

This FAQ addresses frequently asked questions relating to the Oracle Application Server 10g Release 3 (10.1.3.1.0) version of Oracle Process Manager and Notification Server (OPMN) and is divided into the following sections:

[General](#)

[Logging](#)

[Topology and DRM](#)

[Configuration](#)

## 1.0 General

### 1.1 What is OPMN?

Oracle Process Manager and Notification Server (OPMN) is installed and configured with every Oracle Application Server installation type and is essential for running Oracle Application Server. OPMN provides an integrated way to manage all Oracle Application Server components. OPMN consists of two main pieces the process manager and the notification server.

- Process manager is the centralized process management mechanism in Oracle Application Server and is used to manage all Oracle Application Server processes. The PM starts, restarts, stops, and monitors every process it manages. It also performs death-detection and automatic restart of the processes.
- Oracle Notification Server (ONS) is the transport mechanism for failure, recovery, startup, and other related notifications between components in Oracle Application Server.

### 1.2 What are some of the new features in OPMN?

OPMN has following new features:

- Infrastructure
  - Single Configuration File: The information that was stored in ons.conf and dcm.conf is now configured within opmn.xml
  - Logging Mechanism: Standard and debug messages are located in opmn.log, opmn.dbg and opmn.out files.
- Notification Server
  - IPv6 Support: ONS now supports the IPv6 network stack.
  - Dynamic Discovery: Connection topology is dynamically managed.
- Process Management
  - Applications Support: OPMN can now manage J2EE applications.
  - Dynamic Resource Management: Capability to describe a set of desired behaviors and take actions to achieve the desired results
  - Service Fail-over: Mechanism to specify a critical single (or limited number) process that must run somewhere in the cluster
  - Sequential Requests: Option to perform the request upon one process at a time
  - Progressive Request Report: Way to report the result of each part of an OPMN request as it completes

For details refer to the *Oracle Process Manager and Notification Server Administrator's Guide 10g Release 3 (10.1.3.1.0)*.

### 1.3 What algorithm does OPMN use to determine when to restart the process?

The OPMN logic for restarting a managed process (assuming no user requests to stop or restart the process) is a three pronged approach.

1. OS Process Check: Every 2 seconds OPMN queries the OS with the managed process id to see if it has terminated.
2. Forward Ping: Periodically, 20 seconds by default, OPMN sends a ping message to the managed process and expects a result within 20 seconds.
3. Reverse Ping: Every 20 seconds managed process sends OPMN a ping notification.

If check #1 fails OPMN always attempts to restart the managed process.

For check #2 and #3, if OPMN does not get a forward ping response, it flags the managed process as "DEAD" (i.e. unresponsive/unreachable). OPMN will continue to try and ping this "DEAD" process for max retry times. Max retry is the value of either of following two data elements:

- reverseping-failed-ping-limit : when reverse pings are being received, OR
- no-reverseping-failed-ping-limit : when reverse pings are also not being received (within the timeout period specified by reverseping-timeout data element).

If any forward ping succeeds, the process state is set back to "ALIVE" (and the internal count towards max retry is set back to 0). If the forward ping fails consecutively for max retry times, then OPMN will attempt to stop and restart the process.

### 1.4 Can OPMN manage J2EE applications?

Yes. J2EE applications are supported using the OPMN process management mechanism. There are automatic application state and metric updates. You can start, stop, restart, and check the status of your J2EE application using `opmnctl` commands. This allows much finer grained control for performing operations such as application upgrades or resetting of unresponsive applications.



## 2.0 Logging

### 2.1 I don't see `ipm.log` and `ons.log` files anymore. Where do OPMN standard and debug messages go?

OPMN and OPMN-managed processes generate log files during processing to enable you to troubleshoot difficulties you might have in execution of the process. In prior releases, standard and debug messages were located in either the `ipm.log` or `ons.log` files.

Standard and debug messages are located in the following files:

- `opmn.log`: contains standard international log messages, all standard OPMN log messages and messages for ONS and Oracle Process Manager (PM).
- `opmn.dbg`: contains OPMN debug log messages (English only) for ONS and PM.
- `opmn.out`: OPMN console log messages (stdout and stderr).

## 2.2 There used to be a level attribute in the <log-file> tag that I don't find anymore.

### How do I define log levels?

In 10.1.3.1.0, logging is configured by component codes rather than level codes. The logging messages contain the literal value based on logging levels rather than an integer value; for example: none, fatal, error, warn, notify, debug1, debug2, debug3, and debug4.

All OPMN log messages of non-debug type (none, fatal, error, warn and notify) for the ONS and PM components go in `opmn.log` file. If debugging is enabled, then all OPMN debug log messages (of type debug1, debug2, debug3 and debug4) are written into `opmn.dbg` file.

You can configure the following component codes for logged events for both log and debug:

- `internal`: a log for the common internal information for OPMN
- `ons`: a log for the ONS component information for OPMN
- `pm`: a log for the PM component information for OPMN

Both the `ons` and `pm` components consist of subcomponents that can also be configured. Refer to the *Oracle Process Manager and Notification Server Administrator's Guide 10g Release 3 (10.1.3.1.0)* for the list of `ons` and `pm` subcomponents.

## 2.3 Can OPMN rotate its log files?

Yes. OPMN can rotate its log (`opmn.log`) and debug (`opmn.dbg`) files based on size, time or both. You can enable rotation by configuring rotation-size and rotation-hour attributes of <log> and <debug> tags. When the log file reaches the configured size or at the given hour of the day, the OPMN logging mechanism will close the file, rename it with a time stamp suffix, and then create a new log/debug file.

OPMN console log file (`opmn.out`) is not rotated as this is expected to be a very small file in size. Once OPMN gets past a certain point in initialization, it never writes any output to `opmn.out` anymore. So there is only a small set of messages that could ever appear in this file.

## 2.4 Can OPMN rotate managed process's console log file?

The console log for OPMN managed process (e.g. `$OH/opmn/logs/HTTP_Server~1` file for OHS) is simply the raw stdout and stderr output from the process. OPMN creates a file for each managed process and writes a simple time and event stamp for start, stop and restart before creating the processes and handing the file descriptor to the operating system to use for the managed process's stdout and stderr.

At process startup, before handing off an existing console log file to a managed process, OPMN checks the size against a configured limit (rotation-size attribute of <log> tag), and if the file size exceeds the limit, it will rename the existing file to include a time stamp, and then creates a new file for the managed process. If the rotation-size attribute is not configured, OPMN will not be able to rotate the process's console log file.

OC4J also provides a mechanism to manage its stdout/stderr log files, refer to chapter 4 'OC4J Runtime Configuration' of *Oracle Container for J2EE Configuration and Administration Guide 10g Release 3 (10.1.3.1.0)* for more details.



## 3.0 Topology and DRM

### 3.1 What is dynamic discovery of ONS topology?

In prior releases, each OPMN instance had to be configured with the host and port values of the other ONS servers that it communicated with. This list was maintained in the `ons.conf` file that was maintained by DCM. Whenever this file changed, restarting OPMN was necessary to reflect the change.

OPMN can now optionally discover other ONS servers dynamically. Instead of configuring a list of all other servers to connect to, a discovery mechanism consisting of a multicast address or list of discovery servers is used by OPMN. ONS uses the discovery mechanism to announce new servers and join them into the ONS topology dynamically. This reduces the amount of configuration necessary for each Oracle Application Server instance, eliminates the need to restart OPMN when the topology changes, and removes configuration changes when the topology changes. Explicitly configuration of all nodes is still supported if desired.

### 3.2 In prior releases there used to be a file called `ons.conf` that I don't see anymore.

#### Where is topological information defined?

The information that was stored in `ons.conf` is now configured within the topology section under the `<notification-server>` element in `opmn.xml`. The “nodes” list value is the same as what was specified in `ons.conf`, and there are two elements - the `discover` and `gateway`. Also, OPMN used to extract the instance id, instance name, cluster id and cluster name values from `dcm.conf` (in the `dcm/config` directory), but these values are now specified directly as attributes to the `<process-manager>` element.

### 3.3 What is Dynamic Resource Management (DRM)?

Dynamic Resource Management (DRM) is a new OPMN capability designed to describe a set of desired behaviors and take actions to achieve the desired results. DRM functionality provides a way for you to customize the management of your processes through configuration changes only. The DRM enables you to have process management commands issued based on system conditions according to a set of user-configured directives. Some DRM examples are:

- Start an additional OC4J process every day at 5 p.m. to accommodate peak usage hours.
- Restart an OC4J process whenever it's heap usage grows beyond 500 MBs.
- Spawn an additional OC4J process when average response time exceeds 500 milliseconds as long as there are less than 4 processes running.

DRM functionality is currently available within a single application server instance. Local information is used to make local decisions leading to local actions. The scope of DRM will be expanded to the entire cluster in the next major release of Oracle Application Server.

### 3.4 What is a Resource Management Directive (RMD)?

Resource Management Directives are configured in `opmn.xml`. RMD tells DRM when and what to do. Each RMD has a single conditional followed by a list of actions, which are executed in the order in which they are configured. If any of the actions fail, their execution is stopped, and the exceptions (if any) are executed in the order in which they are configured – if an exception fails then exception processing is stopped.

### 3.5 What kind of conditional can be defined in a RMD?

The conditional that can be used include:

- Keywords to reference any or specific ias-components, process-types, process-sets, processes or applications.

- Any DMS metric maintained by OPMN.
- Time of the day
- Internal events (such as instance joins topology)
- Any logical (AND/OR) combination of the above checks



## 4.0 Configuration

### 4.1 In `opmn.xml`, enclosed within the `notification-server` tags, there is a `port` tag that has `local`, `remote` and `request` attributes. What are these for?

The `local` attribute specifies a local port for notification traffic within the Oracle Application Server Instance. This port can also handle administrative requests like starting and stopping the Oracle Application Server Instance. It cannot handle SSL and is instead restricted to the localhost TCP interface for security reasons. The `remote` port is for inter Oracle Application Server Instance notification traffic. It is also capable of handling administrative requests and supports SSL. The `request` port is to send requests for certain OPMN data dumps. It cannot handle any administrative commands and does not support SSL.

### 4.2 How can I send environment variables to managed processes like OHS and OC4J from OPMN?

Within the OHS and OC4J tags, you need to add directives such as the following:

```
<environment>
  <prop name="PATH" value="/opt/orahome/bin"/>
  <prop name="CLASSPATH" value="/opt/orahome/lib">
</environment>
```

### 4.3 What is routing ID?

The routing ID specifies a routing relationship between OC4Js and OHSs. In other words, an OHS routes to every OC4J that it shares a routing ID with. Every OC4J is assigned a routing ID, similarly each OHS is assigned one or more routing IDs to route to.

OPMN passes the routing ID to OC4J as a system property and to OHS as an environment variable when these are started. OC4J adds this routing ID to the ONS notifications it publishes. OHS listens for notifications from OC4J. When an OHS sees the first notification from an OC4J containing a routing ID on its list, it begins routing to it.

The addition of routing IDs and mount point discovery in Oracle Application Server 10g Release 3 (10.1.3.1.0) version of OHS allows `mod_OC4J` to dynamically discover all aspects of OC4J routing.

### 4.4 Should I configure routing ID for OHS in both `mod_oc4j.conf` and `opmn.xml` files?

No. Out of the box, OHS is configured to pick up its routing ID from `opmn.xml` file. Though it is possible to configure routing IDs for OHS in both `opmn.xml` and directly in `mod_oc4j.conf` file, but if OHS is configured with `routing-id` in both places, it considers it an error and fails to start. So routing IDs for OHS should either be configured in `opmn.xml` (specified as module data under `<ias-instance>` element or under `<ias-component>` element of OHS) or `mod_oc4j.conf` but not at both places.

#### **4.5 I am seeing the following error message in `opmn.log` file. How can I correct this?**

**"oc4j register failed. register\_proc - required port missing for OC4J proc"**

From the error message, it seems that OC4J failed to bind to one of its ports (ajp etc). Please make sure that valid available ports have been specified. Also, refer the troubleshooting chapter in *Oracle Process Manager and Notification Server Administrator's Guide 10g Release 3 (10.1.3.1.0)* for details on how to troubleshoot OPMN related errors.

#### **4.6 Where can I find details of various OPMN commands?**

Refer to chapter 4 'opmnctl Commands' of *Oracle Process Manager and Notification Server Administrator's Guide 10g Release 3 (10.1.3.1.0)* for details of various opmnctl commands.

You can also use 'opmnctl usage' command to get the detail usage message of all commands supported by OPMN.



**ORACLE FUSION MIDDLEWARE**

**Oracle Application Server 10g R3 (10.1.3.1.0):  
Oracle Process Manager and Notification Server FAQ**

October 2006

Author: Shail Goel

Contributing Author: John Lang

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:

+1.650.506.7000

Fax +1.650.506.7200

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

Copyright © Oracle Corporation 2005

All Rights Reserved

This document is provided for informational purposes only, and the information herein is subject to change without notice. Please report any errors herein to Oracle Corporation. Oracle Corporation does not provide any warranties covering and specifically disclaims any liability in connection with this document.

Oracle is a registered trademark of Oracle Corporation.

All other company and product names mentioned are used for identification purposes only and may be trademarks of their respective owners.