disabled Element in the opmn.xml File"](#)
- [Section A.1.7, "Management of Oracle9iAS Release 2 \(9.0.2 and 9.0.3\) Instances"](#)
- [Section A.1.8, "Unable to Start OC4J"](#)
- [Section A.1.9, "Unable to Stop Component"](#)
- [Section A.1.10, "globalInitNLS Error"](#)
- [Section A.1.11, "Start Remote Hosts of a Cluster Independently"](#)
- [Section A.1.12, "Log Loader Not Started"](#)
- [Section A.1.13, "dcm-daemon Not Started"](#)
- [Section A.1.14, "SSL Handshake Failed"](#)
- [Section A.1.15, "OPMN Start Up Consumes CPU Processing Capability"](#)
- [Section A.1.16, "Error Messages During Start-up of OPMN"](#)

## A.1.1 Oracle Application Server Process Does Not Start

### Problem

Unable to start an Oracle Application Server process using OPMN.

### Solution

Try the following if you are unable to start an Oracle Application Server process using OPMN:

- Verify and if necessary, correct, the command input. Confirm the spelling and choice of option for the command you are entering.

---

---

**Note:** Do not use command line scripts or utilities from previous versions of Oracle9iAS for starting OPMN or Oracle Application Server components.

---

---

- Review the standard out output log for the Oracle Application Server process. Output from the process console is located in the `ORACLE_HOME/opmn/logs` directory. For example, the standard output log for OracleAS Web Cache may be `WebCache~WebCacheAdmin~1`.
- Verify the dependency requirements for the Oracle Application Server process you are attempting to start. For example, the Oracle Application Server process you want to start may require that the Oracle Application Server Metadata Repository be up and running.
- Verify the element values for the Oracle Application Server component in the `opmn.xml` file. Use the `opmnctl validate` command to verify configuration of `opmn.xml` file. You may have mis-configured the `opmn.xml` for the Oracle Application Server component you are attempting to start.

## A.1.2 Determining if Oracle Application Server Processes are Dying or Unresponsive

### Problem

Your Oracle Application Server processes are dying or unreachable.

### Solution

If your Oracle Application Server processes are dying or unreachable:

- Look at the `ORACLE_HOME/opmn/logs/ipm.log` for Oracle Application Server processes at level of 4 or higher. Look for `process crashed` or `process unreachable` messages. OPMN automatically restarts Oracle Application Server processes that die or become unresponsive.
- Create event scripts for any pre-stop or post-crash events. The event scripts could be used to create a specific log file or send you an email about a failure.

**See Also:** [Section A.2.5, "Troubleshooting with Event Scripts"](#)

- Review the Oracle Application Server component specific output in the `ORACLE_HOME/opmn/logs`.
- Review the Oracle Application Server component specific log file located in Oracle Application Server component directory.

---

**See Also:** [Section A.2.1, "OPMN log Files"](#)

- Use iHAT to view the actions of Oracle Application Server

**See Also:** [Section A.2.3, "iHAT"](#)

### A.1.3 opmnctl Command Execution Times Out

#### Problem

The time it takes to execute an `opmnctl` command is dependent on the type of Oracle Application Server process and available computer hardware. Because of this the time it takes to execute an `opmnctl` command may not be readily apparent.

The default start time out for OC4J is approximately five minutes. If an OC4J process does not start-up after an `opmnctl` command, OPMN will wait approximately an hour before timing out and aborting the request.

#### Solution

To verify successful execution of the `opmnctl` command, try the following:

1. Increase the `start` element `timeout` attribute for the component that is not starting. Set the timeout in the `opmn.xml` file at a level that will allow OPMN to wait for process to come up. This functionality is available with the `startproc` command which will start all the relevant processes configured in `opmn.xml`.
2. Check the `start` element in the `opmn.xml` file and change the `retry` attribute to a higher increment of time.
3. Look at the `ORACLE_HOME/opmn/logs/` for the Oracle Application Server process that is not starting.
4. Review the component-specific log file for the Oracle Application Server component that is not starting. For example, `ORACLE_HOME/discoverer/logs`.
5. Examine the `ORACLE_HOME/opmn/logs/ipm.log` for any indication of problems. Increase the log level in the file to obtain additional information.

**See Also:** [Chapter 3, "opmn.xml Common Configuration"](#)

6. Contact Oracle Technical Support.

If you are having difficulty with an Oracle Application Server instance that is part of a farm, review the `ons.log` file for the Oracle Application Server instance. The "attempting active connection init" message in the `ons.log` file indicates that there is another OPMN configured in the farm that is currently shut-down. OPMN tries to consistently connect to the shut-down OPMN. Use the listed steps to determine why the down OPMN is not running.

### A.1.4 Oracle Application Server Component Automatically Restarted by OPMN

#### Problem

An Oracle Application Server component is automatically restarted by OPMN.

#### Solution

If an Oracle Application Server component is automatically restarted by OPMN, try the following:

- Review the message for the Oracle Application Server component in the `ipm.log` file.
- Verify that the ping timeout for the Oracle Application Server component is sufficient. An Oracle Application Server component that receives a lot of activity may require an increase in the length of time for the timeout. Increase the ping timeout element in the Oracle Application Server component `opmn.xml` file.

### A.1.5 Unexpected opmnctl start Behavior

#### Problem

Occasionally, there is unexpected behavior when you use the `opmnctl start` command to start OPMN; either only OPMN is started or OPMN makes a best effort to start Oracle Application Server OPMN-managed processes. Typically, this unexpected behavior is due to turning-off or rebooting your computer without first shutting down OPMN. When you restart your computer, all OPMN-managed processes are started.

#### Solution

Oracle recommends that you shutdown OPMN before shutting down your computer. Use the `opmnctl stopall` command to stop OPMN and OPMN-managed processes.

On the Microsoft Windows operating system, you can use the Windows services control panel to stop OPMN and OPMN-managed processes.

---

---

**Note:** OPMN keeps a record on disk of the expected status of the processes it manages. If a computer goes down while OPMN is running, upon restart OPMN will use the information cached on disk and make a best effort attempt to automatically restart all processes that were running at the time the system went down. This may catch some users off guard who start only OPMN and notice that processes managed by OPMN have also been started even though an explicit request to start those processes has not been issued. You can suppress this automatic process recovery by removing all files located in the `ORACLE_HOME/opmn/logs/states` directory before attempting to start OPMN.

The states directory and its contents should not be modified by the user if OPMN or any process managed by OPMN is running. Oracle recommends not modifying the `/states` directory.

---

---

### A.1.6 Disabled Element in the opmn.xml File

#### Problem

Unable to start an Oracle Application Server process.

#### Solution

If you are unable to start an Oracle Application Server process, check if an element in the Oracle Application Server `opmn.xml` file is disabled. If an element in the `opmn.xml` file is disabled OPMN will generate an output message of "Missing" or "Disabled".

---

## A.1.7 Management of Oracle9iAS Release 2 (9.0.2 and 9.0.3) Instances

### Problem

By default, the SSL element in the Oracle Application Server 10g Release 2 (10.1.2) `opmn.xml` file is enabled; however, the SSL element is not enabled in the `opmn.xml` file for Oracle9iAS Release 2 (9.0.2 and 9.0.3).

### Solution

If you have an Oracle Application Server farm containing Oracle9iAS Release 2 (9.0.2 and 9.0.3) instances you must enable the SSL element in the Release 2 `opmn.xml` file. Refer to the *Oracle9i Application Server Administrator's Guide*, Release 2 (9.0.2) for the steps to enable the SSL element.

## A.1.8 Unable to Start OC4J

### Problem

If you have multiple Oracle Application Server installations on one host and you start them at the same time (for example, to start a cluster), OPMN may become unresponsive. You may receive an error message such as:

```
"failed to restart a managed process after the maximum retry limit"
```

This may occur when two Oracle homes on the same host use the same port ranges for RMI, JMS, and AJP ports. An OC4J instance in one Oracle home is trying to use the same port as an OC4J instance in a different Oracle home.

Port allocation for all OC4J instances within Oracle Application Server is controlled by OPMN; there can be overlapping port ranges within a single `opmn.xml` file. However, when two OPMN processes on a host start at the same time, there is no coordination between them on port usage.

### Solution

To coordinate port usage, assign unique port ranges to each Oracle home. The OPMN process in one Oracle home and the OPMN in a different Oracle home will not attempt to use the same port numbers when assigning OC4J ports, and will not attempt to bind to the same port.

It is also recommended that you increase the maximum number of retries for starting OC4J instances. If you have identical port ranges in two Oracle homes and increase the number of times OPMN attempts to restart a process, OPMN will eventually select a port that works. This technique ultimately does not eliminate the problem, because there is the possibility that OPMN will not find a port that works in the number of port connection attempts that you have specified in the `opmn.xml` file.

**See Also:** Section 3.6.4, "Resolving OC4J Errors When Starting Multiple Instances Simultaneously on the Same Host" in the *Oracle Application Server Administrator's Guide*.

## A.1.9 Unable to Stop Component

### Problem

If you are unable to stop Oracle Application Server components or OPMN-managed processes using the `opmnctl stop` or `opmnctl stopall` commands, the

component or process was most likely not started using OPMN. The component or process might have been started using a startup script or utility.

**Solution**

Oracle Application Server components and OPMN-managed processes should never be started or stopped manually. Do not use command line scripts or utilities from previous versions of Oracle Application Server for starting and stopping Oracle Application Server components.

Use the Application Server Control Console and the `opmnctl` command line utility to start or stop Oracle Application Server components and OPMN-managed processes.

---

---

**Note:** Oracle Application Server start and stop scripts for OPMN and OPMN-managed processes are available on the Oracle Application Server Metadata Repository Creation Assistant installation disc.

---

---

**See Also:** [Chapter 2, "Using OPMN"](#)

### A.1.10 globalInitNLS Error

**Problem**

You may receive a `globalInitNLS` error when executing the `opmnctl` command. The following error message is displayed:

```
"globalInitNLS: NLS boot file not found or invalid -- default linked-in boot block  
used XML parser init: error 201."
```

**Solution**

This error occurs when the `ORA_NLS33` environmental variable is set. This environmental variable should not be set.

### A.1.11 Start Remote Hosts of a Cluster Independently

**Problem**

Starting a cluster of remote hosts using Application Server Control Console will result in an unknown status. This occurs because ONS is bound to the local host IP address and it is not reachable from remote hosts.

**Solution**

Oracle recommends starting each member of the cluster independently to effectively monitor and obtain the status from remote hosts. Additionally, make sure ONS is not bound to local host IP address.

### A.1.12 Log Loader Not Started

**Problem**

If you run the `opmnctl status` command following the successful execution of the `opmnctl startall` or `opmnctl startproc` command, the status output indicates Log Loader status as down.

---

### Solution

By default, the Log Loader element in the `opmn.xml` file is not configured to start when you execute the `opmnctl startall` or `opmnctl startproc` command.

To start Log Loader, perform the following steps:

1. Select the Logs link on the Application Server Control Console page.
2. From the View Logs page select the **Search Log Repository** link.
3. On the View Logs page, click **Log Loader**.
4. On the Log Loader page, click Start.
5. On the confirmation page select either **Cancel**, **Start**, or **Start and Load Existing Logs**.

On the confirmation page you can:

- select **Cancel** to cancel use of Log Loader
- select **Start** to start the Log Loader and *not* load current contents of the logs. The Log Loader will only load log messages that are written to the logs after the log loader is started.
- select **Start and Load Existing Logs** to start the Log Loader and load current contents of all log files.

You can also start and stop Log Loader using the following commands:

- `prompt > opmnctl startproc ias-component=LogLoader`
- `prompt > opmnctl stopproc ias-component=LogLoader`

The `opmnctl startproc ias-component=LogLoader` command is equivalent to the **Start and Load Existing Logs** selection available on Application Server Control Console.

**See Also:** *Oracle Application Server Administrator's Guide* for more information about starting and stopping Log Loader.

## A.1.13 dcm-daemon Not Started

### Problem

After you run the `opmnctl startall` or `opmnctl startproc` command and then run the `opmnctl status` command, the status output indicates that the `dcm-daemon` is in Down status.

### Solution

By default, the `dcm-daemon` element in the `opmn.xml` file is not configured to start when you execute the `opmnctl startall` or `opmnctl startproc` command. The `dcm-daemon` is started automatically by any DCM client tool, either the `dcmctl` command or Application Server Control Console, during initialization phase. Typically, there is no need to start the `dcm-daemon` explicitly.

If you want to explicitly start the `dcm-daemon`, use the following command:

```
prompt > opmnctl startproc ias-component="dcm-daemon"
```

You may be required to start the `dcm-daemon` explicitly in certain configurations. For example, if you are performing an Oracle Application Server cluster wide DCM operation. To deploy an application on the cluster, the `dcm-daemon` needs to be

running on all Oracle Application Server instances in the cluster. If you execute the `dcmctl` command on a local Oracle Application Server instance, it will start the `dcm-daemon` on the local instance but it will not start the `dcm-daemon` on the other instances in the cluster.

### A.1.14 SSL Handshake Failed

#### Problem

You receive a SSL handshake failed error message similar to the following in your `ons.log` file:

```
"Connection 0.169.254.25.129.6203 SSL handshake failed"
```

in your `ons.log` file.

#### Solution

The SSL handshake failed error message occurred most likely because the SSL enable attribute of ONS in the `opmn.xml` file does not have the same value. The SSL enable values of either true or false must be consistent for all OPMN servers in the application server farm.

### A.1.15 OPMN Start Up Consumes CPU Processing Capability

#### Problem

On some computers, when OPMN starts up, it consumes large amounts of CPU processing capability. This can vary from approximately 50% to 60% of your computer's CPU processing capabilities. In affected computers, the OPMN CPU processing consumption will continue until OPMN is shutdown.

#### Solution

The following are some possible causes for the excessive CPU processing consumption:

- the installation environment used multi-byte text character sets such as Japanese.
- the ONS topology is mis-configured.

For example, if the `ons.conf` files for Oracle Application Server instances across the Oracle Application Server Farm may get out of sync. The contents of `ons.conf` file must be the same for each Oracle Application Server instance in the Farm. If the `ons.conf` file contents in each Oracle Application Server instance are not the same, atypical topology loops can occur that multiply each notification managed by ONS by a factor of two.

- the computer has multiple network interface cards (NICs)

The hostname or IP value configured for OPMN in the `opmn.xml` file does not match the corresponding entry in the `ons.conf` file. OPMN reconnects to itself over and over, thereby increasing CPU usage.

### A.1.16 Error Messages During Start-up of OPMN

#### Problem

When trying to start OPMN using the `opmnctl start` or `opmnctl startall` commands you receive the following error messages:



---

```
pingwait exits with 1220384
```

or

```
pingwait exits with 1220396
```

These error messages are generated when there are syntax errors in the `ORACLE_HOME/opmn/conf/opmn.xml` that need to be corrected.

### Solution

If you encounter these error messages do the following:

- run the following command (with the complete directory path to the `opmn.xml` file):

```
prompt > opmnctl validate opmn.xml
```

- remove all empty tags from the `opmn.xml` file.
- run the following command to update the repository configuration:

```
prompt > dcmctl updateconfig
```

## A.2 Diagnosing OPMN Problems

There are several methods for troubleshooting any problems you may have using OPMN:

- [Section A.2.1, "OPMN log Files"](#)
- [Section A.2.2, "opmnctl debug"](#)
- [Section A.2.3, "iHAT"](#)
- [Section A.2.4, "Oracle Enterprise Manager 10g Application Server Control Console"](#)
- [Section A.2.5, "Troubleshooting with Event Scripts"](#)
- [Section A.2.6, "opmn.xml Environment Variables"](#)

### A.2.1 OPMN log Files

The OPMN log files enable you to troubleshoot difficulties you might have in execution and use of OPMN and OPMN-managed processes. OPMN and OPMN-managed processes generate log files during processing. You can review the following generated log files to verify successful or unsuccessful execution of an OPMN command:

- `ORACLE_HOME/opmn/logs`: The standard output and standard error of OPMN managed processes. These files are also sometimes referred to as "console logs". OPMN creates a log file for each component and assigns a unique concatenation of the Oracle Application Server component with a number. For example, the standard output log for OracleAS Web Cache may be `WebCache~WebCacheAdmin~1`. When a process terminates and is replaced by a new process, console log output from the previous process is preserved and the replacement process appends to the end of the console log file. The process specific console logs are the first and best resource for investigating problems related to starting and stopping components.

- `ORACLE_HOME/opmn/logs/ipm.log`: Review the error codes and messages that are shown in the `ipm.log` file. The PM portion of OPMN generates and outputs the error messages in this file. The `ipm.log` file tracks command execution and operation progress. The level of detail that gets logged in the `ipm.log` can be modified by configuration in the `opmn.xml` file. Refer to [Chapter 3, "opmn.xml Common Configuration"](#) for examples of debug levels.
- `ORACLE_HOME/opmn/logs/ons.log`: Use the `ons.log` file to debug the ONS portion of OPMN or for early OPMN errors. The ONS portion of OPMN is initialized before PM, and so errors that occur early in OPMN initialization will show up in the `ons.log` file.
- `ORACLE_HOME/opmn/logs/opmn.log`: The `opmn.log` file contains output generated by OPMN when the `ipm.log` and `ons.log` files are not available. Typically, the only output written to the `opmn.log` file will be the exit status of a child OPMN process. A status code of 4 indicates a normal reload of OPMN. All other status codes indicate an abnormal termination of the child OPMN process.
- `ORACLE_HOME/opmn/logs/service.log`: (Microsoft Windows only). The `service.log` displays any error messages generated by OPMN services while interacting with service control manager.

## A.2.2 opmnctl debug

Use the `opmnctl debug` command to verify the status of an Oracle Application Server process and whether any actions are pending. This command generates output that can be used in conjunction with contact to your local Oracle support to diagnose your OPMN problem.

The syntax for the `opmnctl debug` command is:

```
opmnctl [<scope>] debug [comp=pm|ons] [interval=<secs> count=<num>]
```

where `@scope` is the optional scope for the request.

Output is generated following execution of the `opmnctl debug` command. Oracle recommends that you contact Oracle support to use the generated output to assist in diagnosis of your problem.

The attributes (`<attr>`) name for this command are either `comp`, `interval`, or `count`. The value for `comp` can be either `ons` or `pm`, representing ONS and PM, respectively. If `comp` is not specified, then both `ons` and `pm` debug information is reported. For example, the following command outputs debug information for ONS.

```
prompt > opmnctl debug comp=ons
```

You can specify the interval in seconds and number of requests sent to OPMN to assist in the debugging process. The values of `<interval>` and `<count>` must always be specified together. Values for them should be integers greater than 0. For example, the following command, outputs debug information at an interval of 5 seconds 3 times.

```
prompt > opmnctl debug comp=pm interval=5 count=3
```

Contact your local Oracle support to assist you in using the `opmnctl debug` command to diagnose your OPMN problem.

## A.2.3 iHAT

Use the iHAT tool to provide a real time, graphical interface view of your enterprise. iHAT displays all Oracle Application Server processes managed by one or more

---

OPMN servers including useful performance metrics about each process as obtained from DMS. The snapshot of the system is updated continuously at a configurable interval.

To download iHAT, visit the OTN:

<http://www.oracle.com/technology/membership>

## A.2.4 Oracle Enterprise Manager 10g Application Server Control Console

Application Server Control Console provides a graphical interface that enables diagnosis of Oracle Application Server components in your network and enterprise. Application Server Control Console features a log page. The log page enables you to view all of the Oracle Application Server log files in one place and trace problems across multiple log files. Application Server Control Console uses an API that contacts OPMN.

You can use Application Server Control Console to enable or disable Oracle Application Server components: You can disable components so they do not start when you start an Oracle Application Server instance.

**See Also:** *Oracle Application Server Administrator's Guide*

## A.2.5 Troubleshooting with Event Scripts

You can create your own event scripts that record Oracle Application Server process event activities. You can create a script that records events prior to the start or stop of Oracle Application Server processes, as well as an unscheduled system crash.

Refer to the `<event-scripts>` element description in [Chapter 3, "opmn.xml Common Configuration"](#).

[Example A-1](#) shows a pre-start event script.

### **Example A-1 Pre-start Event Script**

```
#!/bin/sh
echo
echo =====
echo ===== PRE-START EVENT SCRIPT =====
echo =====

timeStamp="N/A"
instanceName="N/A"
componentId="N/A"
processType="N/A"
processSet="N/A"
processIndex="N/A"
stderrPath="N/A" # not available w/pre-start unless part of restart
stdoutPath="N/A" # not available w/pre-start unless part of restart
reason="N/A"
pid="N/A" # only available with pre-stop, post-crash
startTime="N/A" # only available with pre-stop, post-crash

while [ $# -gt 0 ]; do
  case $1 in
    -timeStamp)   timeStamp=$2; shift;;
    -instanceName) instanceName=$2; shift;;
    -componentId) componentId=$2; shift;;
    -processType) processType=$2; shift;;
```

```

        -processSet)    processSet=$2; shift;;
        -processIndex) processIndex=$2; shift;;
        -stderr)       stderrPath=$2; shift;;
        -stdout)       stdoutPath=$2; shift;;
        -reason)       reason=$2; shift;;
        -pid)          pid=$2; shift;;
        -startTime)    startTime=$2; shift;;
        *) echo "Option Not Recognized: [$1]"; shift;;
    esac
    shift
done

echo timeStamp=$timeStamp
echo instanceName=$instanceName
echo componentId=$componentId
echo processType=$processType
echo processSet=$processSet
echo processIndex=$processIndex
echo stderr=$stderrPath
echo stdout=$stdoutPath
echo reason=$reason
echo pid=$pid
echo startTime=$startTime

```

---

**Note:** The pre-start event script example, [Example A-1](#), will not work for the Microsoft Windows operating system; however, you can create a script, with a `.bat` suffix, with similar functionality.

Use the full path to the `.bat` file when adding the necessary configuration information to the `opmn.xml` file,.

---

## A.2.6 opmn.xml Environment Variables

The environment variable used to launch OPMN server is not inherited by the Oracle Application Server process started by OPMN server. OPMN sets the environment variables at the `ias-instance` level, with the values extracted either from the `ias-instance` configuration or from the OPMN run time environment.

**See Also:** [Chapter 3, "opmn.xml Common Configuration"](#)

## A.3 Need More Help?

You can find more solutions on Oracle *MetaLink* <http://metalink.oracle.com>. If you do not find a solution for your problem, log a service request.

**See Also:**

- *Oracle Application Server Release Notes*, available on the Oracle Technology Network:  
<http://www.oracle.com/technology/documentation/>

---

# Index

## A

- ajp id, 5-3, 8-2
- ajp ports, 8-3
- Application Server Control, 1-2, 1-5, 1-9, A-11
- application-parameters id, 13-3
- async, 1-12, 1-14
- attribute name, 1-12
- attribute syntax, 1-14

## B

- BAM complete configuration, 14-2
- BAM Engine module-id, 14-4
- BAM id, 14-3
- BAM minimum configuration, 14-1
- BAM process module, 14-1
- BAM Properties File id, 14-3
- batch id, 10-7

## C

- cache-timeout attribute, 3-18
- category element, 3-10
- child process, 1-24
- clu, 1-18
- cluster syntax, 1-13
- cmp, 1-18
- command, 1-8
- command definitions
  - opmnctl, 1-13
- complete configuration
  - BAM, 14-2
  - custom process, 15-1
  - DCM Daemon, 13-1
  - Log Loader, 12-2
  - OC4J, 5-1
  - Oracle HTTP Server, 4-2
  - Oracle Internet Directory, 7-2
  - OracleAS Discoverer, 11-2
  - OracleAS Port Tunnel, 8-1
  - OracleAS Web Cache, 6-2
  - OracleAS Wireless, 9-5
- config-file id, 4-3, 5-5
- configset id, 7-5
- connect id, 7-4
- count, A-10
- cpu, 1-18

- custom id, 15-3
- custom module
  - ping, 15-2
- CUSTOM module-id, 15-3
- custom process
  - complete configuration, 15-1
  - minimum configuration, 15-1
- custom process module, 15-1

## D

- data element, 3-10
- database element, 3-15
- datafeeder module-id, 9-11
- db-connect-info element, 7-3
- DCM, 1-24
- DCM Daemon
  - complete configuration, 13-1
  - minimum configuration, 13-1
  - process module, 13-1
- dcm-daemon id, 13-2
- DCMDaemon module-id, 13-2
- default start time, A-3
- dependencies element, 3-14
- DISCO\_DIR id, 11-3
- Disco\_SessionServer module-id, 11-3
- Discoverer id, 11-3
- dmsdump
  - opmnctl, 1-8

## E

- edgeserver module-id, 9-12
- environment element, 3-12
- event script
  - post-crash, 1-22
  - pre-start, 1-21
  - pre-stop, 1-22
- event script arguments, 3-22
- event scripts, 1-21, A-11
- event-scripts element, 3-22
- example elements, 3-1

## F

- farm syntax, 1-14
- fmt option, 1-17
- fmtlist, 1-17

- fmtlist syntax, 1-17
- form factor key, 1-23
- .formfactor file., 1-22
- fsep option, 1-17

## G

- general-parameters id, 10-6

## H

- help
  - opmnctl, 1-8, 1-20
- HTTP ping, 15-2
- HTTP\_Server id, 4-3

## I

- ias-component, 1-11, 1-12
- ias-component element, 3-14
- ias-instance attribute, 3-20
- ias-instance element, 3-11
- IASPT id, 8-2
- IASPT module-id, 8-2
- idx, 1-18
- iHAT, 1-24, A-10
- implicit, OPMN-managed components, 1-4
- ins, 1-18
- instance syntax, 1-13
- instance value, 7-5
- interval, 11-4, A-10
- ipaddr element, 3-3
- ipm.log, A-2, A-4, A-10

## J

- jar-file id, 13-3
- java options id, 5-4
- java-bin id, 5-5, 9-13
- justification, 1-18

## L

- l option, 1-17
- LD\_LIBRARY\_PATH id, 11-3
- local instance
  - starting, 2-1
  - stopping, 2-1
- local port, 1-24
- location module-id, 9-11
- log files
  - OPMN, A-9
- Log Loader
  - complete configuration, 12-2
  - minimum configuration, 12-1
  - process module, 12-1
- log page, A-11
- log-file element, 3-4, 3-6
- LOGLDR module-id, 12-3
- LogLoader id, 12-3
- logloaderd id, 12-3

## M

- managed-process element, 3-20
- max\_retry, 1-19
- mem, 1-18
- messaging module-id, 9-11
- minimum configuration
  - custom process, 15-1
  - DCM Daemon, 13-1
  - Log Loader, 12-1
  - OC4J, 5-1
  - Oracle HTTP Server, 4-2
  - Oracle Internet Directory, 7-1
  - OracleAS Discoverer, 11-1
  - OracleAS Port Tunnel, 8-1
  - OracleAS Web Cache, 6-1
  - OracleAS Wireless, 9-1
- minimum configuraton
  - BAM, 14-1
- mode, 1-12
- module, 3-8
- module element, 3-8
- module-data element, 3-9
- module-id element, 3-11
- modules
  - PM, 1-4
- Monitor Framework process set id, 14-6
- multiple instance
  - starting, 2-2
  - stopping, 2-3

## N

- noheaders option, 1-17
- no-reverseping-failed-ping-limit id, 4-5, 5-6
- notification module-id, 9-11
- notificationeventcollector module-id, 9-11
- notification-server element, 3-3
- numprocs, 4-3
- numprocs attribute, 4-3, 6-3
- numprocs element, 4-3

## O

- oad-extra-options, 11-6
- oad-java-options, 11-6
- oadutil-java-options id, 11-7
- OC4J
  - complete configuration, 5-1
  - minimum configuration, 5-1
  - process module, 5-1
- OC4J element, 5-2
- OC4J id, 5-2
- OC4J module-id, 5-3
- oc4j-options id, 5-5
- OHS module id, 4-3
- OHS2 module id, 4-5
- OID element, 3-16
- OID id, 7-3
- OID module-id, 7-3
- oidmon-parameters id, 7-4
- ONS, 3-3

- ONS (Oracle Notification Server), 1-3
- ons.log, A-3, A-10
- OPMN, 1-1, 1-2, 1-4
  - functionality, 1-1
  - log files, A-9
- OPMN daemon, 1-9, 1-12
- opmn element, 3-3
- OPMN local listener, 1-22
- opmnctl, 1-2, 1-7, 1-8, 1-9
  - help, 1-8, 1-20
  - options, 1-17
  - ping, 1-19
  - process control commands, 1-11
  - quick reference, 1-8
  - stopall, 1-9
  - syntax, 1-7
  - usage, 1-10, 1-11
  - validate, 1-21
- opmnctl attributes, 1-14
- opmnctl command definitions, 1-13
- opmnctl commands, 1-8
- opmnctl debug, A-10
- opmnctl dmsdump, 1-8
- opmnctl help, 1-8
- opmnctl ping, 1-8
- opmnctl reload, 1-8, 1-10, 1-11
- opmnctl restartproc, 1-8, 1-12
- opmnctl shutdown, 1-8, 1-10
- opmnctl start, 1-8, 1-9
- opmnctl startall, 1-8, 1-9
- opmnctl startproc, 1-8, 1-12
- opmnctl status, 1-8
- opmnctl status syntax, 1-16
- opmnctl stopall, 1-8
- opmnctl stopproc, 1-8, 1-10, 1-11, 1-12
- opmnctl usage, 1-8
- opmnctl usage status, 1-16
- opmnctl usage syntax, 1-20
- opmnctl validate, 1-8, A-2
- opmnctl verbose syntax, 1-15
- opmn.xml file, 1-3, 1-5
- option
  - fmt, 1-17
  - fsep, 1-17
  - l, 1-17
  - noheaders, 1-17
  - rsep, 1-17
- options, 1-8
  - opmnctl, 1-17
- Oracle Application Server Integration Business
  - Activity Monitoring, 14-1
- Oracle Enterprise Manager Application Server
  - Control, 1-5
- Oracle HTTP Server
  - complete configuration, 4-2
  - minimum configuration, 4-2
  - process module, 4-1, 4-6
- Oracle Internet Directory
  - complete configuration, 7-2
  - minimum configuration, 7-1

- process module, 7-1
- Oracle Notification Server see ONS
- Oracle Process Manager Modules see PM
  - Modules, 1-4
- Oracle Process Manager see PM, 1-3
- Oracle wallet, 3-6
- OracleAS Discoverer
  - complete configuration, 11-2
  - minimum configuration, 11-1
  - process module, 11-1
- OracleAS Port Tunnel
  - complete configuration, 8-1
  - minimum configuration, 8-1
  - process module, 8-1
- OracleAS Web Cache
  - complete configuration, 6-2
  - minimum configuration, 6-1
  - process module, 6-1
- OracleAS Wireless
  - complete configuration, 9-5
  - minimum configuration, 9-1
  - process module, 9-1
- OSSO element, 3-18

## P

- path attribute, 3-4
- performance module-id, 9-11
- pid, 1-18
- ping
  - custom module, 15-2
  - opmnctl, 1-8, 1-19
- ping element, 3-26
- ping parameters category, 4-4
- ping timeout, A-4
- ping-host id, 15-6
- ping-limit id, 15-7
- ping-parameters id, 15-6
- ping-port id, 15-6
- ping-timeout id, 15-7
- ping-type id, 15-6
- ping-url id, 4-4, 15-6
- PM, 1-3
- PM Modules, 1-4
- PM modules, 1-5
- por, 1-18
- port element, 3-3, 3-27
- post-crash element, 3-24
- post-crash event script, 1-22
- PREFERENCE\_PORT id, 11-3
- pre-start event script, 1-21
- pre-stop element, 3-24
- pre-stop event script, 1-22
- process control command
  - opmnctl, 1-11
- process crashed, A-2
- process module
  - BAM, 14-1
  - custom, 15-1
  - DCM Daemon, 13-1
  - Log Loader, 12-1

- OC4J, 5-1
- Oracle HTTP Server, 4-1, 4-6
- Oracle Internet Directory, 7-1
- OracleAS Port Tunnel, 8-1
- OracleAS Web Cache, 6-1
- OracleAS Wireless, 9-1
- process unreachable, A-2
- process-manager element, 3-6
- process-modules element, 3-8
- process-set, 1-11, 1-12
- process-set element, 3-27
- process-set-id attribute, 3-20
- process-type, 1-11, 1-12
- process-type id attribute, 3-20
- prs, 1-18
- prt, 1-18
- publish-subscribe model, 1-3
- Put Path\_value here value, 10-6

## Q

- quick reference
  - opmnctl, 1-8

## R

- range attribute, 5-4
- range value, 8-3
- ready-parameters id, 15-7
- reload
  - opmnctl, 1-8, 1-10, 1-11
- remote instance
  - starting, 2-2
  - stopping, 2-2
- remote port, 1-24
- remote request, 1-24
- /reports/rwservlet/pingserver?start=auto id, 10-7
- ReportsServer1 id, 10-5
- restart element, 3-26
- restart-args id, 15-5
- restart-executable id, 15-5
- restart-on-death attribute, 6-3
- restart-parameters category, 4-4
- restart-parameters id, 5-5
- restartproc
  - opmnctl, 1-8, 1-12
- reverseping-failed-ping-limit element, 5-6
- reverseping-failed-ping-limit id, 4-5, 5-6
- reverseping-timeout id, 4-5, 5-6
- rmi id, 5-4
- rsep option, 1-17
- run-oidctl id, 7-4

## S

- scope, 1-7, 1-8, 1-13
- script ping, 15-2
- script-args id, 15-7
- script-executable id, 15-7
- SessionServer id, 11-3
- shutdown
  - opmnctl, 1-8, 1-10

- sleep id, 7-4
- SSL, 1-22, 3-17
- SSL element, 3-5, 3-18, A-5
- sta, 1-18
- standard out output log, A-2
- start
  - opmnctl, 1-8, 1-9
- start element, 3-25, A-3
- start order dependencies, 1-22
- startall
  - opmnctl, 1-8, 1-9
- start-args id, 15-4
- start-executable id, 15-4
- starting
  - local instance, 2-1
  - multiple instance, 2-2
  - remote instance, 2-2
- starting OPMN, 2-1
- startodisrv id, 7-6
- startoidldapd id, 7-5
- startoidrepld id, 7-5
- start-parameters category, 4-3
- start-parameters id, 5-4
- startproc
  - opmnctl, 1-8, 1-12
- statname, 1-18
- status
  - opmnctl, 1-8, 1-16
- stm, 1-18
- stop element, 3-25
- stopall
  - opmnctl, 1-8, 1-9
- stop-args id, 15-5
- stop-executable id, 15-4
- stop-parameters id, 5-5
- stopping
  - local instance, 2-1
  - multiple instance, 2-3
  - remote instance, 2-2
- stopproc
  - opmnctl, 1-8, 1-10, 1-11, 1-12
- sync, 1-12, 1-14
- Syntax
  - opmnctl ping, 1-19
- syntax, 1-14
  - attribute, 1-14
  - cluster, 1-13
  - farm, 1-14
  - fmtlist, 1-17
  - instance, 1-13
  - opmnctl, 1-7
  - opmnctl help, 1-20
  - opmnctl options, 1-17
  - opmnctl reload, 1-11
  - opmnctl restartproc, 1-12
  - opmnctl shutdown, 1-10
  - opmnctl start, 1-9
  - opmnctl startall, 1-9
  - opmnctl startproc, 1-12
  - opmnctl status, 1-16



- opmnctl stopall, 1-9
- opmnctl stopproc, 1-12
- opmnctl usage, 1-20
- opmnctl validate, 1-21
- opmnctl verbose, 1-15
- scope, 1-13
- syntax value, 1-14

## T

- targets.xml file, 8-2
- telnetdispatcher module-id, 9-12
- telnetserver module-id, 9-12
- timeout, 1-12, 1-14
- timeout attribute, 3-18, A-3
- timeout value, 1-9
- tnsnames.ora file, 7-3
- typ, 1-18

## U

- uid, 1-17, 1-18
- uniqueid, 1-12, 1-14, 1-17
- urlping-parameters id, 10-7
- usage
  - opmnctl, 1-8, 1-10, 1-11, 1-20
- use-ping-for-ready id, 15-8
- utm, 1-18

## V

- validate
  - opmnctl, 1-8, 1-21
- value, 1-14
- value syntax, 1-14
- variable element, 3-12
- verbose, 1-7
  - opmnctl, 1-15

## W

- wallet file, 1-24
- WebCache id, 6-2
- WebCache module-id, 6-3
- WebCacheAdmin id, 6-3
- width, 1-18
- wireless id, 9-10
- working-dir attribute, 3-22, 7-3

