Oracle® Fusion Applications

Performance and Tuning Guide

11*g* Release 8 (11.1.8)

E16686-10

June 2014

Documentation for administrators that describes how to monitor and optimize performance for Oracle Fusion Applications.

ORACLE®

Oracle Fusion Applications Performance and Tuning Guide 11*g* Release 8 (11.1.8)

E16686-10

Primary Author: Richard Gugeler

Contributors: Madhubala Ponnekanti, Samuel To, Dillip Praharaj, Avanish Srivatsav, Saul Reyes, Gerard Antony, Mrithiumjay Srinivas

Contents

3 Tuning the Database

4 Diagnosing the Oracle Fusion Applications Middle Tier Performance

Preface

Welcome to the Oracle Fusion Applications *Performance and Tuning Guide*! This guide describes performance and tuning checks and tweaks that are specific to Oracle Fusion Applications.

Audience

This document is intended for Oracle Fusion Applications administrators and developers, and operators working in a runtime production environment, and assumes familiarity with Java and SQL.

Documentation Accessibility

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

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle 11g Fusion Middleware documentation set:

- Oracle Fusion Applications Administrator's Guide
- Oracle Fusion Middleware Administrator's Guide
- Oracle JRockit JDK Tools Guide
- Oracle JRockit Flight Recorder Run Time Guide
- Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server
- Oracle Fusion Middleware Configuring and Using the Diagnostics Framework for Oracle WebLogic Server

- Oracle Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework

Conventions

The following text conventions are used in this document:

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

What's New in This Guide

The following topics introduce the new and changed features of the Oracle Fusion Applications Performance and Tuning Guide 11*g* Release 8 (11.1.8) and other significant changes that are described in this guide, and provides pointers to additional information.

New and Changed Features for 11*g* Release 8 (11.1.8)

The Oracle Fusion Applications Performance and Tuning Guide 11*g* Release 8 (11.1.8) includes the following new and changed features for this document.

- A new chapter, Chapter 4, "Diagnosing the Oracle Fusion Applications Middle Tier Performance,", describes how to diagnose Java applications in the middle tier using Oracle Enterprise Manager Cloud Control (Cloud Control).

New and Changed Features for 11*g* Release 7 (11.1.7)

There are no new features for the Oracle Fusion Applications Performance and Tuning Guide 11*g* Release 7 (11.1.7).

Recommended database settings have been updated in Table 3–2.

1

Monitoring and Tuning Oracle Fusion Applications

This chapter discusses how to find the information you need to examine so you can tune your system. It includes how to monitor and tune the database and Oracle Fusion Applications, and troubleshooting.

This chapter includes these sections:

1.1 Introduction

Every system of hardware and installed applications is different. Even though Oracle Fusion Applications are written and installed using industry-standard best practices, you can custom tailor your system to improve how it supports your environment.

But to tune your system, you need to locate and examine data. This chapter will explain what data you need to examine, and what tools you will use to gather the data.

1.2 Monitoring and Tuning Oracle Fusion Applications

In general, most of the settings that come default in Oracle Fusion Applications are already tuned.

These guidelines are provided to help ensure your Oracle Fusion Applications instance runs optimally. Note that all metrics listed are from Oracle Enterprise Manager Cloud Control.

- Monitor the key host metrics, shown in Table 1–1, to ensure the underlying server hosts are healthy. Rather than constantly checking the metric values, you can set up alert thresholds in Cloud Control and receive notification when thresholds are exceeded. For more information, see "Creating Monitoring Templates" in the *Oracle Fusion Applications Administrator's Guide*.
- Monitor the key component metrics, such as WebLogic server metrics, to ensure each component is healthy.
- Monitor the number of incidents and logs to ensure the application is configured properly and not constantly wasting resources generating error messages. Review log levels to ensure they are not set too low. See "Troubleshooting Oracle Fusion

Applications Using Incidents, Logs, QuickTrace, and Diagnostic Tests" in the *Oracle Fusion Applications Administrator's Guide* for more information

- Monitor the database to ensure it is operating optimally. Follow the guidelines in Chapter 3, "Tuning the Database," to make sure that statistics are being collected.

Table 1–1 Key Host Metrics

Metric Category	Metric Name	Warning Threshold	Critical Threshold	Comments
Disk Activity	Disk Device Busy	>80%	>95%	
Filesystems	Filesystem Space Available	<20%	<5%	
Load	CPU in I/O wait	>60%	>80%	
	CPU Utilization	>80%	>95%	
	Run Queue (5 min average)	>2	>4	The run queue is normalized by the number of CPU cores.
	Swap Utilization	>75%	>90%	
	Total Processes	>15000	>25000	
	Logical Free Memory %	<20	<10	
	CPU in System Mode	>20%	>40%	
Network Interfaces Summary	All Network Interfaces Combined Utilization	>80%	>95%	
Switch/Swap Activity	Total System Swaps	>3	>5	Value is per second.
Paging Activity	Pages Paged-in (per second)			
	Pages Paged-out (per second)			The combined value of Pages Paged-in and Pages Paged-out should be <=1000

1.2.1 How to Analyze Host Metrics

Administrators will find it useful to study these suggestions on further analysis to undertake when a metric value exceeds threshold. The commands provided are for the Linux operating system.

When logical free memory/swap activity or paging activity is beyond threshold

This usually happens when memory is not sufficient to handle demands from all the running processes.

- Linux: Check `cat/proc/meminfo` and confirm total RAM is expected.

  Windows: Open the Task Manager, click the Performance tab and check the Physical Memory section.

- Check if there are unallocated huge pages. If there are and the WebLogic Server/Oracle instances are not expected to use them, reduce the huge page pool size.

- Linux: Run `top` and sort by resident memory (type **OQ**). Look for processes using the most resident memory and investigate those processes.

  Windows: Open the Task Manager, click the Processes tab and click the Mem Usage column to sort the processes by memory usage.

When page activity is beyond threshold

Follow the steps in "When logical free memory/swap activity or paging activity is beyond threshold" to view and analyze memory usage.

When Network Interface Error Rates Is Beyond Threshold

The normal cause is misconfiguration between the host and the network switch. A bad network card or cabling also can cause this error. You can run `/sbin/ifconfig` to identify which interface is having packet errors. Contact network administrator to ensure the host and the switch are using same data rate and duplex mode.

Otherwise, check if cabling or the network card is faulty and replace as appropriate.

When Packet Loss Rate Is Beyond Threshold

The normal cause of this error is network saturation of bad network hardware.

- Run `lsof -Pni | grep ESTAM` to determine which network paths are generating the problem.
- Then run `mtr <target host>` or `ping <target host>` and look for packet lost on that segment.

  ```
  20 packets transmitted, 20 received, 0% packet loss, time 18997ms
  rtt min/avg/max/mdev = 0.168/0.177/0.200/0.010 ms
  ```

  The packet loss should be 0% and rtt should be less than .5 ms.
- Ask the network monitoring staff to look for saturation or network packet loss from their side.

When Network Utilization Is Beyond Threshold

The normal cause is very heavy application load.

- Run `top` or `lsof` to determine which processes are moving a lot of data.
- Use `tcpdump` to sample the network for usage patterns.
- Use `atop`, `iftop`, `ntop` or `pkstat` to see which processes are moving data.

When CPU Usage or Run Queue Length Is Beyond Threshold

The normal cause is runaway demand, a poorly performing application, or poor capacity planning.

- Linux: Run `top` to identify which application/process is using time.

  Windows: Open the Task Manager, click the Processes tab and click the CPU column to sort the processes based on CPU usage.
- If top processes are WebLogic Server JVM processes, conduct a basic WebLogic Server health check. That is, review logs to see if there are configuration errors causing excessive exceptions, and review metrics to see if the load has increased. Use JVMD for a more detailed analysis.
- If top processes are Oracle processes, use Enterprise Manager to look for high load SQL.

When System CPU Usage Is Beyond Threshold

- High system CPU use could be due to kernel processes looking for pages to swap out during a memory shortage. Follow the steps listed in the "When logical free memory/swap activity or paging activity is beyond threshold" section to further diagnose the problem.

- High system CPU use is also frequently related to various device failures. Run `{{dmesg | less}}` and look for repeated messages about errors on some particular device, and also have hardware support personnel check the hardware console to see if there are any errors reported.

When Filesystem Usage Is Beyond Threshold

The normal cause is an application that is logging excessively or leaving behind temporary files.

- Run `lsof -d 1-99999 | grep REG | sort -nrk 7 | less` to see currently open files sorted by size from largest to smallest. Investigate the large files.
- Run `du -k /mount_point_running_out_of_space > /tmp/sizes` to get space used for directories under the mount point. This may take a long time. While it is running, run `sort -nr /tmp/sizes` and find the directories using most space and investigate those first.

When Total Processes Is Beyond Threshold

The normal cause is runaway code or a stuck NFS filesystem.

- Linux: Run `ps aux`. If many processes are in status D, run `df` to check for stuck mounts.

  Windows: Run Task Manager, click the Processes tab, and check the list of running processes.

  If there are hundreds or thousands of processes of a particular program, determine why.
- Run `ps o pid,nlwp,cmd | sort -nrk 2 | head` to look for processes with many threads.

When Disk Device Busy Is Beyond Threshold

- Check for disk drive failure.

  Linux: As root, check `/var/log/messages*` and `/var/log/mcelog` to see if there are any error messages indicating disk failure. For a RAID array, the disk controller needs to be checked. The commands will be specific to the controller manufacturer.

  Windows: Run `perfmon` and look at the Alert logs. Run `chkdsk` to check for disk failure.
- Look for processes that are using the disk. From a shell window, execute `ps aux | grep ' D. '` several consecutive times to look for processes with "stat" D.

1.2.2 How to Check for Network Connectivity Issues

Poor performance is a major indicator of network connectivity problems.

- Check for cumulative dropped packets drops for each host.

  Linux

```
netstat -s | grep 'TCP data loss'
  4007 segments retransmited
  3302 TCP data loss events
```

  Windows

```
netstat -s
```

Look for "TCP Statistics for IPv4" and for "Segments Retransmitted."

The counts for Linux and Windows should be 0 or growing very slowly over time.

- Check for realtime dropped packets on specific network paths. The `ping` command is the same for Windows and Linux systems.

```
ping -c 20 other_host
  20 packets transmitted, 20 received, 0% packet loss, time 18997ms
  rtt min/avg/max/mdev = 0.168/0.177/0.200/0.010 ms
```

Packet loss should be 0%.

rtt should be less than .5 ms, except that it can be higher between the browser and load balancer.

In Windows, the command

```
ping -n 20 other_host
```

will show similar output, only with more information:

```
Ping statistics for 123.45.67.89:
Packets: Sent = 20, Received = 20, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 96ms, Maximum = 121ms, Average = 104ms
```

- Check for network interface errors.

Linux

```
/sbin/ifconfig eth0 | grep errors
   RX packets:842803463 errors:0 dropped:0 overruns:0 frame:0
   TX packets:667946307 errors:0 dropped:0 overruns:0 carrier:0
```

Windows

```
ipconfig /all
```

will not show TX or RX packets information. However, Microsoft Network Monitor (free from the Microsoft Download site) will, when it is installed and configured properly.

1.2.3 How to Analyze WebLogic Server Metrics

These metrics provide an indication of whether the WebLogic Server is in a healthy state. Performance may degrade if any of the metrics is exceeding its threshold.

Table 1–2 describes the WebLogic Server metrics you should monitor in Cloud Control. See the "Creating Monitoring Templates" section in the *Oracle Fusion Applications Administrator's Guide* to create a monitoring template.

Table 1–2 WebLogic Server Metrics

Metric Category	Metric Name	Warning Threshold	Critical Threshold	Comments
Datasource Metrics	Connections in Use	>250	>400	
	Connection Requests that Waited (%)	>10%	>20%	
	Connection Creation Time (ms)			
JVM Garbage Collectors	Garbage Collector - Percent Time spent (elapsed)	>10%	>20%	
JVM Metrics	Heap Usage	>90%	>98%	
Response	Status		=Down	This provides instance availability.
Server Servlet/JSP Metrics	Request Processing Time (ms)	>10s	>15s	
Server Work Manager Metrics	Work Manager Stuck Threads	>5	>10	
JVM Threads	Deadlocked Threads	>2	>5	
Module Metrics By Server	Active Sessions			

When CPU Usage On Host Is Beyond Threshold and WebLogic Server Process Is Identified as Top CPU Consumer

- Examine the % Time spent in the GC metric to see if JVM is doing excessive GC (>60 percent). If so, follow the process for diagnosing WebLogic Server heap pressure.
- Look for incident creation rate and error logs and see if something is triggering a massive amount of logging/errors.
- In JVMD, select the CPU state filter and look at top methods. Look for threads that are consistently in a CPU state.

When There Is a Spike in Active Web Sessions

- Check access logs to see if there is a spike in the number of users.
- Check if there are stuck threads, which could cause users to log in again.
- Check session distribution across WebLogic Server managed servers and see if there is a problem with the load balancer.
- Check session timeout in `web.xml`, and see if it is too high or too low.

When There Are Stuck Threads On the System

- Get the ECID from the stuck thread error in the WebLogic Server log.
- From the Request Monitor, search for the ECID and get details from JVMD.
- Alternatively, use JVMD to search for stuck threads and see the timing breakdown.
- A stuck thread will also result in an incident with a JFR recording. Use Oracle JRockit Mission Control (JRMC) to analyze the recording. JRMC is a set of tools that runs on the Oracle JRockit JVM and deliver advanced, unobtrusive JVM

monitoring and management, suitable for use in development and production environments.

When There Are Deadlocks Detected On the System

- In JVMD, inspect the threads that are in a blocked state.
- Deadlock threads normally also will be reported as a stuck thread in the WebLogic Server log. Use the Request Monitor to search for the ECID and expand down into JVMD to show the blocking thread.

When Request Processing Time Is Beyond Threshold

- Examine the % Time spent in GC metric to see if JVM is doing excessive garbage collection
- Look for incident create rate and error logs and see if something is triggering a massive amount of logging/errors.
- In JVMD, look at the thread states and see where most processing time is going.
- Check the metric **Garbage Collection - Invocation Time (ms)** under the **JVM Garbage Collectors** metric category. Sometimes if you run many managed server instances on the same host, you may be able to reduce time spent in garbage collection by reducing the number of garbage collector threads in each JVM. The default is based on the number of CPUs and could be too high if there are multiple active JVMs running on the same machine. In those cases, if you are using JRockit, add the `-XXgcThreads=4` option when starting the JVM. To add the option, edit the *`DOMAIN_HOME`*`/bin/fusionapps_start_params.properties` file, look for `-Xgc:genpar` and add the `-XXgcThreads=4` option after it (for example `-Xgc:genpar -XXgcThreads=4`). The value 4 directs the JVM to use four threads to perform garbage collection. You can try different values from 4 to the number of CPU cores and observe if the % Time spent in GC metric improves. For other platforms, see Section 1.3, "Tuning Platforms for Oracle Fusion Applications."

When Percent Time Spent in GC Is Beyond Threshold

- Check the session count. If there is a sudden surge of sessions due to user load, the JVM could be short on heap. Increase heap if possible, or add additional managed server instances.
- Look at the stuck threads count. Stuck threads could increase the number of active session, as users could be launching new sessions hoping for a faster response.
- Look at the incident creation rate and error logs and see if something is triggering a massive amount of logging/errors. The incident creation/logging operations could be causing a high amount of object creation and garbage collection stress.
- Generate a heap dump using JVMD and analyze the top retainer of memory.
- Use JRMC to connect and extract a JFR recording. Examine the Memory panel and allocation details to see what is doing a lot of allocations.

When Percent Connection Requests Waiting Is Beyond Threshold

- Examine the number of sessions and request rate, and see if there is a spike in the load that would account for an increased demand for connections.
- In JVMD, see where time is spent. For example, requests could be running longer due to slow SQLs (and retain the connection longer). In that case, identify and tune slow SQLs.

- Consider increasing the initial capacity setting of the corresponding data source.

1.2.4 How to Analyze Oracle HTTP Server Metrics

These metrics provide an indication of whether the Oracle HTTP Server is in a healthy state. Performance may degrade if any of the metrics is exceeding its threshold.

Table 1–3 describes the Oracle HTTP Server metrics you should monitor in Cloud Control. See the "Creating Monitoring Templates" section in the *Oracle Fusion Applications Administrator's Guide* to create a monitoring template. Also see Section 1.4, "Tuning Oracle HTTP Server," and "Monitoring the Oracle Fusion Applications Middle Tier" in the *Oracle Fusion Applications Administrator's Guide*.

Table 1–3 Oracle HTTP Server Metrics

Metric Category	Metric Name	Warning Threshold	Critical Threshold
OHS Server Metrics	Busy Threads (%)	>85%	>95%
	Request Throughput (requests per second)	TBD	Yes
OHS Response Code Metrics	HTTP 4xx errors		
	HTTP 5xx errors		
OHS Virtual Host Metrics	Request Processing Time for a Virtual Host	>10s	>15s

When Busy Threads % Is Beyond Threshold

- Check request throughput to see if load has increased. If the increased load is expected and CPU and memory resources on the OHS host has not exceeded threshold, consider increasing ServerLimit/MaxClients and ThreadsPerChild in `httpd.conf`.
- Check request process time on both OHS and underlying WebLogic Server to see if requests are taking longer. If WebLogic Server response time is increasing, check the key metrics for the WebLogic Server.
- If possible, ensure the client browser cache is enabled to reduce number of requests submitted.
- Check OHS Response Code Metrics. If there is a sudden increase of HTTP 4xx errors or HTTP 5xx errors, check the health of the underlying WebLogic Servers.
- Check and increase the minimum and maximum spare threads for Oracle HTTP Server.

  In the `httpd.conf` file located in `instance_home/config/ohs/<ohs_name>/httpd.conf`:

  – Increase MaxSpareThreads to 800.

  – Increase MinSpareThreads to 200.

When Request Processing Time for a Virtual Host Exceeds Threshold

- Check the key host metrics to ensure the OHS host is healthy.
- For each URL requested, OHS will first check DocumentRoot before passing the request to WebLogic Server. Check the utilization and health of the disk to which the DocumentRoot is pointing. If it is a NFS mount, check the health of the NFS mount point.

- Check the key metrics for the underlying WebLogic Server(s) and see if they are healthy.
- OHS accesses /tmp for each POST request, so check the performance of the /tmp filesystem.

1.2.5 How to Analyze Oracle Business Intelligence Server Metrics

These metrics provide an indication of whether the Oracle Business Intelligence Server is in a healthy state.

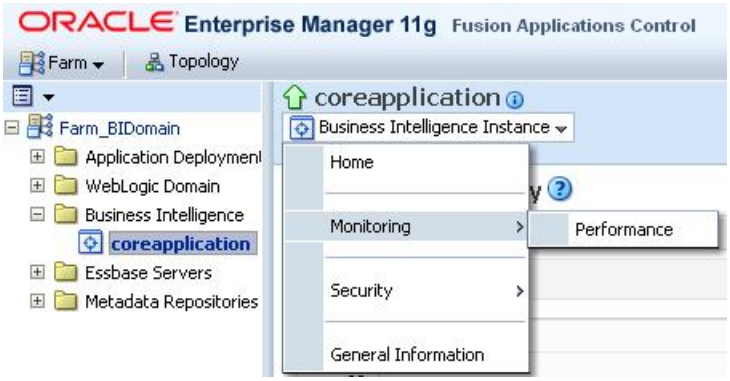
To start monitoring:

1. Log in to Oracle Enterprise Manager Fusion Applications Control. See the "Starting Fusion Applications Control" section in the *Oracle Fusion Applications Administrator's Guide*.

   Note: Creating a monitoring template for Oracle Business Intelligence Server metric targets is not available in Cloud Control.

2. From the navigation pane, expand the farm, and then Business Intelligence.
3. Click **coreapplication**.
4. From the Business Intelligence Instance menu, choose **Monitoring > Performance**, as shown in Figure 1–1.

Figure 1–1 Checking Oracle Business Intelligence Performance Metrics



5. Use Fusion Applications Control to configure parameters related to Oracle Business Intelligence Suite Enterprise Edition.

Fusion Applications Control can monitor various BI components, including:

- Weblogic Analytics Application
- Oracle BI Presentation Services
- Oracle BI Server
- Oracle Weblogic Server (administration and managed servers)

1.2.6 How to Gather Key Identity Management Server Metrics

Oracle Access Manager and Oracle Identity Manager are both WebLogic Server instances. See Section 1.2.3, "How to Analyze WebLogic Server Metrics" to monitor their health. See also Section 1.2.9, "How to Tune and Troubleshoot Oracle Identity Management."

Use Cloud Control to monitor the Oracle Internet Directory and Oracle Identity Manager databases. For information on creating monitoring templates in Cloud Control to obtain metrics, see the "Creating Monitoring Templates" section in the *Oracle Fusion Applications Administrator's Guide*.

See Table 1–4 for Oracle Identity Manager metrics.

Table 1–4 Oracle Identity Manager Metrics

Metric Category	Metric Name	Warning Threshold	Critical Threshold
Oracle Identity Manager Cluster			
Provisioning Requests	Completed Provisioning Requests		
	Failed Provisioning Requests		
Reconciliations (Last 24 Hours)	Jobs Completed		
Response	Status		
Role Grant Requests	Completed Role Grant Requests		
	Completed Role Grant Requests Processing Time (per sec)		
	Failed Role Grant Requests		
	Pending Role Grant Requests		
Self Service Requests	Completed Self Service Requests		
	Completed Self Service Requests Processing Time (per sec)		
	Failed Self Service Requests		
	Pending Self Service Requests		
Oracle Identity Manager Server			
Response	Status		
Resource Utilization	CPU Utilization (%)	80	90
	Memory Utilization (%)	80	90
Adapters	Average Adapter Execution Time (ms)		
	Completed Adapter Executions		
	Maximum Adapter Execution Time (ms)		
	Minimum Adapter Execution Time (ms)		

See Table 1–5 for Oracle Internet Directory metrics.

Table 1–5 Oracle Internet Directory Metrics

Metric Category	Metric Name	Warning Threshold	Critical Threshold
LDAP Operation Response Time	Bind Operation Response Time		
	Request Processing Time (ms)		
LDAP Server Resource Usage	Total CPU Usage (%)	80	90

Table 1–5 (Cont.) Oracle Internet Directory Metrics

Metric Category	Metric Name	Warning Threshold	Critical Threshold
	Total Memory Usage (%)	80	90
Response	Status		
(Critical Events) System Resource Events (3113 Errors)	Number of 3113 Error Occurrences		
(Critical Events) System Resource Events (3114 Errors)	Number of 3114 Error Occurrences		
LDAP Failed Bind Operations Profile	Failed Bind Operations		
LDAP Server Resource Usage	Completed Bind Operations		
	Completed Compare Operations		
	Completed Modify Operations		
	Completed Search Operations		
	Total Operations		
User LDAP Operations Statistics	Failed Base Search Operations		
	Failed Bind Operations		
	Failed Compare Operations		
	Failed Delete Operations		
	Successful Base Search Operations		

To enable the collection of user LDAP Operation Statistics, edit the configuration using Fusion Applications Control:

1. From the navigation pane, expand the farm and then the Oracle Internet Directory target.
2. From the **Administration** menu, choose **Server Properties**.
3. Click **User Statistics Collection** to enable this feature.
4. Click **Apply** to save your changes.
5. From the **Administration** menu, choose **Shared Properties**.
6. Enter a valid DN (for example, `cn=orcladmin`) to enable user statistics collection for that user.

1.2.7 How to Analyze Key Enterprise Scheduler Metrics

The metrics shown in Table 1–6 provide an indication of whether the Enterprise Scheduler instance is performing well. See the "Creating Monitoring Templates" section in the *Oracle Fusion Applications Administrator's Guide*.

Table 1–6 Key Enterprise Scheduler Metrics

Metric Category	Metric Name	Warning Threshold	Critical Threshold	Comments
Completed Job Summary	Average Elapsed Time (ms)			You can define different thresholds for different job names.
Long Running Job	Elapsed Time (ms)			
WorkAssignment Metrics aggregated across Group Members	Average Wait Time for Requests in Ready State (seconds)			

When the Value of Average Elapsed Time for the Completed Jobs Is Higher Than Expected

- Check the key host and WebLogic Server metrics and see if any component that could be involved in process batch jobs is in an unhealthy state.
- Follow the steps listed in Section 2.1.4, "How to Troubleshoot Slow Batch Jobs" and analyze several jobs to see if there are any common causes.

When the Value of Elapsed Time Under the Long Running Job Category Is Higher Than Expected

- Open the Enterprise Scheduler home page in Oracle Enterprise Manager Fusion Applications Control and examine the Top 10 Long Running Jobs.
- Identify the job of interest, and follow the steps in Section 2.1.4, "How to Troubleshoot Slow Batch Jobs."

When Average Wait Time For Requests in Ready State (seconds) Is Higher Than Expected

Follow the steps in Section 2.1.4.1, "Troubleshooting Jobs That Are in Wait/Ready/Blocked State for a Long Time."

1.2.8 How to Monitor Key SOA Metrics

Monitoring SOA involves monitoring SOA Infrastructure, SOA Composite and SOA servers. See the "Creating Monitoring Templates" section in the *Oracle Fusion Applications Administrator's Guide*.

Use Table 1–7 to locate the key performance metrics for SOA Composite.

Table 1–7 SOA Composite Metrics

Metric Category	Metric Name
Mediator Case	Invocation count throughput in last 5 minutes
SOA Composite - Response Metrics	Composite Status
SOA Composite - Component Detail Metrics	Component: Business Faults
	Component: Error Rate (%)
SOA Composite - Services/References Detail Metrics	Service/Reference: Average Incoming Messages Processing Time (ms)
	Service/Reference: Average Outbound Messages Processing Time (ms)

Table 1–7 (Cont.) SOA Composite Metrics

Metric Category	Metric Name
	SOA Composite: Error Rate (%)
	SOA Composite: Synchronous Response Time (ms)
	SOA Composite: Total Business

Use Table 1–8 to locate the key performance metrics for SOA Infrastructure.

Table 1–8 SOA Infrastructure Metrics

Metric Category	Metric Name
SOA Infra Response	Up Down Status
SOA Infrastructure - Message Detail Metrics	Errors (minute)
SOA Infrastructure - Service Engine Detail Metrics	Service Engine: Error Rate (%)

1.2.9 How to Tune and Troubleshoot Oracle Identity Management

Follow the steps in this section to tune Oracle Identity Management specifically for Oracle Fusion Applications.

Most of these settings should be set by default if your environment is newly provisioned. If your environment is upgraded from a previous release, you will need to manually check and adjust the settings.

Optimize LDAP Search

Description: Optimize LDAP search by enabling search filters.

Solution:

- Create an ldiff file named `searchfilter_oid_tuning.ldif` with this content:

```
dn: cn=dsaconfig, cn=configsets, cn=oracle internet directory
changetype: modify
add: orclinmemfiltprocess;dn
orclinmemfiltprocess;dn:
cn=Roles,cn=fscm,cn=FusionDomain,cn=JPSContext,cn=FusionAppsPolicies
orclinmemfiltprocess;dn:
cn=Roles,cn=crm,cn=FusionDomain,cn=JPSContext,cn=FusionAppsPolicies
orclinmemfiltprocess;dn:
cn=Roles,cn=hcm,cn=FusionDomain,cn=JPSContext,cn=FusionAppsPolicies
orclinmemfiltprocess;dn: cn=Permission
Sets,cn=fscm,cn=FusionDomain,cn=JPSContext,cn=FusionAppsPolicies
orclinmemfiltprocess;dn: cn=Permission
Sets,cn=hcm,cn=FusionDomain,cn=JPSContext,cn=FusionAppsPolicies
orclinmemfiltprocess;dn: cn=Permission
Sets,cn=crm,cn=FusionDomain,cn=JPSContext,cn=FusionAppsPolicies
orclinmemfiltprocess:dn; cn=Permissions,cn=JAAS
Policy,cn=fscm,cn=FusionDomain,cn=JPSContext,cn=FusionAppsPolicies
orclinmemfiltprocess;dn:
cn=hcm,cn=FusionDomain,cn=JPSContext,cn=FusionAppsPolicies
orclinmemfiltprocess;dn:
cn=crm,cn=FusionDomain,cn=JPSContext,cn=FusionAppsPolicies
orclinmemfiltprocess;dn:
cn=fscm,cn=FusionDomain,cn=JPSContext,cn=FusionAppsPolicies
```

- At the command prompt, run this command:

```
ldapmodify -p portNum -h hostname -D cn=orcladmin -f searchfilter_oid_
tuning.ldif
```

Log Levels

Description: Oracle Identity Management stack WebLogic Server log levels are too fine-grained and need to be set to Severe.

Solution: In all WebLogic Servers in the Oracle Identity Management domain, change log levels to SEVERE. This is a two-part process.

- **Part 1:** Manually edit the `logging.xml` file, or by using the Oracle WebLogic Server Administration Console.

  Edit the `logging.xml` file that is in each server directory of the Oracle Identity Management Domain domain, such as OAM_Server1, OIM_Server1, and SOA, and set `level='SEVERE'` for all log_handlers and loggers. The path to each `logging.xml` file will resemble:

  DOMAIN_HOME`/config/fmwconfig/<servername>`

- **Part 2:** Edit the log levels in the Oracle WebLogic Server Administration Console:
  - Log in to the console (`http://hostname:port/console`).
  - Click the Servers link.
  - Click the desired server.
  - Click the **Logging** tab.
  - Scroll down and click the **Advanced** link.
  - In the Message destination(s) section, change the log levels as shown here:

    ```
    Log file : Severity level: warning
    Standard out :Severity level: error
    Domain log broadcaster :Severity level: error
    Memory buffer: Memory Buffer Severity level: error
    ```

  - Save the changes.
  - Repeat this for all WebLogic Servers in the Oracle Identity Management stack, such as OAM_Server1, OIM_Server1, and SOA.
  - Click **Activate Changes**.
  - Restart the server.

Tune Two OID Configuration Parameters

Description: Two Oracle Internet Directory configuration parameters, `orclmaxcc` and `orclserverprocs`, need to be appropriately tuned.

Solution: Change `orclmaxcc` to 10 and tune the number of OID processes:

- Name the sample script `config_oid_tuning.ldif`. You will need to set `cn=oid1` to your component name. In a multi-component environment, this needs to be changed accordingly. You will need to set orclserverprocs to the number of cores in the OID server that is used.

  ```
  dn: cn=oid1,cn=osdldapd,cn=subconfigsubentry
  changetype: modify
  replace: orclmaxcc
  ```

```
orclmaxcc: 10
orclserverprocs: <number of cores>
```

- Apply the script by running this command at the command prompt:

```
ldapmodify -p portNum -h hostname -D cn=orcladmin -f config_oid_tuning.ldif
```

Enable Timing Logging

Description: Add parameters to enable timing logging for OID.

Solution:

- Add this entry to the config.xml file in `./oid/user_projects/domains/oid_domain/config/` and the `./oim/user_projects/domains/oim_domain/config/` directories for each WebLogic Server in the Oracle Identity Management domain:

```
    <web-server>
      <web-server-log>
        <file-name>logs/access.log.%yyyyMMdd%</file-name>
        <rotation-type>byTime</rotation-type>
        <number-of-files-limited>true</number-of-files-limited>
        <rotate-log-on-startup>true</rotate-log-on-startup>
        <buffer-size-kb>0</buffer-size-kb>
        <logging-enabled>true</logging-enabled>
        <elf-fields>date time time-taken bytes c-ip s-ip sc-status
sc(X-ORACLE-DMS-ECID) cs-method cs-uri
cs(User-Agent) cs(ECID-Context) cs(Proxy-Remote-User)
cs(Proxy-Client-IP)</elf-fields>
        <log-file-format>extended</log-file-format>
        <log-time-in-gmt>false</log-time-in-gmt>
        <log-milli-seconds>true</log-milli-seconds>
      </web-server-log>
    </web-server>
```

- To set the access log format, add this string to the `httpd.conf` file in the `/u01/ohsauth/ohsauth_inst/config/OHS/ohs1` path.

```
LogFormat "%h %l %u %t \"%r\" %>s %b %D %{X-ORACLE-DMS-ECID}o" common
```

Increase Policy Cache Timeout

Description: By default, entries in the security policy cache time out every 12 hours. When these entries time out, sporadic slowness may be experienced because they need to be repopulated. To avoid this, increase the timeout value.

Solution:

- Open the *DOMAIN_HOME*`/config/fmwconfig/jps-config.xml` file.
- Add the following entry to the <serviceInstance name="policystore.ldap" provider="policystore.provider"> section and select a timeout value (in milliseconds):

```
<property name="oracle.security.jps.policystore.refresh.purge.timeout"
value="1296000000"/>
```

  This example will set the timeout to 15 days. Adjust the property value as needed.

```
<serviceInstance name="pdp.service" provider="pdp.service.provider">
...
<property name="oracle.security.jps.policystore.refresh.purge.timeout"
value="1296000000"/>
```

Adjust Connection Settings Between Managed Servers and LDAP Server

Description: Adjust the following LDAP settings to improve system stability and performance.

Solution:

- From each Fusion Applications domain, login to the instance of WebLogic Server called the Administration Server.
- Select Security realms and click **myrealm**.
- Click **Providers**.
- Click **OIDAuthenticator**.
- Select the **Provider Specific** tab.
- Check the **Keep Alive Enabled** option.
- Set **Connection Timeout** to 60.
- Set **Parallel Connect Delay** to 1.
- Set **Results Time Limit** to 300000.
- Select the **Ignore Duplicate Membership** option.

Adjust Connection Pool Sizes to Oracle Access Manager

Description: Increase the connection pool size to Oracle Access Manager to avoid contention when load is high.

Solution:

- From each Fusion Applications domain, login to the Administration Server instance of WebLogic Server.
- Select **Security Realms** and click **myrealm**.
- Click **Providers**.
- Click **OAMIdentityAsserter**.
- Select the **Provider Specific** tab.
- Set **Minimum Access Server Connections In Pool** to 10.
- Set **Maximum Access Server Connections In Pool** to 100.
- Set **Connection Timeout** to 60.
- Set **Parallel Connect Delay** to 1.
- Set **Results Time Limit** to 300000.
- Select the **Ignore Duplicate Membership** option.

Adjust Configuration Settings for Oracle Access Manager Domain

Description: The following setting changes will ensure optimal performance for Oracle Access Manager under typical Oracle Fusion Applications workload.

Solution: Edit the settings shown in Table 1–9 in the *OAM Domain Home*`/config/fmwconfig/oam-config.xml` file and set the recommended values:

Table 1–9 Recommended Settings for Best Oracle Access Manager Performance

Location	Parameter	Value
DOMAIN_HOME/config/fmwconfig/oam-config.xml - <Setting Name="UserIdentityStore" Type="htf:map">	SearchTimeLimit	<Setting Name="SearchTimeLimit" Type="xsd:integer">120</Setting>
DOMAIN_HOME/config/fmwconfig/oam-config.xml - <Setting Name="UserIdentityStore" Type="htf:map">	MIN_CONNECTIONS	<Setting Name="MIN_CONNECTIONS" Type="xsd:integer">10</Setting>
DOMAIN_HOME/config/fmwconfig/oam-config.xml - <Setting Name="UserIdentityStore" Type="htf:map">	MAX_CONNECTIONS	<Setting Name="MAX_CONNECTIONS" Type="xsd:integer">100</Setting>
DOMAIN_HOME/config/fmwconfig/oam-config.xml - <Setting Name="Webgate_IDM" Type="htf:map">	cachePragmaHeader	Delete value
DOMAIN_HOME/config/fmwconfig/oam-config.xml - <Setting Name="Webgate_IDM" Type="htf:map">	cacheControlHeader	Delete value
DOMAIN_HOME/config/fmwconfig/oam-config.xml - <Setting Name="Webgate_IDM" Type="htf:map">	numOfConnections	<Setting Name="numOfConnections" Type="xsd:string">10</Setting>

1.3 Tuning Platforms for Oracle Fusion Applications

If you are using the IBM AIX, Solaris SPARC, or Solaris x64 operating system, Oracle recommends that you incorporate these settings.

IBM AIX

Incorporate these settings for best performance when using IBM JVM 9:

```
-Xgcpolicy:gencon
-Xcompressedrefs
-XtlhPrefetch
```

Solaris SPARC

- In *all* the domains, make these changes to the *DOMAIN_HOME*/bin/fusionapps_start_params.properties file:
  - Edit fusion.default.SunOS-sparc.memoryargs and add these arguments:

    ```
    -XX:ReservedCodeCacheSize=128m -XX:+UseParallelOldGC
    -XX:ParallelGCThreads=4
    ```

  - Restart all the servers of all domains.
- In *all* the domains, make these changes to the *DOMAIN_HOME*/bin/fusionapps_start_params.properties file:
  - Add the fusion.AdminServer.SunOS-sparc.memoryargs entry with these arguments:

    ```
    -XX:PermSize=256m -XX:MaxPermSize=512m -XX:+UseParallelGC
    ```

```
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=@HEAP_DUMP_PATH@
-XX:+ParallelGCVerbose -XX:ReservedCodeCacheSize=128m
-XX:+UseParallelOldGC -XX:ParallelGCThreads=2
```

  - Change `@HEAP_DUMP_PATH@` to a valid path for the Heap Dump files to be written.
  - Restart the Admin servers of all the domains.

- For the BIDomain, make these changes to the `$BIDomain/bin/fusionapps_start_params_bi.properties` file:
  - Edit `fusion.default.SunOS-sparc.memoryargs` and append these arguments:

```
-XX:ReservedCodeCacheSize=128m -XX:+UseParallelOldGC
-XX:ParallelGCThreads=4
```

  - Restart all the servers of the BIDomain.
- For the BIDomain, make these changes to the `$BIDomain/bin/fusionapps_start_params_bi.properties` file:
  - Add the `fusion.AdminServer.SunOS-sparc.memoryargs` entry with these arguments:

```
-XX:PermSize=256m
-XX:MaxPermSize=512m -XX:+UseParallelGC -XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=@HEAP_DUMP_PATH@ -XX:+ParallelGCVerbose
-XX:ReservedCodeCacheSize=128m -XX:+UseParallelOldGC
-XX:ParallelGCThreads=2
```

  - Change `@HEAP_DUMP_PATH@` to a valid path for the Heap Dump files to be written.
  - Restart the Admin server of the BIDomain.
- For the CommonDomain, make these changes to the `$CommonDomain/bin/fusionapps_start_params.properties` file:
  - Edit `fusion.HelpPortalCluster.SunOS-sparc.memoryargs` and add these arguments:

```
-XX:ReservedCodeCacheSize=128m -XX:+UseParallelOldGC
-XX:ParallelGCThreads=4
```

  - Restart the HelpPortalCluster servers of the CommonDomain.
- For the ProcurementDomain, make these changes to the `$ProcurementDomain/bin/fusionapps_start_params.properties` file:
  - Edit `fusion.ProcurementCluster.SunOS-sparc.memoryargs` to add the arguments:

```
-XX:ReservedCodeCacheSize=128m -XX:+UseParallelOldGC
-XX:ParallelGCThreads=4
```

  - Restart the ProcurementCluster of the ProcurementDomain.
- For the ProjectDomain, make these changes to the `$ProjectDomain/bin/fusionapps_start_params.properties` file:
  - Edit `fusion.ProjectsFinancialsCluster.SunOS-sparc.memoryargs` and add these arguments:

```
-XX:ReservedCodeCacheSize=128m -XX:+UseParallelOldGC
-XX:ParallelGCThreads=4
```

  - Restart the ProjectFinancialCluster of the ProjectDomain.

Solaris x64

- In *all* domains, make these changes to the *DOMAIN_HOME*/bin/fusionapps_start_params.properties file:
  - Edit fusion.default.SunOS-i386.memoryargs to add these arguments:

```
-XX:ReservedCodeCacheSize=128m -XX:+UseParallelOldGC
-XX:ParallelGCThreads=4
```

  - Restart all the servers of all the domains.
- In *all* domains, make these changes to the *DOMAIN_HOME*/bin/fusionapps_start_params.properties file:
  - Add the fusion.AdminServer.SunOS-i386.memoryargs entry with these arguments:

```
-XX:PermSize=256m -XX:MaxPermSize=512m -XX:+UseParallelGC
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=@HEAP_DUMP_PATH@
-XX:+ParallelGCVerbose -XX:ReservedCodeCacheSize=128m
-XX:+UseParallelOldGC  -XX:ParallelGCThreads=2
```

  - Change @HEAP_DUMP_PATH@ to a valid path for the Heap Dump files to be written.
  - Restart the Admin servers of all the domains.
- For the BIDomain, make these changes to the $BIDomain/bin/fusionapps_start_params.properties file:
  - Edit fusion.default.SunOS-i386.memoryargs and append these arguments:

```
-XX:ReservedCodeCacheSize=128m -XX:+UseParallelOldGC
-XX:ParallelGCThreads=4
```

  - Restart all the servers of the BIDomain.
- For the BIDomain, make these changes to the $BIDomain/bin/fusionapps_start_params.properties file:
  - Add the fusion.AdminServer.SunOS-i386.memoryargs entry with these arguments:

```
-XX:PermSize=256m
-XX:MaxPermSize=512m -XX:+UseParallelGC  -XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=@HEAP_DUMP_PATH@ -XX:+ParallelGCVerbose
-XX:ReservedCodeCacheSize=128m -XX:+UseParallelOldGC
-XX:ParallelGCThreads=2
```

  - Change @HEAP_DUMP_PATH@ to a valid path for the Heap Dump files to be written.
  - Restart the Admin server of the BIDomain.
- For the CommonDomain, make these changes to the $CommonDomain/bin/fusionapps_start_params.properties file:

  - Edit `fusion.HelpPortalCluster.SunOS-i386.memoryargs` to add these arguments:

    ```
    -XX:ReservedCodeCacheSize=128m -XX:+UseParallelOldGC
    -XX:ParallelGCThreads=4
    ```

  - Restart the HelpPortalCluster of the CommonDomain.

- For the ProcurementDomain, make these changes to the `ProcurementDomain/bin/fusionapps_start_params.properties` file:

  - Edit fusion.ProcurementCluster.SunOS-i386.memoryargs and add these arguments:

    ```
    -XX:ReservedCodeCacheSize=128m -XX:+UseParallelOldGC
    -XX:ParallelGCThreads=4
    ```

  - Restart the ProcurementCluster server of the ProcurementDomain.

1.4 Tuning Oracle HTTP Server

This section lists several Oracle HTTP Server configuration changes that may improve performance. These settings are the default if your environment is newly provisioned. If your environment in upgraded, you will have to manually apply these setting changes.

Follow these steps to tune the Oracle HTTP Server.

- Avoid restarts of httpd-worker processes by increasing MaxSpareThreads and MinSpareThreads.

  These restarts affect the recreation of connections and threads in Oracle HTTP Server processes during varying load patterns and could negatively affect performance. The recommendation is to increase the minimum and maximum spare threads for Oracle HTTP Server to 200 and 800 respectively.

  Edit the `httpd.conf` file located in `instance_home/config/OHS/<ohs_name>/httpd.conf` so it resembles:

  ```
  <IfModule mpm_worker_module>
  ...
  MinSpareThreads 200
  MaxSpareThreads 800
  ...
  </IfModule>
  ```

- Increase the Keep Alive timeout between Oracle HTTP Server and WebLogic Servers.

  By default, a connection between an Oracle HTTP Server and a WebLogic Server is closed if it is idle for 20 seconds.

  Since there is a cost in re-establishing these connections, it is beneficial to increase this timeout to 5 minutes. To do this, the `mod_wl_ohs.conf` and `config.xml` files for the target WebLogic Server domains need to be changed.

  Edit the `<instance home>/config/OHS/<ohs name>/mod_wl_ohs.conf` file so it resembles this example:

  ```
  LoadModule weblogic_module ${ORACLE_HOME}/ohs/modules/mod_wl_ohs.so

  <IfModule weblogic_module>
    ...
  ```

```
        KeepAliveSecs 300
    ...
    </IfModule>
```

Edit the WebLogic Server configuration.

- Login to the WebLogic Server console of the target domain.
- Select **Environment > Servers**.
- Select the server you want to change.
- Select the Protocols tab and then the HTTP subtab.
- Change both the Duration and HTTPS Duration to 300.

- Disable compression for .swf files.

  Add this line to the `<instance home>/config/OHS/<ohs name>/moduleconf/mod_deflate.conf` file.

  ```
  SetEnvIfNoCase Request_URI \.(swf)$ no-gzip dont-vary
  ```

  The .swf files are already compressed; there is no need for Oracle HTTP Server to compress them again.

- Set the Expires header for Business Intelligence static resources. Adding the HTTP Expires header for static resources related to business intelligence improves performance by allowing the browser to locally cache those artifacts instead of making repeated requests for them.

  Edit the *instance home*`/config/OHS/`*ohs name*`/moduleconf/FusionVirtualHost_bi.conf` file and add the entries noted in Example 1–1.

Example 1–1 Additions to FusionVirtualHost_bi.conf

```
#Internal virtual host for BI, replace biprove.domain.host and
biprov.wls.managed.port with appropriate values

<VirtualHost <OHS_INTERNAL_HOST>:<INTERNAL_PORT> > ServerName http://<INTERNAL_
ENDPOINT_HOST>:<INTERNAL_ENDPOINT_PORT>

       RedirectMatch 301 ^/analytics$ /analytics/
       # Add the following line
       RedirectMatch 301 ^/analytics/res$ /analytics/res/

       <LocationMatch ^/analytics/>
              SetHandler weblogic-handler
              WeblogicCluster <biprov.domain.host>:<biprov.wls.managed.port>
       </LocationMatch>

       # Add the following section
       <LocationMatch ^/analytics/res/>
              SetHandler weblogic-handler
              WeblogicCluster <biprov.domain.host>:<biprov.wls.managed.port>
              SetOutputFilter DEFLATE
              SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip dont-vary
              ExpiresActive on
              ExpiresDefault "access plus 1 weeks"
              Header set Cache-Control "Public"
   </LocationMatch>

#External virtual host for BI
```

```
<VirtualHost <OHS_EXTERNAL_HOST>:<EXTERNAL_PORT> > ServerName https://<EXTERNAL_
ENDPOINT_HOST>:<EXTERNAL_ENDPOINT_PORT>

   RedirectMatch 301 ^/analytics$ /analytics/
   # Add the followig line
   RedirectMatch 301 ^/analytics/res$ /analytics/res/

   # BI EE
   <LocationMatch ^/analytics/>
      SetHandler weblogic-handler
      WeblogicCluster <biprov.domain.host>:<biprov.wls.managed.port>
                WLProxySSL ON
                WLProxySSLPassThrough ON
                RewriteEngine ON
                RewriteOptions inherit
   </LocationMatch>

   # Add the following section
   <LocationMatch ^/analytics/res/>
           SetHandler weblogic-handler
           WeblogicCluster <biprov.domain.host>:<biprov.wls.managed.port>
                WLProxySSL ON
                WLProxySSLPassThrough ON
                RewriteEngine ON
                RewriteOptions inherit
           SetOutputFilter DEFLATE
           SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip dont-vary
           ExpiresActive on
           ExpiresDefault "access plus 1 weeks"
           Header set Cache-Control "Public"
   </LocationMatch>
```

- Add the favicon.ico file to the online help system document root.

  Most browsers will attempt to access the `favicon.ico` (favorite icon) file from the root of a web site. The favorite icon is displayed next to the browser's address bar.

  If the file does not exist, the browser will attempt to fetch it the next time the browser is started. If the file exists, it will be cached in the browser's cache.

  Oracle Fusion Applications does not ship with a `favicon.ico` file. If you inspect the online help system access log, you will see multiple requests for `/favicon.ico` that result in the 404 File Not Found error response.

  You can avoid many of these hits by placing a `favicon.ico` file in the location specified by the `DocumentRoot` location in the online help system configuration file. The icon file must be 16x16 pixels in Windows Icon (.ico) file format.

  By default, the DocumentRoot is configured to be `$`*ORACLE_INSTANCE*`/config/$COMPONENT_TYPE/$COMPONENT_NAME/htdocs`. You can put a `favicon.ico` file at that location. If you have multiple online help system instances, you should put the file in the `DocumentRoot` location of each instance.

- Remove unneeded OHS configuration rules for Internet Explorer.

  If there are many users over WAN using Internet Explorer, the default settings in Oracle Fusion Applications may result in poor performance due to constant re-establishment of HTTP connections.

To avoid the problem, edit these three files:

- <APPTOP>/instance/CommonDomain_webtier<n>/config/OHS/<ohs_name>/FusionSSL.conf
- <APPTOP>/instance/CommonDomain_webtier<n>/config/OHS/<ohs_name>/ssl.conf
- <IDMTOP>/config/instance/ohs1/config/OHS/<ohs_name>/ssl.conf

and remove these entries:

- BrowserMatch ".*MSIE.*" \
- nokeepalive ssl-unclean-shutdown \
- downgrade-1.0 force-response-1.0

2

Troubleshooting Process

This chapter discusses the basic process to use to troubleshoot Oracle Fusion Applications, and presents specific steps for the most commonly-seen problems.

This chapter includes these sections:

2.1 Introduction to Troubleshooting

This section provides guidelines and a process for using the information in this chapter. Using the following guidelines and process will focus and minimize the time you spend resolving problems.

Guidelines

When using the information in this chapter, Oracle recommends:

- After performing any of the solution procedures in this chapter, immediately retrying the failed task that led you to this troubleshooting information. If the task still fails when you retry it, perform a different solution procedure in this chapter and then try the failed task again. Repeat this process until you resolve the problem.
- Making notes about the solution procedures you perform, symptoms you see, and data you collect while troubleshooting. If you cannot resolve the problem using the information in this chapter and you must log a service request, the notes you make will expedite the process of solving the problem.

Process

Follow the process outlined in Table 2–1 when using the information in this guide. If the information in a particular section does not resolve your problem, proceed to the next step in this process.

Table 2–1 Process for Resolving Performance Issues

Step	Section to Use	Purpose
1	Chapter 1, "Monitoring and Tuning Oracle Fusion Applications"	Collect symptoms about the performance problem to determine if the problem is related the following: ■ Response time or throughput ■ Widespread or limited to specific users and flows
	Oracle Fusion Applications Administrator's Guide: Chapter 9 and Chapter 10 for locating key metrics Chapter 12 for monitoring the middle tier Chapter 13 for monitoring and tuning the Oracle database	Determine what changed since the system was last performing well.
2	Section 2.1.1 through Section 2.1.7	Use Section 2.1.1 if the problem is widespread. Otherwise, review the problem description in Section 2.1.2 through Section 2.1.7 to if there is a match These sections describe: ■ Possible causes of the problems ■ Solution procedures corresponding to each of the possible causes
4	Section 2.3	Use My Oracle Support to get additional troubleshooting information about Oracle Fusion Applications or performance, scalability, and reliability. My Oracle Support provides access to several useful troubleshooting resources, including Knowledge Base articles and Community Forums and Discussions.
5	Section 2.3	Log a service request if the information in this chapter and My Oracle Support does not resolve your problem. You can log a service request using My Oracle Support at `https://support.oracle.com`.

2.1.1 How to Troubleshoot Overall System Slowness

To troubleshoot overall system slowness:

- Navigate to the Fusion Applications target in Enterprise Manager.
- Select the application that is having the problem, such as Payables.
- Navigate to the product home page, such as the Payables product home page.
- Select **Monitoring > System Performance** to open the System Performance page.
- Click the name of the target metric name for **Request Processing Time**. A popup is displayed.
- Click **Problem Analysis** to see the related metrics in the problem analysis page.

  You should be able to see the Source Metric and Related Metrics in the Problem Analysis page.
- Check if any key metrics are beyond threshold, or look for sudden changes.

In particular, check the **heap usage** metrics.

If the heap is constantly close to 100 percent, search using the string OutOfMemoryErrors in the Oracle WebLogic Server server_name.out in the following directories:

```
(UNIX) DOMAIN_HOME/servers/server_name/logs
(Windows) DOMAIN_HOME\servers\server_name\logs
```

If there are OutOfMemoryErrors, a heap dump would have been generated in the directory specified by the `-XX:HeapDumpPath` parameter from the Oracle WebLogic Server startup JVM option. Submit the heap dump to Oracle Support for further analysis of what is retaining memory. In many cases a bounce is needed to resolve the issue.

- Scroll down and you will be able to see the Related Targets at the bottom of the page.
- Select the Related Targets Topology to view the Topology of the target on which the metric is being analyzed.
- Review incidents and logs by clicking the **View Related Log Messages** link at the top right side of the page. The target context, the chart time duration and related targets are chosen to do log search.
- Follow the appropriate triage process depending on which metric is in question.
- Check the log level to make sure it is not set to FINEST.

  Follow the steps listed in "Configuring Settings for Log Files During Normal Operation" in the *Oracle Fusion Applications Administrator's Guide* to review the log level settings.
- Review the results of the Automatic Database Diagnostic Monitor (ADDM) analysis. See Section 3.2.10, "How to Review Automatic Database Diagnostic Monitor Reports Regularly."

2.1.2 How to Find the Slowest JSPX/JSFF Pages

An administrator would want this information to understand which part of the application has poor performance.

To find the slowest JSPX/JSFF pages:

- Navigate to **Middleware Targets** in Oracle Enterprise Manager Fusion Applications Control.
- Select **Application Dependency and Performance** from the drop-down menu.
- Expand the ADF node in the tree on the left hand side of the screen.
- Click JSF pages and sort by response time.
- Look for the slowest page name, and find that page name in the tree on the left. Click to drill down.
- If applicable, it will show the managed beans for that page.

2.1.3 How to Troubleshoot Slow UI Request

A sluggish UI response directly affects each user's experience. Administrators will want to use the suggested troubleshooting methods in this section to track down and fix slow UI response.

2.1.3.1 Troubleshooting Historical Requests

These steps are applicable to requests that have completed.

- Find Execution Context Identifiers (ECIDs) submitted by the user and find the slowest requests using server logs. Compare response time as recorded in Oracle HTML Server (OHS) and WebLogic Server access logs. If there is a big gap, check OHS health. Otherwise, look at request details in WebLogic Server.

  Find the nodes that serviced the request by searching for the ECID using Request Monitor in Enterprise Manager.

  Drill down into the JVM Diagnostics screen.

  The data displayed should be in the context of that ECID. Observe the thread state transitions and check the call stacks to find the top methods and top SQL statements.

- Check if slowness is caused by customized code.
- If slowness is caused by lock contention, check the thread that is holding the lock and what it is doing.
- If the request is slow enough to cause a STUCK thread, an incident should have been generated. Search the log file for the incident number The incident directory would have a WebLogic Server diagnostic image which includes a JRockit Flight Recorder (JFR) recording from running WebLogic Server.
- If there are not enough details in the JVM Diagnostic data, follow instructions in Section 2.2, "View Detailed Timing Of a Request Using a JRockit Flight Recorder (JFR) File" to extract and view JFR recordings.

2.1.3.2 Using Automatic Incident to Capture Diagnostic Information for Slow Requests

Fusion Applications can be configured to automatically generate an incident when requests slower than a specified threshold are detected.

Two modes are available:

- Generate an incident whenever a slow request is detected.
- Generate an incident at fixed intervals if slow requests are detected in the previous interval. With this mode, even if there is a flood of slow requests in a given time period, only one incident will be generated.

  The incident will have JFR recording and the ECID of the slow requests that should provide information about where time was spent. Follow the steps in Section 2.2, "View Detailed Timing Of a Request Using a JRockit Flight Recorder (JFR) File" on how to view detailed timing of a request using the JFR recording.

  These Oracle WebLogic Scripting Tool commands can be used to configure the feature:

  – Generate an incident on detection of any request that takes longer than 20 seconds.

    ```
    updateDMSEventDestination(id='HTTPRequestTrackerDestinatio
    n', props={'requestThresholdSeconds' : '20'},
    server="<server>")
    ```

  – Generate an incident on detection of any request that takes longer than 20 seconds during the past 15 minutes.

```
updateDMSEventDestination(id='HTTPRequestTrackerDestinatio
n', props={'requestThresholdSeconds' : '20',
'generateIncidentMinutes' : '15'}, server="<server>")
```

2.1.3.3 Troubleshooting Live Requests

When there is a performance issue that is reproducible, this end-user tracing feature makes it easier for the end user who is experiencing the problem to record a trace that will contain the necessary diagnostic information.

To create the trace, follow the steps in Tracing Options in "How to Use the Troubleshooting Options" in the *Oracle Fusion Applications Developer's Guide*.

2.1.3.4 Troubleshooting Stuck Threads

Problem

Stuck threads may result if the server is nearing out of memory. If the server is close to out of memory, all requests should slow down. To resolve an out-of-memory issue, see Section 2.2, "View Detailed Timing Of a Request Using a JRockit Flight Recorder (JFR) File."

If a request is taking longer than 10 minutes, the stuck thread is reported to Oracle WebLogic Server *server_name*.out in the following directories:

```
(UNIX) DOMAIN_HOME/servers/server_name/logs
(Windows) DOMAIN_HOME\servers\server_name\logs
```

For example:

```
<Mar 4, 2011 7:44:08 AM PST> <Error> <WebLogicServer> <BEA-000337> <[STUCK]
ExecuteThread: '19' for queue: 'weblogic.kernel.Default (self-tuning)' has been
busy for "600" seconds working on the request
"weblogic.servlet.internal.ServletRequestImpl@18986012[
GET /productManagement/faces/PimDashboardUiShellPage?_afrLoop=1398820150000&_
afrWindowMode=0&_adf.ctrl-state=a44e7uxcc_13 HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, application/x-ms-application,
application/x-ms-xbap, application/vnd.ms-xpsdocument, application/xaml+xml,
application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: fr
UA-CPU: x86
...
]", which is more than the configured time (StuckThreadMaxTime) of "600" seconds
. Stack trace:
Thread-164 "[STUCK] ExecuteThread: '19' for queue: 'weblogic.kernel.Default
(self-tuning)'" <alive, in native, suspended, priority=1, DAEMON> {
    jrockit.net.SocketNativeIO.readBytesPinned(SocketNativeIO.java:???)
    jrockit.net.SocketNativeIO.socketRead(SocketNativeIO.java:24)
    java.net.SocketInputStream.socketRead0(SocketInputStream.java:???)
    java.net.SocketInputStream.read(SocketInputStream.java:107)
...
```

In this example, the request has been running longer than the configured 600 seconds. Here is the associated stack trace showing the thread is stuck:

```
Thread-164 "[STUCK] ExecuteThread: '19' for queue: 'weblogic.kernel.Default
(self-tuning)'" <alive, in native, suspended, priority=1, DAEMON> {
jrockit.net.SocketNativeIO.readBytesPinned(SocketNativeIO.java:???)
jrockit.net.SocketNativeIO.socketRead(SocketNativeIO.java:24)
java.net.SocketInputStream.socketRead0(SocketInputStream.java:???)
```

```
java.net.SocketInputStream.read(SocketInputStream.java:107)
...
```

Solution

If the stack shows the thread is waiting for a response from another server, check the status of the other server and see it has performance problems before proceeding with this solution.

To determine what the stack thread was doing prior to becoming stuck, perform the following steps:

1. Look at the next few log messages in *server_name*.out for a message indicating an incident has been created. For example:

   ```
   <Mar 4, 2011 7:44:10 AM PST> <Alert> <Diagnostics> <BEA-320016>
   <Creating diagnostic image in DOMAIN_HOME/servers
   /ProductManagementServer_1/adr/diag/ofm/SCMDomain/
   ProductManagementServer_1/incident/incdir_394 with a lockout minute
   period of 1.>
   ```

   The above message may not always appear after each stuck thread reported. It is printed at most four times an hour. If the message does not appear, manually look for the incident directory by checking the readme file in the subdirectories under the following directories:

   ```
   (UNIX) DOMAIN_HOME/servers/server_name/adr/diag/ofm/domain_name/server_
   name/incident
   (Windows) DOMAIN_HOME\servers\server_name\adr\diag\ofm\domain_name\server_
   name\incident
   ```

   The incident directory contains a WLDF diagnostic image which contains the JFR recording, and a file containing the thread dump.

2. Review thread dump to see call stack of the thread. If thread is blocked waiting for lock, check what the thread holding the lock is doing.

3. If call stack involves executing JDBC calls, you can go to Grid Control and check the top activity around that time window, and see if there is a session with a matching module and action. See Section 4.2.5, "Finding the Top SQL Queries."

4. Review the JRockit flight recording file JRockitFlightRecorder.jfr for more details. You will also need the ECID of the request which is recorded in the readme.txt file of the incident directory, and also the Oracle WebLogic Server log.

5. Perform the tasks in Section 2.2, "View Detailed Timing Of a Request Using a JRockit Flight Recorder (JFR) File."

Since the ECID of the request that caused the stuck thread is recorded in the error message, you also can follow the steps for troubleshooting slow requests that already have completed as documented in Section 2.1.3.1, "Troubleshooting Historical Requests."

2.1.3.5 Troubleshooting Slow Requests Using JFR Recording

See Section 2.2, "View Detailed Timing Of a Request Using a JRockit Flight Recorder (JFR) File."

2.1.3.6 Troubleshooting Memory Leaks and Heap Usage Pressure

Problem

Application performance degrades over time, heap usage and garbage collection activity increases overtime, sometimes `OutOfMemoryErrors` are seen.There could be memory leaks in the application, which causes the amount of free memory in the JVM to continuously decrease.

Solution

To solve this problem, perform the following:

1. Review the `server_name.out` file for `OutOfMemoryErrors` errors, which indicate a heap dump file has been written. The `server_name.out` file is located in the following directories:

   (UNIX) *DOMAIN_HOME*/servers/*server_name*/logs
   (Windows) *DOMAIN_HOME*\servers*server_name*\logs

2. Restart the Managed Server.

   See the following documentation resources to learn more about other methods for starting and stopping the Managed Servers:

   - "Starting Managed Servers with a Startup Script" section in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.
   - "Starting Managed Servers with the java weblogic.Server Command" section in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.
   - "Starting and Stopping Oracle Fusion Middleware" section in *Oracle Fusion Middleware Administrator's Guide*.
   - "Start and Stop Servers" and various startup and shutdown procedures in the Cluster section of the *Administration Console Online Help*.

   If the problem persists, proceed to Step 3.

3. Open the file with a heap-dump analysis tool that can handle binary `HPROF` format, such as Eclipse Memory Analyzer.
4. Review what objects and classes are retaining most memory. Send the heap dump file to Oracle Support for further analysis.
5. Sometimes it may be necessary to take several heap dumps to see what objects or classes are consuming and increasing the amount of memory.

   To take heap dumps on demand, use the `jrcmd` command-line tool. See the "Running Diagnostic Commands" chapter in the *Oracle JRockit JDK Tools Guide*. Many heap dump analysis tools, such as Eclipse Memory Analyzer, enable you to compare two heap dumps to identify memory growth areas.

   Heap dumps provide information on why memory is retained. Sometimes it is necessary to know how memory is allocated to further resolve the issue. For these cases, proceed to Step 6.

6. Use the JRockit Memory Leak Detector tool that is part of JRockit Mission Control Client to understand how memory is allocated.

   For more information, see the JRockit Mission Control online help.

2.1.4 How to Troubleshoot Slow Batch Jobs

Batch jobs are often critical parts of key business processes. If throughput or execution time is beyond expectation, these steps may help diagnose the root cause.

2.1.4.1 Troubleshooting Jobs That Are in Wait/Ready/Blocked State for a Long Time

- Launch Oracle Enterprise Manager Cloud Control and look up the WebLogic Server domain target, or launch Fusion Applications Control for the domain.
- Expand Scheduling Service and select the one to which the job is submitted.
- Select **Job Request > Search Job Request**.
- Enter the appropriate search criteria.
- Expand the display to the request details. The top of the page will have an information block explaining why the job is not yet run.
- If many jobs are in a wait state, check **Performance > Historical Reports > View: Requests Metrics By Work Assignment**.
- Inspect Wait Time versus Processing Time.
  - If Wait Time is high, check several jobs and see why they are in a wait state.
  - If there is spare CPU capacity on the servers where the jobs are running, consider adding more threads to the work assignment.

2.1.4.2 Troubleshooting Jobs in Running State for a Long Time

See the "Managing Logging for Oracle Enterprise Scheduler" section in the *Oracle Fusion Applications Administrator's Guide*.

- Sometimes a job could be in a running state even if it had completed. See the ESS Troubleshooting Guide to first determine if the job is still running.
- Otherwise, open the request details page and click **View Log Message** to get the ECID. Go to the Java Diagnostics for the cluster to which the ESS job sends the request, such as for service calls, search for the ECID and see what, if any, Java diagnostics data is recorded.

2.1.4.3 Troubleshooting Slow BI Publisher Jobs

See the "Managing Logging for Oracle Enterprise Scheduler" section in the *Oracle Fusion Applications Administrator's Guide*.

- Open the home page of the Oracle Fusion instance, product family, or product in Cloud Control. The **Top Long Running Job Requests** and **Recently Completed Job Requests** region shows the health of ESS jobs. Since BI Publisher jobs are scheduled as ESS jobs, the health of those jobs will be shown in this region.
- Click the request ID link to show the Request Detail page.
- Select **Actions > Request Log** in the right hand side drop down and look for the log entries specific to this job request.
- Select **Actions > JVM Diagnostics** in the right hand side drop down and view the health of this job request from the JVM perspective.
  - The JVM page shows the data pertaining to the ECID corresponding to this job request.
  - Inspect thread, cpu, memory and database diagnostic data for this specific job.

2.1.4.4 Troubleshooting Slow SOA Jobs

- Open the home page of the Oracle Fusion product family in Cloud Control. The **Recently Completed SOA Instances** region shows the health of recently-completed SOA jobs in the product family context.
- Click **Composite Name** to show the detailed information about the composite in the home page.
- Click **Dehydration Diagnostics** to view the overall database activity associated with the dehydration store and any abnormal bottlenecks. To view database diagnostics, click individual SQL IDs.
- Click **Faults and Rejected Message** to view more details about faulted SOA instances.
- Select **SOA Composites > Logs > View Log Messages** to view the related log entries.
- Select **SOA Composites > Trace Instance** to trace a particular SOA instance. To display all SOA instances, click **Search** without specifying a filter.
- Click the instance ID to view the complete trace details for this instance.
- Click **JVM Diagnostics** to view the health of this SOA instance from a JVM perspective.
  - The JVM page shows the data pertaining to the ECID corresponding to this job request.
  - Inspect thread, cpu, memory and database diagnostic data for this specific job.

2.1.4.5 Troubleshooting Slow Oracle Enterprise Scheduler Jobs of SQL Type

When the user submits a SQL job type, the job remains in a `RUNNING` state for too long.

Solution

To resolve this problem, perform the following steps:

1. Use Fusion Applications Control to find the database session ID that was used to process the job:
   a. Search for the request, as described in the "Searching for Oracle Enterprise Scheduler Job Requests" section in the *Oracle Fusion Applications Administrator's Guide*.
   b. On the Request Details page, in the **Request Properties** section, next to the **Execution Type** field, click the eye glasses icon.

      The Spawned Process Details dialog is displayed. This will open a pop-up with the database session ID that was used to process this job
   c. Take note of the value in the **Session Id** field, and then click **OK**.
2. Use Grid Control to ensure the request processor and request dispatcher are running:
   a. Run an Active Session History (ASH) report for the session within the relevant time window to inspect top SQL statements and top wait events.

      To run ASH reports using Oracle Enterprise Manager:

      - On the Performance page, under Average Active Sessions, click **Run ASH Report**.

The Run ASH Report page appears.

- Enter the date and time for the start and end of the time period when the transient performance problem occurred.

Click **Generate Report**.

The Processing: View Report page appears while the report is being generated.

After the report is generated, the ASH report appears under Report Results on the Run ASH Report page.

- Optionally, click **Save to File** to save the report in HTML format for future analysis.

b. Identify time consuming SQL statements and tune following normal SQL tuning procedures. See Section 4.2.5, "Finding the Top SQL Queries."

2.1.4.6 Troubleshooting Slow Java Jobs

- Open the request details page and click **View Log Message** to get the ECID.
- Open the Java Diagnostics for the cluster to which the ESS job sends the request, such as for service calls, or the ESS server itself if most of the logic executes there.
- Search for the ECID to view any Java Diagnostics data that is recorded.
- If additional trace data is needed, and the job can be re-submitted, follow the steps listed in the "Using SQL Tracing" section in the *Oracle Fusion Applications Administrator's Guide* to enable SQL trace and resubmit the job. Review the collected SQL trace files for analysis.

2.1.5 How to Troubleshoot a Slow BPEL Instance

- From Fusion Applications Control, select the SOA instance under SOA.
- Select **Instances** and search for the instance in question.
- Click the **Instance** link to display the flow trace to obtain the ECID.
- Click the BPEL component to see the audit trail that records the completion time for each step.
- Look for big gaps in the timing.

  If a gap is due to a service call, use the request monitor and search for the ECID, check JVMD or JRockit Flight Recorder (JFR) to determine why it took so long.

2.1.6 How to Troubleshoot High Connection Usage

Problem

The connection usage on the Oracle Database is high, or there is an Oracle process on the database host consuming high amount of CPU.

Solution

To find out the source of the connection causing the high CPU on To adjust the reference pool size from Fusion Applications Control:

1. Oracle Fusion Applications set values on a number of `v$session` attributes to indicate how the connection is being used. When looking at a connection consuming high CPU on the database, or when trying to understand what

connections are used for what processes, inspect the value of these attributes as follows:

Attribute in v$session	Value Being Set
`Process`	Data Source Name (for example, ApplicationDB)
`Program`	Oracle WebLogic Server Domain plus the Managed Server name, prefixed by DS (for example, `DS/FinancialDomain/AccountsReceivableServer_1`)
`Module`	Oracle Application Development Framework: ADF BC application module name Oracle Enterprise Scheduler: ■ Java job type: Class name, except `oracle.apps` ■ PLSQL: the package and procedure name (for example, `mypkg.myproc`) ■ Other jobs: Static: Executable name Oracle BI Publisher: Name of the report
`Action`	Oracle Application Development Framework: jspx name Oracle Enterprise Scheduler: Job definition name Oracle BI Publisher, if request is submitted: ■ Oracle Enterprise Scheduler: Oracle Enterprise Scheduler job definition name ■ Oracle BI Publisher Scheduler Job: Oracle BI Publisher job name submitted by the user ■ Oracle BI Publisher online: Static string `BIP:Online` ■ Oracle BI Publisher Web services: Name of the web services
`Client_Identifier`	Application User Name

2. If the error messages related to connection pool capacity being reached are also seen in Oracle WebLogic Server logs, use the solution for connection leaks described in Section 2.1.6.1.

2.1.6.1 Troubleshooting Connection Leaks

Problem

When there are errors in the log, and the error message indicates connection pool size has been reached

Solution

To resolve this problem:

1. When data source is at maximum capacity and there are errors during connection reservation requests, then there may be connection leaks in the code
2. Enable JDBC profiling from the Oracle WebLogic Server Administration Console:
   a. In the **Domain Structure**, expand **Services** and then **Data Sources**.
   b. Click the data source that needs to profiled, such as **ApplicationDB**.

c. In the Settings page, click the **Configuration** tab, then click the **Diagnostics** subtab.

d. Check the profiles that need to be collected (**PROFILE_TYPE_CONN_USAGE_STR**).

e. Click **Save**.

3. Configure the diagnostic archive where the profiling data is saved from the Oracle WebLogic Server Administration Console:

   a. In the **Domain Structure**, expand **Services**, **Diagnostics**, and then **Archives**.

   b. Click the server where you want to make changes (archives are stored for each server)

   c. In the Settings page, you can change archive location, size and how to retire data.

   d. Check the profiles that need to be collected (**PROFILE_TYPE_CONN_USAGE_STR**).

   e. Click **Save**.

4. To retrieve profiling data, use the sample code (`http://download.oracle.com/docs/cd/E15051_01/wls/docs103/wldf_configuring/access_diag_data.html#wp1100898`), with changes to the URL, username and password in the initialize method.

5. Run the sample code as a standalone program.

6. The program will capture the stack trace for each request for a connection from that data source. Inspect the callers to see the suspicious stack. This sample program requires connecting to a live Oracle WebLogic Server instance.

   The diagnostic archive file under the archive location can also be provided to Oracle Support for further analysis.

   Oracle WebLogic Server will not report a leak unless inactive connection timeout connection pool setting is set to a positive value. This cannot be done for Oracle Fusion Applications, as it will break functionality.

2.1.6.2 Troubleshooting Excessive Activation

Problem

When response time suddenly increases with rising user count, even though there is no memory pressure, it is possible that the reference pool size for key application modules needs to be increased. If there is a JFR recording to review, and you observe many events containing callstacks containing the `activateState` method, you should also try adjusting the reference pool size.

Solution

To adjust the reference pool size from Fusion Applications Control:

1. Review the number of web sessions from Performance Summary pages:

   For Fusion Applications Control:

   a. From the navigation pane, expand the farm, **Application Deployments**.

   b. From the **Applications Deployments** page, select the application.

c. From the **Application Deployments** menu, choose **ADF** > **ADF Performance**.

The ADF Performance page is displayed.

d. Click the **Application Module Pools** tab.

e. Sort the request by descending order.

f. For the top 10 or so application modules, click the application module name to view the **Activations** count.

For Grid Control:

a. Click the **Targets** tab.

b. Click the **Middleware** secondary tab.

c. From the **Search** list, select **Oracle WebLogic Server Domain**, and then click **Go**.

d. Select a domain.

The WebLogic Server Domain home page is displayed.

e. In the table on the right-hand side of the page, expand the **Application Deployments** node.

f. Click the target application.

The Application Deployment page is displayed.

g. From the **Application Deployments** menu, choose **ADF** > **ADF Performance**.

The ADF Performance page is displayed.

h. In the **Application Module Pools** table, from the **View** list, select **Total Requests**, and once selected, from the **Total Requests** column, click **Sort Descending**.

i. For the top 10 or so application modules, click to see the details of each one.

j. After selecting an application module, on the **Requests** graph, from the **Select metric to display in chart** list, select **Passivation** and **Activation** to add to the graph.

If the activation count is close to passivation and is constantly above 0, then follow Step 2 to adjust.

2. If the activation count constantly increases, increase the application module reference pool size from Fusion Applications Control:

a. From the **Application Deployments** menu, choose **ADF** > **Configure ADF Business Components**.

The ADF Configuration BC Configurations page is displayed.

b. From the **Application Modules** section, click the application module of interest. From the left hand side, select the local by selecting a name that ends in **Local**.

c. Click the **Pooling and Scalability** tab, and change the **Reference Pool Size** parameter.

2.1.7 How to Troubleshoot Oracle Business Intelligence

The usual indication that you should troubleshoot Oracle BI will be sluggish performance of BI components embedded across various applications. BI components

can be in the form of such things as charts, tables, dashboards, and queries. Many of the configuration issues can be detected from the `nqquery.log` and `nqserver.log` log files.

A configuration problem generally will produce an error message. You would examine the same two log files for either sluggish response or an error.

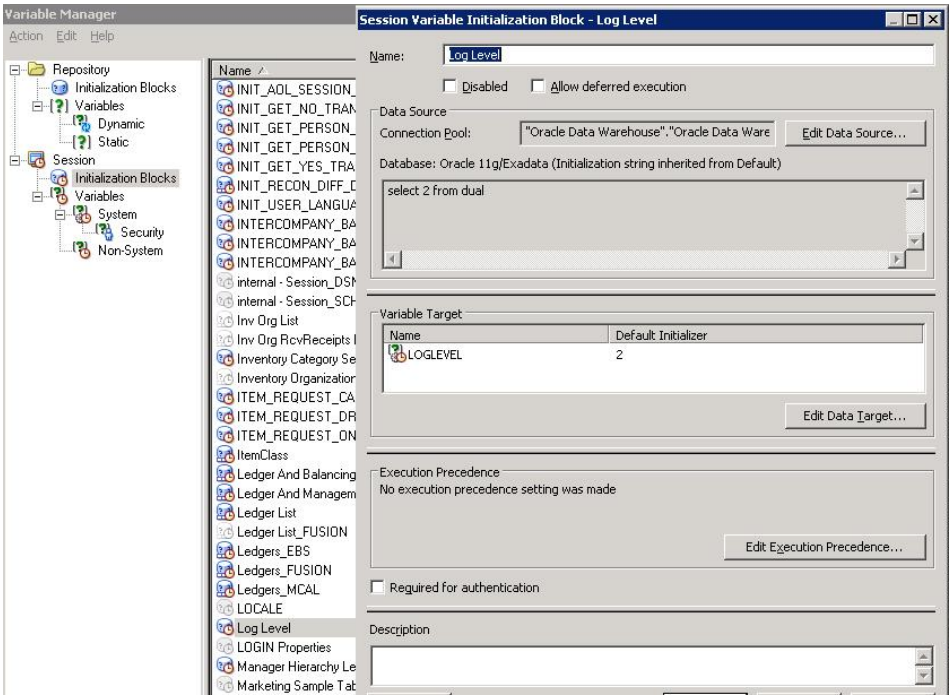
Oracle BI Query Logging

To debug a query issue, you need to enable query logging. Query logging can be enabled by setting the LOGLEVEL variable to a value from 0 through 7. 0 denotes no logging, and 7 will generate detailed logging.

It is sufficient to use loglevel 2 to obtain logical BI server queries and the corresponding physical database queries in the log file.

Figure 2–1 shows how to use the Variable Manager to set the log level to 2.

Figure 2–1 Setting the Log Level for Oracle BI



Once you have enabled logs, you can obtain log files from `$ORACLE_INSTANCE/diagnostics/logs/OracleBIServerComponent/coreapplication_obis1/`

Relevant files are:

- `nqquery.log` - Will contain logical SQL, physical SQL and an execution plan chosen by OBIS for the logical SQL.
- `nqserver.log` - Will contain server-related data, such as initialization block-related errors.

Note that a single logical SQL can spawn multiple physical SQL statements.

If you run large queries and logs are being rotated quickly, you can modify parameters in `logconfig.xml` to control the frequency for log rotation. `logconfig.xml` is located at `$ORACLE_`

`INSTANCE/config/OracleBIServerComponent/coreapplication_obis1/logconfig.xml` where ORACLE_INSTANCE is a path similar to `/u47/st99/instance/BIInstance`.

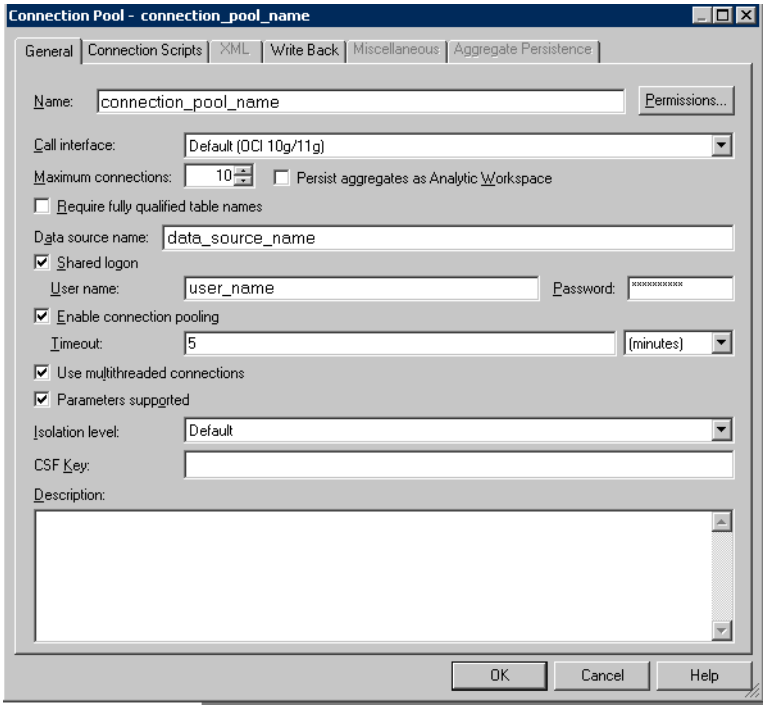
Once you have logging enabled, you can search for response time and identify where time is being spent by the query. In this example of a useful search string, logical query execution took 1 second, and physical query execution also took 1 second.

```
Physical query response time 1 (seconds)
Logical Query Summary Stats: Elapsed time 1, Response time 1, Compilation time 0
(seconds)
```

BI Connection Pool Settings

If you anticipate a higher load on a system, you can change the number of **Maximum connections** for various data sources to make resource use more efficient, as shown in Figure 2–2. This change must be propagated if the location of the Business Intelligence metadata is replaced.

Figure 2–2 Changing the Maximum Number of Connections



2.2 View Detailed Timing Of a Request Using a JRockit Flight Recorder (JFR) File

Problem

Certain requests are slow and there is a need to find out where time is spent

Solution

The JRockit Flight Recorder (JFR) file contains a record of various events that consume time, and can be used to help understand why a request is taking time

To resolve this problem, create a JFR file:

1. Extract a JFR file from an Oracle WebLogic Server server by running the following command:

   ```
   (UNIX) JROCKIT_HOME/bin/jrcmd jrockit_pid dump_flightrecording recording=1
   copy_to_file=path compress_copy=true
   (Windows) JROCKIT_HOME\bin\jrcmd.exe jrockit_pid dump_flightrecording
   recording=1 copy_to_file=path compress_copy=true
   ```

   See the "Running Diagnostic Commands" chapter in the *Oracle JRockit JDK Tools Guide* for more information about the `jrcmd` command-line tool.

2. To view the file, start the JRockit Mission Control Client from the following directories:

   ```
   (UNIX) JAVA_HOME/bin/bin/jrmc
   (Windows) JAVA_HOME\bin\jrmc.exe
   ```

3. Choose **File** > **Open File** to select the JFR file.

4. Locate the slowest requests or investigate a specific request:

To locate the slowest requests:	To investigate a specific request:
1. In the JRockitFlightRecorder.jfr page, click the **Events** icon. 2. Click the **Log** tab at the bottom of the page. 3. In the **Event Type** navigation pane on the left, locate **Dynamic Monitoring System** and then **HttpRequest**. 4. Click **HTTP request**; de-select all the other event types. 5. In the **Log** tab, in the **Event Log** section, click the **Duration** column to sort the duration in descending order. Each row corresponds to a HTTP Request and the duration column shows the response time for that request. 6. Click the row in the table to view the attributes of the requests. 7. In the **Event Attributes** sections, note the start time and the thread that serviced the request.	1. Find the Execution Context Identifier (ECID) of that request. If the request is related to an incident triggered by a `STUCK` thread, the incident `readme.txt` file will contain the ECID. Alternatively, you can search the Oracle WebLogic Server HTTP `access.log` for requests from specific users. See the "Managing Oracle Fusion Applications Log Files" chapter in the *Oracle Fusion Applications Administrator's Guide*. 2. In the JRockit Mission Control Client, in the JRockitFlightRecorder.jfr page, choose the **WebLogic** icon, and then If the **Weblogic** icon is not available, choose **Help** > **Install Plugins** to download the Oracle WebLogic Server plug-in. 3. Click the **ECIDs** tab at the bottom of the age. 4. In the **ECIDs** section, from **Filter Column** list, select **ECID**. 5. Enter the ECID in the search box and click **Enter**. 6. In the results table, highlight the row with the matching ECID and right-click to bring up the menu. 7. Choose **Operative Set** > **Clear**, and then **Operative Set** > **Add matching ECID** > *ECID* to add the ECID to the operative set. This enables users to view only events associated with the operative set. 8. Click the **Events** icon. 9. In the Event Type navigation pane on the left, locate **Dynamic Monitoring System** and then **HttpRequest**. 10. Click **HTTP request**; de-select all the other event types. ** 11. In the **Event Log** section, click **Show Only Operative Set**. Each row corresponds to the request with the matching ECID 12. Click the row in the table to view the attributes of the requests. 13. Note the start time and the thread that serviced the request

5. Once the start time and the thread that serviced the request are identified, in the Logs tab, drag the time selector at the top of the screen to include only the time window for the duration of the request.
6. In the **Event Log** section, perform the following search:
   a. Deselect **Show Only Operative Set**.
   b. Enter the thread name in the search box.

c. From the **Filter Column** list, select **Thread**.

d. Choose <Enter>.

7. In the **Event Type** navigation pane on the left, click the events of interest. Typically, these events are located under nodes **Dynamic Monitoring System**, **Java Application**, and **WebLogic** > **JDBC**.

   The selected events appear in the table in the **Event Log** section.

8. Click the **Start Time** column to sort y the time when these events occur, or click the **Duration** column to view the events that took longest

   The **JDBC Statement Execute** events corresponds to SQL execution. If there are slow SQL statements, the event details give the SQL text. These events do not have callstacks.

9. To see to callstack for slow SQL statements, view the **Socket Read** event that happens right after the **JDBC Statement Execute** event.

   This event corresponds to Oracle WebLogic Server waiting for the SQL results to return, and it has callstack in the event details.

10. Review the callstacks for long **Java Blocked** and **Java Wait** events to see if the cause can be identified. See the "Analyzing Flight Recorder Data in JRockit Mission Control" section in the *Oracle Fusion Middleware Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*.

11. If more details are needed to compare with what is captured in the default recording, and the user can reproduce the slowness, start an explicit recording; that is, one you start explicitly and let run for some predetermined length of time or until you manually stop it,

    The simplest way to control JRockit Flight Recorder is by using the Oracle JRockit Mission Control Client. JRockit Mission Control Client is a tools suite that you can use to monitor, manage, profile, and eliminate memory leaks in your Java application, without introducing the performance overhead normally associated with these types of tools.

    The JRockit Mission Control Client executable is located in `JROCKIT_HOME/bin`. If this directory is on your system path, you can start JRockit Mission Control by simply typing **jrmc** in a command (shell) prompt.

    Otherwise, you have to type the full path to the executable file, as shown here:

    ```
    JROCKIT_HOME\bin\jrmc.exe (Windows)
    JROCKIT_HOME/bin/jrmc (Linux)
    ```

    On Windows installations, you can start JRockit Mission Control from the **Start** menu.

2.3 Using My Oracle Support for Additional Troubleshooting Information

You can use My Oracle Support (formerly MetaLink) to help resolve Oracle Fusion Applications problems. My Oracle Support contains several useful troubleshooting resources, such as:

- Knowledge base articles
- Community forums and discussions
- Patches and upgrades
- Certification information

Note: You also can use My Oracle Support to log a service request.

You can access My Oracle Support at https://support.oracle.com.

3

Tuning the Database

This chapter discusses database tweaks that are specific to Oracle Fusion Applications.

This chapter contains the following sections:

3.1 Introduction

Oracle Fusion Applications set values on a number of v$session attributes to indicate how the connection is being used. When looking at a connection consuming high CPU on the database, or when trying to understand what connections are used for what processes, these attribute values may help provide the answer.

Table 3–1 v$session Attribute Values

Attribute in v$session	Value Being Set
Process	Data Source Name (such as ApplicationDB)
Program	WebLogic Server Domain plus Managed Server name, prefixed by DS (such as DS/FinancialDomain/AccountsReceivableServer_1)

Module	■ For ADF: ADF BC application module name
	■ For ESS:
	  ■ For Java job type, the class name (minus oracle.apps)
	  ■ For PL/SQL, the package and procedure name (such as mypkg.myproc)
	  ■ For other job types, the executable name should be passed
	■ For BI Publisher: the name of the report

Table 3–1 (Cont.) v$session Attribute Values

Attribute in v$session	Value Being Set

Action	▪ For ADF: jspx name
	▪ For ESS: jobDefinitionName
	▪ For BI Publisher, if request is submitted:
	  ▪ Using ESS scheduler: ESS Job Definition Name
	  ▪ Using BIP Scheduler Job: BIP Job Name submitted by the user
	  ▪ Using BIP online: static string "BIP:Online"
	  ▪ Using BIP Webservices: the name of the webservices

Client_Identifier	Application User Name

3.1.1 Monitoring the Oracle Fusion Applications Database

See the "Monitoring and Tuning Oracle Database for Oracle Fusion Applications" chapter in the *Oracle Fusion Applications Administrator's Guide*.

3.2 Tuning the Database

Table 3–2 provides the database initialization parameter guidelines for Oracle Fusion Applications. Within the Oracle Fusion Applications ecosystem, there exist four types of databases. They are Online Transaction Processing (OLTP) Starter or Production configuration, IDM, and DW. These parameter values are intended to provide a baseline. The database that is installed during Oracle Fusion Applications provisioning is configured with the suggested values. As your deployment and workloads characteristics change, these values may need to be adjusted.

Tuning the database involves adjusting the sizing parameters based on the available resource and load on the database. The `sga_target`, `pga_aggregate_target` and `processes` parameters from Table 3–2 are examples of such parameters that need to be tuned based on SGA and PGA advisories and looking into the number of open processes during peak load.

In addition, you may consider setting a minimum value for `SHARED_POOL_SIZE` and `DB_CACHE_SIZE` to minimize frequent resizing.

Table 3–2 Common init.ora Parameters

Parameter Name	Description, Default value and suggested start value
`audit_trail`	Enables or disables database auditing. Oracle Release 2 (11.2.0.2) default value: `DB` The suggested value for all Oracle Fusion Applications databases is `NONE`.
`_fix_control`	This parameter addresses the dynamic sampling of global temporary tables. Oracle Release 2 (11.2.0.2) default value: N/A The suggested value for an Oracle Fusion Applications Starter or Production database is `5483301:OFF, 6708183:ON`

Table 3–2 (Cont.) Common init.ora Parameters

Parameter Name	Description, Default value and suggested start value
plsql_code_type	PLSQL_CODE_TYPE specifies the compilation mode for PL/SQL library units. INTERPRETED: PL/SQL library units are compiled to PL/SQL bytecode format. Such modules are executed by the PL/SQL interpreter engine. NATIVE: PL/SQL library units are compiled to native (machine) code. Such modules are executed natively without incurring any interpreter impacts. Oracle Release 2 (11.2.0.2) default value is INTERPRETED. The suggested value for all Oracle Fusion Applications databases is NATIVE.
nls_sort	NLS_SORT specifies the collating sequence for ORDER BY queries. ■ If the value is set to BINARY, the collating sequence for ORDER BY queries is based on the numeric value of characters (a binary sort that requires fewer system resources). ■ If the value is a named linguistic sort, sorting is based on the order of the defined linguistic sort. Most (but not all) languages supported by the NLS_LANGUAGE parameter also support a linguistic sort with the same name. Oracle Release 2 (11.2.0.2) default value: Derived from NLS_LANGUAGE. The suggested value for all Oracle Fusion Applications databases is BINARY.
open_cursors	Specifies the maximum number of open cursors (handles to private SQL areas) a session can have at once. It is important to set the value of OPEN_CURSORS high enough to prevent your application from running out of open cursors. Oracle Release 2 (11.2.0.2) default value is 50. The suggested value for all Oracle Fusion Applications databases is 500.
session_cached_cursors	Specifies the number of session cursors to cache. Repeated parse calls of the same SQL statement cause the session cursor for that statement to be moved into the session cursor cache. Subsequent parse calls find the cursor in the cache and do not reopen the cursor. Oracle uses a least recently used algorithm to remove entries in the session cursor cache to make room for new entries when needed. This parameter also constrains the size of the PL/SQL cursor cache that PL/SQL uses to avoid having to re-parse as statements are re-executed by a user. Oracle Release 2 (11.2.0.2) default value is 50. The suggested value for all Oracle Fusion Applications databases is 500.
_b_tree_bitmap_plans	This enables use of bitmap access paths for b-tree indexes. Oracle Release 2 (11.2.0.2) default value: TRUE The suggested value for all Oracle Fusion Applications databases is FALSE.
processes	Sets the maximum number of operating system processes that can be connected to Oracle concurrently. The value of this parameter must account for Oracle background processes. SESSIONS parameter is deduced from this value. Oracle Release 2 (11.2.0.2) default value: 100 The suggested value for an Oracle Fusion Applications DW or IDM database (OID, OIM) is 2500. The suggested value for an Oracle Fusion Applications Starter or Production database is 5000.
sga_target	Setting this parameter to a nonzero value enables Automatic Shared Memory Management. Consider using automatic memory management, both to simplify configuration and to improve performance. Oracle Release 2 (11.2.0.2) default value: 0 The suggested value for Oracle Fusion Applications IDM databases (OID, OIM) is 4 GB. The suggested value for an Oracle Fusion Applications Starter database is 9 GB. The suggested value for an Oracle Fusion Applications DW database is 8 GB. The suggested value for an Oracle Fusion Applications Production database is 18 GB (based on reference hardware having 32 GB of physical memory).

Table 3–2 (Cont.) Common init.ora Parameters

Parameter Name	Description, Default value and suggested start value
pga_aggregate_target	Specifies the target aggregate Program Global Area (PGA) memory available to all server processes attached to the instance. Oracle Release 2 (11.2.0.2) default value is 0. The suggested value for Oracle Fusion Applications IDM databases (OID, OIM) is 2 GB. The suggested value for an Oracle Fusion Applications Starter database is 4 GB. The suggested value for an Oracle Fusion Applications DW database is 4 GB. The suggested value for an Oracle Fusion Applications Production database is 8 GB (based on reference hardware having 32 GB of physical memory).
star_transformation_enabled	Determines whether a cost-based query transformation will be applied to star queries. Oracle Release 2 (11.2.0.2) default value is FALSE. The suggested value for an Oracle Fusion Applications DW database is TRUE.
query_rewrite_integrity	Determines the degree to which Oracle must enforce query rewriting Oracle Release 2 (11.2.0.2) default value is ENFORCED. The suggested value for an Oracle Fusion Applications DW database is TRUSTED.
PARALLEL_MAX_SERVERS	Specifies the maximum number of parallel execution processes and parallel recovery processes for an instance. Oracle Release 2 (11.2.0.2) default value is Derived. The suggested value for an Oracle Fusion Applications DW database is 16. The suggested value for all Oracle Fusion Applications IDM databases is NUM_CPU_CORES (number of CPU cores on the database server). The suggested value for all Oracle Fusion Applications Production databases is NUM_CPU_CORES (number of CPU cores on the database server).
JOB_QUEUE_PROCESSES	Specifies the maximum number of job slaves per instance that can be created for the execution of DBMS_JOB jobs and Oracle Scheduler (DBMS_SCHEDULER) jobs. Oracle Release 2 (11.2.0.2) default value is 1000. The suggested value for an Oracle Fusion Applications DW database is 10. The suggested value for all Oracle Fusion Applications IDM databases is NUM_CPU_CORES (number of CPU cores on the database server). The suggested value for all Oracle Fusion Applications Production databases is NUM_CPU_CORES (number of CPU cores on the database server).
disk_asynch_io	Controls whether I/O to datafiles, control files, and logfiles is asynchronous. Oracle Release 2 (11.2.0.2) default value is TRUE. The suggested value for all Oracle Fusion Applications databases is TRUE.
filesystemio_options	Specifies I/O on file system files to be asynchronous I/O or direct I/O. Oracle Release 2 (11.2.0.2) default value: This parameter is platform-specific and has a default value that is best for a particular platform. The suggested value for all Oracle Fusion Applications databases is to use a value that is supported by the specific platform. *Do not* set it to NONE.
DB_SECUREFILE	Specifies whether or not to treat LOB files as SecureFiles. Oracle Release 2 (11.2.0.2) default value is PERMITTED. The suggested value for all Oracle Fusion Applications databases is ALWAYS.
_active_session_legacy_behavior	Enables active session legacy behavior. Oracle Release 2 (11.2.0.2) default value: FALSE The suggested value for all Oracle Fusion Applications Production databases is TRUE.
shared_servers	Specifies the number of server processes that you want to create when an instance is started. Oracle Release 2 (11.2.0.2) default value: 0 The suggested value for all Oracle Fusion Applications IDM database is 0. The suggested value for all Oracle Fusion Applications Production database is 0 or 1. A value of 1 is acceptable if the dispatchers parameter has the SERVICE clause defined as XDB.

3.2.1 How to Configure the Database for Performance

This section contains an overview for configuring a database for performance. Although it is expected that modifications will be made to the database on an ongoing basis to maintain or improve performance, significant benefits can be gained by proper initial configuration of the database.

In addition to tuning the database parameters, the database administrator should properly configure the REDO Logs, and the UNDO and TEMP tablespaces, to meet the demands of the expected or observed database workload. This is an empirical task. The recommendations in this section are intended to provide initial guidance in these areas.

UNDO

It is recommended that the default mode of automatic undo management be leveraged to maximize performance and efficiency. Oracle Enterprise Manager Fusion Applications Control Automatic Undo Management Advisor should be leveraged to set configuration details for undo tablespace and retention settings. This advisor also provides access to the Undo Advisor that assesses the effect and provides advice of a new undo retention setting. The suggested minimum size for the UNDO tablespace is 6 GB with `auto-extend` enabled.

TABLESPACES

The location of the datafiles should be optimized for I/O performance and growth. Also, Fusion Applications Control Segment Advisor should be leveraged to optimize the use of segment space and assure performance degradation does not occur. The advisor can provide historical growth trends of segments, which can be used to proactively plan for growth.

TEMP

Oracle recommends the use of locally-managed temporary tablespaces with UNIFORM extents and the default size of 1 MB. Some workflows in Oracle Fusion applications can generate a large amount of disk sorts that require high temporary space requirements. Therefore, the use of multiple temporary tablespaces and tablespace groups is recommended to meet these requirements and assure optimal performance. A suggested minimum size for the TEMP tablespace (or tablespace group) assigned to the Oracle Fusion Applications schema owners is 6 GB with `auto-extend` enabled.

REDO LOGS

Under demanding workloads, the size of the redo log files can influence performance. Generally, larger redo log files provide better performance. Undersized log files increase checkpoint activity and reduce performance. You can obtain sizing advice on the Redo Log Groups page of Fusion Applications Control. In addition, depending on your storage configuration and performance characteristics, you may need to redistribute redo logs to optimize I/O performance. The suggested minimum setting for Redo Logs is to have 3 log files of 2 GB each.

3.2.2 How to Configure Kernel Parameters

The parameters listed here are only a subset of the parameters needed to properly configure your database environment. Consult the Oracle Database Installation Guide for your specific operating system to determine the overall requirements for the kernel parameter settings. Installation guides are available in the Oracle Database Documentation Library.

If the current values of these parameters are larger than what is recommended, no change is necessary.

- SHMMAX - Set to the larger of either the largest System Global Area (SGA) on the system or half of the physical memory available.
- SEMMNS - Set to the larger of either 32000 or twice the sum of the PROCESSES initialization parameter for each Oracle database.
- SHMALL - Set to the sum of all SGAs divided by the page size (`getconf PAGE_SIZE`).
- SHMMNI - Recommended value is 4096.
- vm.nr_hugepages - Should be set for SGA larger than 8 GB. The value should be set to the sum of all SGAs.

3.2.3 How to Configure the Database Listener

The database listener is responsible for accepting and routing connection requests to the database. The parameters described in this section can change the behavior of the listener and, therefore, need to be examined and changed accordingly to address the needs of your particular database connection load.

SQLNET.EXPIRE_TIME Parameter

For Fusion application environments that leverage a firewall between the middle and database tiers, it is recommended to set the database listener setting SQLNET.EXPIRE_TIME. This parameter is defined in the sqlnet.ora file that is located in the TNS_ADMIN location. This parameter is used to set a time interval, in minutes, to determine how often to probe connections to verify their status.

Typically, a firewall is configured to terminate connections after it has been idle for a specified amount of time. Setting SQLNET.EXPIRE_TIME to an interval smaller than the firewall's will generate enough traffic within an appropriate interval to prevent the firewall from determining the connections are in an idle state and terminating these connections. A setting of SQLNET.EXPIRE_TIME = 10 is commonly used.

INBOUND_CONNECT_TIMEOUT Parameter

On occasion, due to network latency, connections can exceed the default timeout of 60 seconds. Typical symptoms of this include observing "TNS-12537: TNS:connection closed" errors in the database alert logs. One of the things that can resolve these issues is to increase the timeout parameters. A suggested value of INBOUND_CONNECT_TIMEOUT_*listenerName* = 120 (listener.ora) and SQLNET.INBOUND_CONNECT_TIMEOUT=130 (sqlnet.ora) is recommended.

3.2.4 How to Tune the Real Application Cluster (RAC)

Oracle Real Application Clusters (RAC) is a cluster database with a shared cache architecture that overcomes the limitations of a traditional shared-nothing approach to provide highly scalable and available database solutions for all business applications.

RAC supports the transparent deployment of a single database across a cluster of servers, providing fault tolerance from hardware failures or planned outages in terms of availability, scalability, and low-cost computing.

RAC provides:

- High Availability

RAC provides the highest availability for applications by removing the single server as a single point of failure.

- Flexible Scalability

  When more processing power and resources are needed, adding a server to the database cluster without taking users offline will gain horizontal scalability.

- Automatic Workload Management

  Application workloads can be individually managed and controlled using managed services. Users connecting to a service are load balanced across the server pool.

Initial Setting

The network between the nodes of a RAC cluster must be private.

Supported links: Giga-bit Ethernet and InfiniBand

Supported transport protocols: UDP or RDS

Use multiple or dual-ported Network Interface Cards (NICs) for redundancy and increase bandwidth with NIC bonding.

Set these parameters with respect to the ORACLE/OS release for Interconnect performance.

```
net.core.rmem_default = 262144
net.core.rmem_max = 262144
net.core.wmem_default = 262144
net.core.wmem_max = 262144
```

Troubleshooting

Use this information to identify if proper interconnect is used and picked up by clusterware.

- To identify the interconnect:

  ```
  show parameter cluster_interconnects
  or
  select name,ip_address,is_public from v$cluster_interconnects
  ```

  The Alert log will have an entry similar to this:

  ```
  Cluster communication is configured to use the following interface(s) for this
  instance
   123.45.67.89
  cluster interconnect IPC version:Oracle UDP/IP (generic)
  ```

- Identify network and contention issues.

  Check for "gc cr lost blocks" wait event in Automatic Workload Repository (AWR)/sysstats.

  If found, check for these errors on the NIC:

  – Dropped packets/fragments

  – Buffer overflows

  – Packet reassembly failures or timeouts

  – TX/RX errors

  Use these commands to find any errors:

```
netstat -s
Ifconfig -a
ORADEBUG
```

- Identify Interconnect performance from AWR.
  - Under **Global Cache and Enqueue Services - Workload Characteristics**

    Avg global cache cr block receive time (ms): should be <=15 ms
  - **Global Cache and Enqueue Services - Messaging Statistics**

    Avg message sent queue time on ksxp (ms): should be <1 ms
  - Under **Interconnect Ping Latency Stats**

    Avg Latency 8K msg should be close to Avg Latency 500B msg.
- These wait events from AWR and sysstat can indicate contention related to RAC.

  ```
  GC current block busy
  GV cr block busy
  GC current buffer busy
  GC buffer busy acquire/release
  ```

  See the *Oracle Real Application Clusters Administration and Deployment Guide* for a complete list.

  These wait events in the AWR indicate that there might be a Hot Block that is causing these wait events. From the AWR Segment Statistics, you can find the objects.

  ```
  Enq:TX Index Contention
  Gc buffer busy
  Gc current block busy
  Gc current split
  ```

  This issue will be noticed if multiple sessions are inserting into a single object or are using a sequence, and the indexed column is sequentially increasing. To address the specific issues:
  - Identify the indexes and Global Hash Partition them.
  - Increase the Sequence Cache if ordering is not a problem.

3.2.5 How to Optimize SQL Statements

Oracle Fusion Applications use Cost Based Optimization (CBO) to choose the most efficient execution plan for SQL statements. Using this approach, the optimizer determines the most optimal execution plan by costing available access paths and factoring information based on statistics for the schema objects accessed by the SQL statement.

3.2.5.1 Collecting Optimizer Statistics

For the query optimizer to produce an optimal execution plan, the statistics in the data dictionary should accurately reflect the volume and data distribution of the tables and indexes. To this end, database statistics should be refreshed periodically. However, that does not necessarily imply that you should gather statistics frequently. Systems that are close to going live typically experience inserts of a large amount of data, as data from legacy systems is migrated. In that scenario, the statistics would probably need to be refreshed quite frequently (for instance, after each major load), as large loads could change the data distribution significantly. Once the system reaches steady state, the

frequency of statistics collection at the schema and database level should be reduced to approximately once a month. However, statistics on some volatile tables can be gathered as frequently as required.

Oracle Fusion Applications has an automated way of gathering the statistics. It uses `DBMS_STATS` with the `AutoTask` feature. For more information about `AutoTask`, see the "Enabling and Disabling Automatic Optimizer Statistics Collection" section in the *Oracle Database Performance Tuning Guide*.

Gathering Statistics for Cost Based Optimizer

Oracle Fusion Applications use automatic optimizer statistics collection. That is, the database automatically gathers the optimizer statistics. Automatic statistics eliminates any manual intervention, thereby significantly reducing the onus on a system administrator. For a DW database, optimizer statistics are collected as part of ETL and BI apps, so it is recommended not to use automatic statistics gathering.

Table Statistics and Number of Distinct Values (NDV)

For versions prior to Oracle 12g, a database performance problem can occur by missing a non-popular value in the sample created from the table, leading to a frequency histogram where the number of buckets is less than the number of values.

To work around this problem, set this entry in `init.ora`:

```
_fix_control='5483301:off','6708183:ON';
```

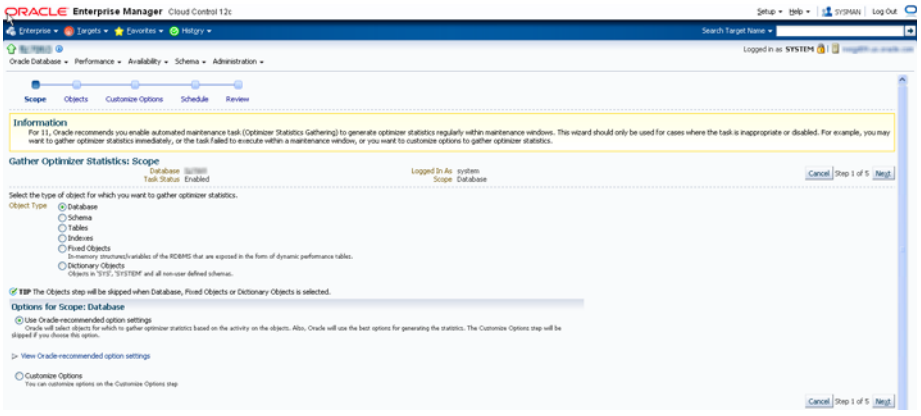
Manual Statistics Gathering

Automatic optimizer statistics collection is sufficient for most of the database objects, but in a database that is close to going live or for tables that are modified significantly, manual statistic gathering is needed.

In these cases, use the `DBMS_STATS.GATHER_TABLE_STATS` procedure. Do not explicitly set any parameters apart from owner and table_name.

All the parameters, such as `estimate_percent`, `parallel degree` and `method_opt`, are globally seeded for Oracle Fusion database tables. Therefore, there is no need to use them when manually gathering the statistics. See Figure 3–1.

Figure 3–1 Using the Fusion Applications Control to Gather the Statistics



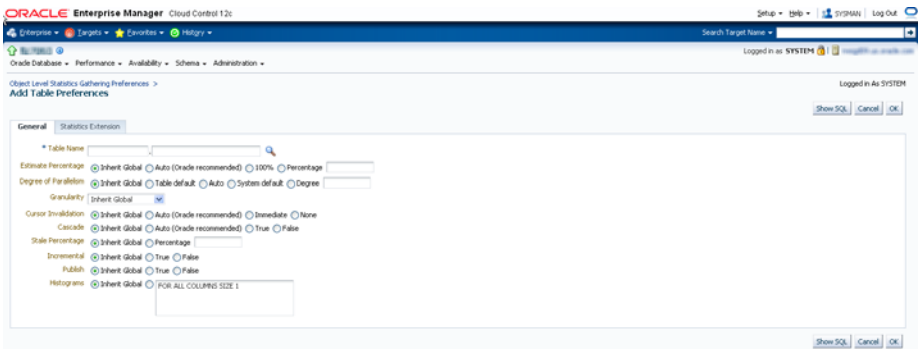
Histograms

All the columns that need a histogram are seeded using `DBMS_STATS.SET_TABLE_PREFS` and the automatic optimizer statistic collection will create histograms for all the

pre-identified columns apart from the columns identified by DBMS_STATS under size auto.

You can use DBMS_STATS.SET_TABLE_PREFS to manually seed histograms. See Figure 3–2

Figure 3–2 Using Oracle Enterprise Manager Cloud Control



MDS DB: Collecting database statistics for optimizing the MDS database repository performance

Ensure auto-stats collection is enabled.

In most cases, the first 32 characters of PATH_FULLNAME in the MDS_PATHS table are the same. You can prevent the database putting them in the same section of the histogram by doing the following:

- Drop the histogram for PATH_FULLNAME column by executing the following as system.

```
execute dbms_stats.delete_column_stats(ownname=>'mdsSchemaOwner',
tabname=>'MDS_PATHS', colname=>'PATH_FULLNAME', col_stat_type=> 'HISTOGRAM');
```

- Set table preferences to exclude collecting histogram for the PATH_FULLNAME column.

```
execute dbms_stats.set_table_prefs(mdsSchemaOwner, 'MDS_PATHS', 'METHOD_OPT',
'FOR COLUMNS SIZE 1 PATH_FULLNAME');
```

3.2.5.2 Pinning Packages and Cursors

Pinning the objects in the shared pool reduces the possibility of ora-4031 error messages, and increases the performance of the OLTP applications.

Objects are cached in a library cache that uses a Least Recently Used (LRU) algorithm to flush the objects. The problem worsens if these large library cache objects are executed only infrequently. That is, if they are loaded into the library cache whenever required if they have aged out of the cache. This causes most of the problems leading to ora-4031 errors and poor performance for UI-specific flows.

Consider a large package, or any object, that has to be loaded into the shared pool. Large PL/SQL objects present particular challenges. The database has to search for free space for the object. If it cannot get enough contiguous space, it will free many small objects to satisfy the request. If several large objects need to be loaded, the database has to throw out many small objects in the shared pool. Finding candidate objects and freeing memory is very costly. These tasks will affect CPU resources.

The same situation applies to SQL statements that are executed occasionally but are very important from a response time perspective. For example, a statement that is part

of a CEO's dashboard is executed once every 3 to 4 hours, but it takes a long time for parsing.

A common question asked is how to keep the LRU algorithm from forcing objects out of the shared pool. This is mainly useful in cases where a SQL statement that is very expensive to parse may just be executed once every 12 hours, such as a dashboard SQL statement. But due to the LRU algorithm, the results are flushed before the next execution. In these cases, you may prefer to pin the cursor so that subsequent parse times are reduced.

The `DBMS_SHARED_POOL` package provides procedures to facilitate this. With this, you have an easy way to ensure that the specified cursors always remain in the Most Recently Used (MRU) end of the cache. This prevents the cursor from being paged out and then re-parsed upon re-load. The `DBMS_SHARED_POOL.KEEP` procedure is used to pin the cursor and `DBMS_SHARED_POOL.UNKEEP` is used to unpin the cursor.

3.2.6 How to Configure the Database Resource Manager

To better ensure system stability during periods of high system load and prevent runaway queries, Oracle Database Resource Manager can be enabled on the Oracle Fusion Applications database instance.

For more details about the feature, including an explanation on when enabling this feature is desired, see the "Managing Resources with Oracle Database Resource Manager" chapter in the *Oracle Database Administrator's Guide*.

Use the resource manager to:

- Assign connections to different resource consumer groups.
- Create directives to manage resource allocations for connections in different resource consumer groups.

3.2.6.1 Assigning Connections to Resource Consumer Groups

There are multiple ways a connection can be assigned to a resource consumer group. See the *Oracle Database Administrator's Guide* for complete details.

See Table 3–1 for connection attributes settings for Oracle Fusion Applications.

3.2.6.2 Using the Oracle Database Resource Manager

This section shows how to use the Oracle Database Resource Manager for Oracle Fusion Applications. Connection attribute mappings, and explicit assignment using a database login trigger are used to assign connections to different resource consumer groups. The plan is enabled by default for a newly-provisioned instance. If you are upgrading from earlier releases, you can run the scripts listed in "Enabling the Resource Plan" to create the same resource plan that is enabled by default in a freshly-provisioned database. The resource plan is called `FUSIONAPPS_PLAN`.

The plan has two resource consumer groups:

- **FUSIONAPPS_ONLINE_GROUP** - Connections used for servicing ADF UI pages are assigned to this resource group via a login trigger.
- **FUSIONAPPS_BATCH_GROUP** - All other connections used by Fusion Applications are assigned to this group.

Resource Directives

These resource directives are defined:

- Connections in FUSIONAPPS_ONLINE_GROUP get priority for 45% of the CPU. Parallel query is disabled, and any queries consuming more than 120s of CPU or resulting in more than 10GB of I/O will be canceled.
- Connections in FUSIONAPPS_BATCH_GROUP get priority for 35% of the CPU. There are no other resource restrictions.
- Connections used for sysdba activities get priority for 15% of the CPU. There are no other resource restrictions.
- All non-Oracle Fusion applications connections get priority for 5% of the CPU.

  Note that if a particular resource group does not consume all its allocated CPU, the unused CPU can be used by other resource consumer groups.

Mapping Connections

To map connections to resource consumer groups, this approach is used:

- All schemas containing the string FUSION are mapped to FUSIONAPPS_BATCH_GROUP.
- A login trigger is used to check connections to schema FUSION_RUNTIME. If the connection is coming from an ADF server, assign the connection to FUSIONAPPS_ONLINE_GROUP.

Enabling the Resource Plan

To enable this resource plan, follow these steps:

- Run this script using sqlplus as sysdba to create the resource plan, the resource consumer groups and the resource directives.

```
begin
  DBMS_RESOURCE_MANAGER.CLEAR_PENDING_AREA();
  DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();

  begin
    DBMS_RESOURCE_MANAGER.DELETE_PLAN('FUSIONAPPS_PLAN');
  exception
    when others then
      null;
  end;

  begin
    dbms_resource_manager.delete_consumer_group(CONSUMER_GROUP => 'FUSIONAPPS_
ONLINE_GROUP');
  exception
    when others then
      null;
  end;

  begin
    dbms_resource_manager.delete_consumer_group(CONSUMER_GROUP => 'FUSIONAPPS_
BATCH_GROUP');
  exception
    when others then
      null;
  end;

  dbms_resource_manager.create_consumer_group(CONSUMER_GROUP => 'FUSIONAPPS_
ONLINE_GROUP', COMMENT => 'Consumer Group for online users');
  dbms_resource_manager.create_consumer_group(CONSUMER_GROUP =>'FUSIONAPPS_
```

```
BATCH_GROUP', COMMENT => 'Consumer Group for batch');

  dbms_resource_manager.create_plan(PLAN => 'FUSIONAPPS_PLAN',
  COMMENT => 'Fusion Applications Resource Plan');

  dbms_resource_manager.create_plan_directive(
    plan => 'FUSIONAPPS_PLAN',
    group_or_subplan => 'FUSIONAPPS_ONLINE_GROUP',
    comment => 'Online users at level 1',
    mgmt_p1 => 45,
    parallel_degree_limit_p1 => 0,
    switch_time => 120,
    switch_io_megabytes => 10000,
    switch_group => 'CANCEL_SQL');

  dbms_resource_manager.create_plan_directive(
    plan => 'FUSIONAPPS_PLAN',
    group_or_subplan => 'FUSIONAPPS_BATCH_GROUP',
    comment => 'Batch users at level 1',
    mgmt_p1 => 35);

  dbms_resource_manager.create_plan_directive(
    plan => 'FUSIONAPPS_PLAN',
    group_or_subplan => 'SYS_GROUP',
    comment => 'System administrator group at level 1',
    mgmt_p1 => 15);

  dbms_resource_manager.create_plan_directive(
    plan => 'FUSIONAPPS_PLAN',
    group_or_subplan => 'OTHER_GROUPS',
    comment => 'Other users at level 1',
    mgmt_p1 => 5);

  for rec in (select username from dba_users where username like '%FUSION%')
loop
    if (rec.username <> 'FUSION_READ_ONLY') then
      dbms_resource_manager.set_consumer_group_mapping('ORACLE_
USER',rec.username,'FUSIONAPPS_BATCH_GROUP');
    end if;
  end loop;

  begin
    dbms_resource_manager.set_consumer_group_mapping('ORACLE_
USER','SEARCHSYS','FUSIONAPPS_BATCH_GROUP');
  exception
    when others then
      null;
  end;

  DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
  DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
  DBMS_RESOURCE_MANAGER.CLEAR_PENDING_AREA();
  dbms_resource_manager_privs.grant_switch_consumer_group('PUBLIC','FUSIONAPPS_
ONLINE_GROUP',FALSE);
  dbms_resource_manager_privs.grant_switch_consumer_group('PUBLIC','FUSIONAPPS_
BATCH_GROUP',FALSE);
end;
/
```

- As `sysdba`, issue the following command to enable the resource plan:

```
(if using spfile)
 ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = FUSIONAPPS_PLAN SCOPE = BOTH;

(if not using spfile)
 ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = FUSIONAPPS_PLAN;
```

- Run this script using sqlplus as `sysdba` to create the login trigger:

```
CREATE OR REPLACE TRIGGER fusion_resource_trigger
   AFTER logon ON fusion_runtime.schema
declare
login_sid pls_integer;
login_program varchar2(40);
old_plan21    varchar2(44);
  begin

select 'Y' into login_program from dual where exists
(select program from v$session where audsid=userenv('sessionid') and
  process like 'ApplicationDB%' and
  ( program is not null and
    program not like 'JDBC Thin Client' and
    program not like '%ess_server%' and
    program not like '%soa_server%' and
    program not like '%SearchServer%' and
    program not like '%search_server%' and
    program not like '%odi_server%' and
    program not like '%bi_server%'
    ) );

if login_program = 'Y'
then
dbms_session.switch_current_consumer_group('FUSIONAPPS_ONLINE_GROUP',old_
plan21,false);
else
null;
end if;
EXCEPTION
when NO_DATA_FOUND
then
null;
  end;
  /
```

- If applicable, change the `init.ora` parameter to enable the resource plan:

```
RESOURCE_MANAGER_PLAN = FUSIONAPPS_PLAN
```

- Bounce all middle tiers.

3.2.6.3 Monitoring the Resource Manager

For details, see the *Oracle Database Administrator's Guide*. In particular, **v$rsrc_consumer_groups** provides information on CPU wait time and the number of SQL statements canceled.

"Scripts and Tips for Monitoring CPU Resource Manager", which is note ID 1338988.1 in My Oracle Support, also has information about monitoring CPU usage for different resource consumer groups. If a resource consumer group is spending significant time waiting for CPU, review the resource directives and see if the waits are expected and if any adjustments are needed.

3.2.7 How to Tune the PS_TXN Table in the FUSION Schema

Oracle Fusion Applications use the PS_TXN table to store the intermediate processing state. When there are many concurrent users, this table receives a high number of inserts and could suffer from concurrency issues. To detect this contention issue, check the wait event "enq: HW contention" in the AWR report of the database.

Follow the steps outlined in note ID 1444959.1 in My Oracle Support to alleviate the contention.

3.2.8 How to Optimize Maintenance Windows

By default, Oracle enables various AUTOTASK clients and preconfigures maintenance windows that can be optimized to reduce unnecessary overhead.

3.2.8.1 Disabling AUTOTASK

Starting with Oracle Release 11.2.0.2, you should evaluate these AUTOTASKs. We recommended that they be disabled if the respective features are not being used.

- AUTO SPACE ADVISOR
- SQL TUNING ADVISOR

The syntax for disabling these advisors is:

- execute `dbms_auto_task_admin.disable(client_name => 'auto space advisor',operation => NULL, window_name => NULL);`
- execute `dbms_auto_task_admin.disable(client_name => 'sql tuning advisor',operation => NULL, window_name => NULL);`

3.2.8.2 Adjusting Default Maintenance Windows

Starting with Oracle Release 11.2.0.2, various weekday and weekend windows are preconfigured with durations of 4 and 20 hours respectively. You should adjust these windows to run during time frames with minimal application workloads. In addition, you should reduce the weekend windows from the default configuration to a duration lower than the "byhour" value defined in the maintenance window's "repeat interval" settings. These values can be reviewed by querying the DBA_SCHEDULER_ WINDOWS view. A suggested maintenance window configuration is to enable both the WEEKNIGHT_WINDOW and WEEKEND_WINDOW with a duration of 4 hours each.

Example syntax for how to reduce the duration for maintenance windows is:

- execute `dbms_scheduler.set_attribute(name => WEEKEND_WINDOW', attribute => 'duration', value => '0 04:0:0.0');`
- execute `dbms_scheduler.set_attribute(name => WEEKNIGHT_WINDOW', attribute => 'duration', value => '0 04:0:0.0');`

3.2.9 How to Reduce Connection Creation Originating from SOA Server

With the default settings, you may see connection creations originating from the different SOA servers in each domain, even if the environment is idle.

These connections are opened, used and then closed. Under high load, these connection requests could cause performance issues.

To reduce the number of connection creations:

- Open the WebLogic console for each domain.

- Uncheck the "Remove Infected Connections Enabled" checkbox under the "Connection Pool" tab for the following data sources:
  - EDNDataSource
  - EDNLocalTxDataSource
  - EDNLocalTxSource
  - EDNSource
  - SOADataSource
  - SOALocalTxDataSource
  - mds-soa

3.2.10 How to Review Automatic Database Diagnostic Monitor Reports Regularly

ADDM automatically detects and reports performance problems with the database. The results are displayed as ADDM findings on the Database Home page in Oracle Enterprise Manager Cloud Control (Cloud Control). Reviewing the ADDM findings enables you to quickly identify the performance problems that require your attention. You always should review the results of the ADDM analysis.

Each ADDM finding provides a list of recommendations for reducing the effect of the performance problem. You should review ADDM findings and implement the recommendations every day as part of regular database maintenance. Even when the database is operating at an optimal performance level, you should continue to use ADDM to monitor database performance on an ongoing basis.

For instance, ADDM will report if database cache sizes need to be increased.

4

Diagnosing the Oracle Fusion Applications Middle Tier Performance

This chapter describes how to diagnose Java applications in the middle tier using Oracle Enterprise Manager Cloud Control (Cloud Control).

This chapter includes the following topics:

4.1 Introduction to Java Diagnostics in the Middle Tier

Occasionally, developers and IT administrations spend a lot of time diagnosing the root cause of hard to reproduce problems. Many times, the problems occurring in production environments. This can cause severe impact on the business.

Oracle Enterprise Manager Cloud Control (Cloud Control) enables you to diagnose performance problems in Java applications in the production environment. By eliminating the need to reproduce problems, it reduces the time required to resolve these problems. This improves application availability and performance. Using Java Virtual Machine (JVM) diagnostics, you can identify the root cause of performance problems in the production environment without having to reproduce them in the test or development environment. It does not require complex instrumentation or restarting of the application to get in-depth application details. Application administrators will be able to identify Java problems or Database issues that are causing application downtime without any detailed application knowledge.

Java Virtual Machine (JVM) is a virtual machine that can execute Java bytecode.

To configure your environment for JVM Diagnostics, see the "Using JVM Diagnostics" chapter in the *Oracle Enterprise Manager Getting Started with Fusion Middleware Management*.

4.2 Diagnosing the Oracle Fusion Applications Middle Tier

This section describes how to diagnose Java applications in the middle tier using Cloud Control.

4.2.1 Viewing JVM Pool Summary Information on a WebLogic Domain

To get started with monitoring Java Virtual Machine Pool on a WebLogic Domain using Cloud Control:

1. From the **Targets** menu, choose **Targets** > **Middleware**.

   The Middleware target home page displays.

2. Search for Oracle WebLogic Domains:

   a. Select an area under the **Area** drop-down list. For example, Oracle WebLogic Domains, Clusters, and Servers.

   b. Enter appropriate information in the fields **Comment**, **Contact**, **Cost Center**, **Customer Support Identifier**, **Department**, **Deployment Type**, **Lifecycle Status**, **Line of Business**, **Location**, **Status**, and **Target Version**.

   c. Click **Search**.

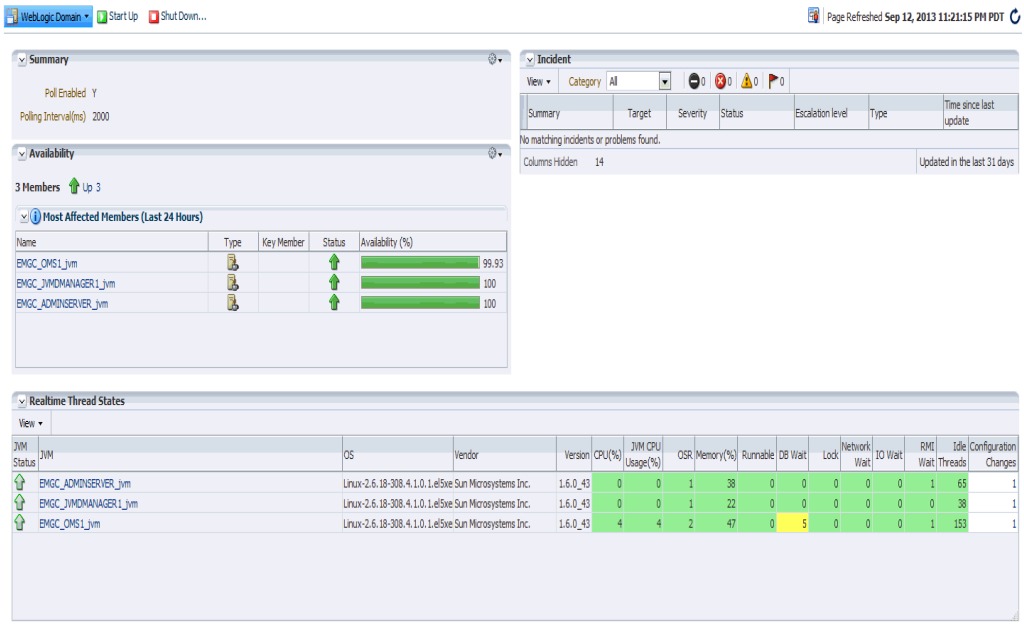
3. Select a domain which has the JVM Diagnostics Agent deployed on it:

   a. From the Target Navigation area, select Java Virtual Machine Pools and expand it.

   Note: All the WebLogic Domains that have a JVM Diagnostics Agent deployed on it appear under this folder.

4. From the **WebLogic Domain**, choose **Diagnostics** > **Java Virtual Machine Pool Home**.

   The JVM Pool Home page displays.



5. Take note of the following:

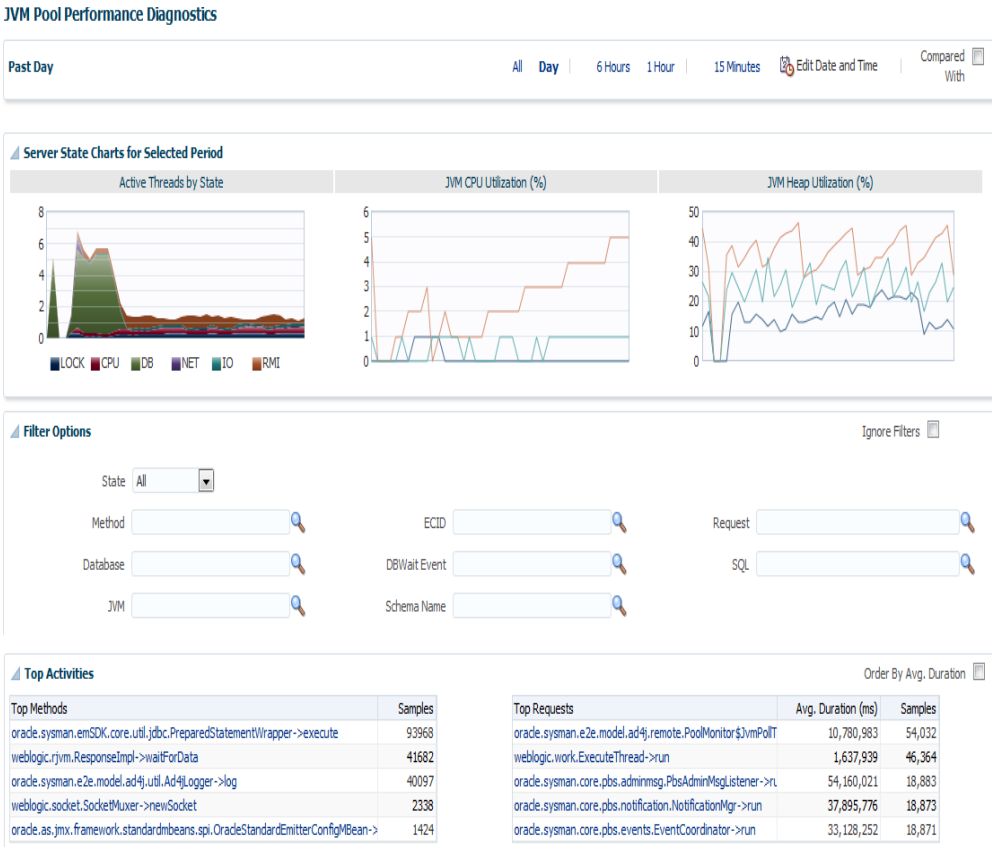
   - In the **Summary** section, `Polling Enabled` indicates if polling of that domain has been enabled. `Polling Interval` denotes the time interval (in milli seconds) at which the domain is polled.

- In the **Availability** section, the JVMs in that pool are listed along with their availability status in the last 24 hours.
- In the **Realtime Thread States** section, view the overall state of the threads including locks.
- In the **Incident** section, the servers in the domain whose target status is Down are listed. The incidents will be automatically cleared when the underlying issues are resolved.

a. From the **WebLogic Domain** menu, choose **Diagnostics** > **JVM Performance Diagnostics**.

The JVM Pool Performance Diagnostics page displays. This page provides information for the pool of JVMs in the domain for the selected time period. You can also modify the time period using Edit Date and Time option at the top of the page.

The following figure shows the JVM Pool Performance Diagnostics page for the domain:



b. View the various charts:

Server State Charts For Selected Period section

This section displays the Active Threads, CPU Utilization, and Heap Utilization of IO, CPU, lock, and network resources during the selected time. The number of Java threads (daemon and non-daemon) that are currently running in the virtual machine for this Oracle WebLogic Domain:

Active Threads by State: This chart displays the number of Java threads that are currently running in the domain or server. It is color-coded by thread state.

JVM CPU Utilization (%): This chart shows the CPU utilization across the JVMs in the pool.

JVM Heap Utilization (%): This chart shows the heap utilization across the JVMs in the pool.

Filter Options section

This section displays the options by which you can filter the JVMs in the pool:

- State
- Method
- Database
- JVM
- ECID
- DBWait Event
- Schema Name
- Request
- SQL.

Top Activities section

This section displays the top activities in the pool:

Top Methods: This chart shows the most expensive Java methods in the selected time period.

Top Requests: This chart shows the top page requests in the selected time period.

Top DBWait Events: This chart shows the cross-tier correlation with the database.

Top SQLs: This chart displays the list of SQL calls ordered by their cost (the number of samples) The SQL calls can also be ordered by average duration by selecting this option in the top right corner of the section.

Top Databases: This chart shows the top impacted databases by the JVMs in this pool.

JVMs In Pool section	This section displays the Active Threads, CPU Utilization, and JVM Heap Utilization of IO, CPU, lock, and network resources during the selected time. **Active Threads:** This chart displays the number of active Java threads that are currently running in the domain. It is color-coded by thread state. **CPU Utilization (%):** This chart shows the CPU utilization across the JVMs in the pool. **JVM Heap Utilization (%):** This chart shows the heap utilization across the JVMs in the pool. **Garbage Collections (Invocations/min):** This chart shows the number of times the JVM garbage collector was invoked in the time period. It includes both major and minor garbage collections.

Note: The **Compared with** feature enables you to compare the diagnostics across two specified periods of time.

4.2.2 Viewing JVM Summary Information on a WebLogic Managed Server

In a production environment, a Managed Server hosts applications and the resources needed by those applications. A domain, which is a logically related group of Oracle WebLogic Server resources, can have any number of managed Servers. An Administration Server manages these servers.

To get started with monitoring Java Virtual Machine (JVM) on a WebLogic Managed Server within a domain using Cloud Control:

1. From the **Targets** menu, choose **Targets** > **Middleware**.

   The Middleware target home page displays.

2. Search for the Oracle WebLogic Domains:

   a. Select an area under the **Area** drop-down list. For example, Oracle WebLogic Domains, Clusters, and Servers.

   b. Enter appropriate information in the fields **Comment**, **Contact**, **Cost Center**, **Customer Support Identifier**, **Department**, **Deployment Type**, **Lifecycle Status**, **Line of Business**, **Location**, **Status**, and **Target Version**.

   c. Click **Search**.

3. Select a domain which has the JVM Diagnostics Agent deployed on it:

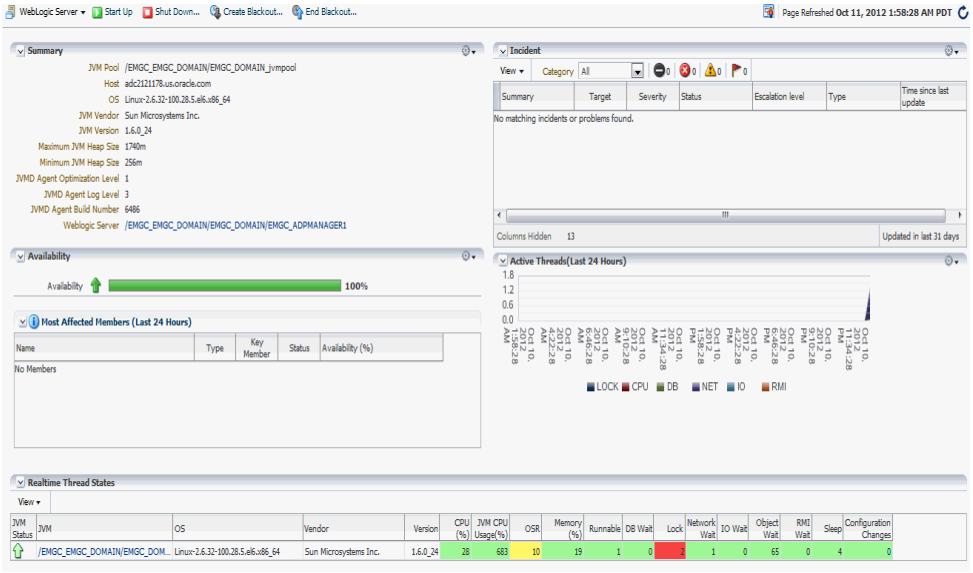
   a. From the **Target Navigation** area, select **Java Virtual Machine Pools** and expand it.

   Note: All the WebLogic Domains and Servers that have JVM Diagnostics Agents deployed on them appear under this folder.

4. Expand the **WebLogic Domain** and select a Managed Server.

5. From the **WebLogic Server menu**, choose **Diagnostics** > **Java Virtual Machine Home**.

The JVM Home page displays.



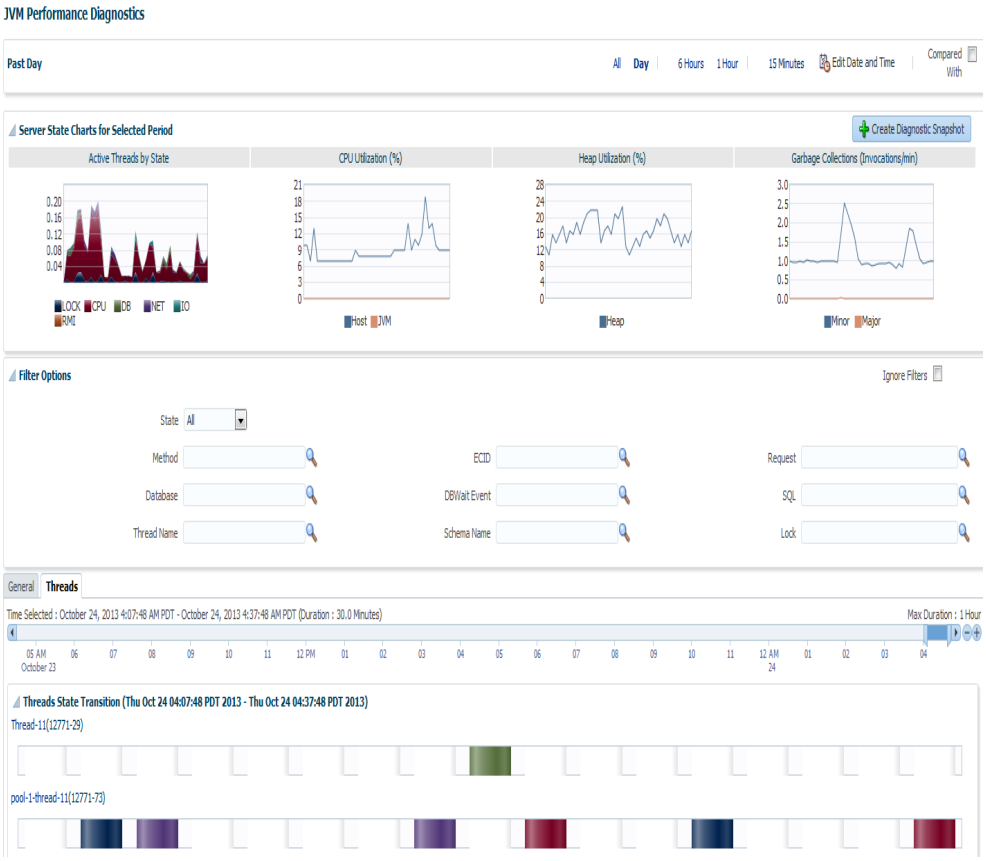
6. Take note of the following:

   - In the **Summary** section, the details of the JVM are displayed. The details contain the following information:
     - JVM Pool
     - Host machine
     - Operating System (OS) running on the host machine
     - JVM Vendor
     - JVM Version
     - Maximum JVM Heap Size
     - Minimum JVM Heap Size
     - JVMD Agent Optimization Level
     - JVMD Agent Log Level
     - JVMD Agent Build Number
     - WebLogic Server
     - List of composite applications to which the JVM belongs to (if applicable).
   - In the **Availability** section, the availability status in the last 24 hours of the JVM is listed.
   - In the **Realtime Thread States** section, view the overall state of the threads including locks.
   - In the **Incident** section, the JVM corresponding to the server whose target status is Down is listed. This incident will be automatically cleared when the underlying issue is resolved.
   - In the **Active Threads (Last 24 Hours)** section, view a graph analyzing the active threads in the JVM in the last 24 hours.

a. From the **WebLogic Server** menu, choose **Diagnostics** > **JVM Performance Diagnostics**.

The JVM Performance Diagnostics page displays. This page provides information for a single JVM for the selected time period. You can also modify the time period using Edit Date and Time option at the top of the page.

The following figure shows the JVM Performance Diagnostics page for a single JVM:



The **Thread Transition Graph** is displayed as ADF Bar Graph. By default, the data is shown for 30 minutes and the maximum time window is one hour.

b. View the various charts:

Server State Charts For Selected Period section

This section displays the Active Threads, CPU Utilization, and Heap Utilization of IO, CPU, lock, and network resources during the selected time. The number of Java threads (daemon and non-daemon) that are currently running in the virtual machine for this Oracle WebLogic Server:

Active Threads by State: This chart displays the number of Java threads that are currently running in the server. It is color-coded by thread state.

CPU Utilization (%): This chart shows the CPU utilization for the JVM.

Heap Utilization (%): This chart shows the heap utilization for the JVM.

Garbage Collections (Invocations/min): This chart shows the number of times the JVM garbage collector was invoked in the time period. It includes both major and minor garbage collections.

Filter Options section

This section displays the options by which you can filter the data:

- State
- Method
- Database
- JVM
- ECID
- DBWait Event
- Schema Name
- Request
- SQL.

General tab

This tab displays data from the JVM:

Active Threads by State: This chart displays the number of Java threads that are currently running in server. It is color-coded by thread state.

Top Requests: This chart shows the top page requests in the selected time period.

Top Methods: This chart shows the top methods in the selected time period.

Top SQLs: This chart shows the top SQLs in the selected time period.

Top DBWait Events: This chart shows the top DB Wait Events in the selected time period.

Top Databases: This chart shows the top impacted databases by the JVMs in this pool.

Threads tab

This tab displays data from the JVM.

Threads State Transition: This chart shows how the threads have transitioned from one state to the other in the selected period. You can change the time interval and move it to a different time period by using the quick time selection control at the top of the page. Each colored bar in the chart is a bucket. Depending on the time window selection, each bucket may contain one or more thread snapshot samples. You can hover over a bucket to see the summary of the samples in that bucket. The color change within a bucket or from one to next indicates thread state transition, for example from Runnable to Not Active or Runnable to DB Wait. When you click on a bucket, the Sample Analyzer pops up for detailed analysis of the thread snapshot. On Sample Analyzer popup, you can click **Next** or **Previous** to navigate from one sample to the next along the time line.

Metric By Active States: This chart shows how long each of the threads have been in the various states.

Note: The **Compared with** feature enables you to compare the diagnostics across two specified periods of time.

4.2.3 Viewing JVM Pool Summary Information on a JVM for a WebLogic Domain

To get started with monitoring a Java Virtual Machine (JVM) in a Pool using Cloud Control:

1. From the **Targets** menu, choose **Targets** > **Middleware**.

   The Middleware target home page displays.

2. Search for the Oracle WebLogic Domains:

   a. Select an area under the **Area** drop-down list. For example, Oracle WebLogic Domains, Clusters, and Servers.

   b. Enter appropriate information in the fields **Comment**, **Contact**, **Cost Center**, **Customer Support Identifier**, **Department**, **Deployment Type**, **Lifecycle Status**, **Line of Business**, **Location**, **Status**, and **Target Version**.

   c. Click **Search**.

3. Select a domain which has the JVM Diagnostics Agent deployed on it:

   a. From the **Target Navigation** area, select **Java Virtual Machine Pools** and expand it.

   Note: All the WebLogic Domains and Servers that have JVM Diagnostics Agents deployed on them appear under this folder.

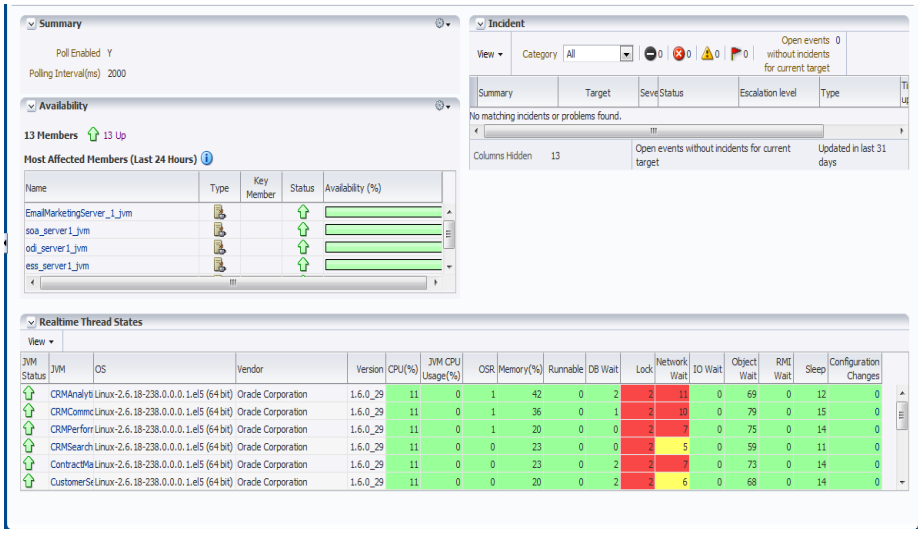
   Note: The WebLogic Domain is suffixed with *_jvmpool* and the WebLogic Servers are suffixed with *_jvm*.

   ---

   b. Select a WebLogic Domain or WebLogic Server under this folder.

4. When you select a WebLogic Domain, select **Java Virtual Machine Pool Home** from the **Java Virtual Machine Pool** menu.

   The JVM Pool Summary page displays.



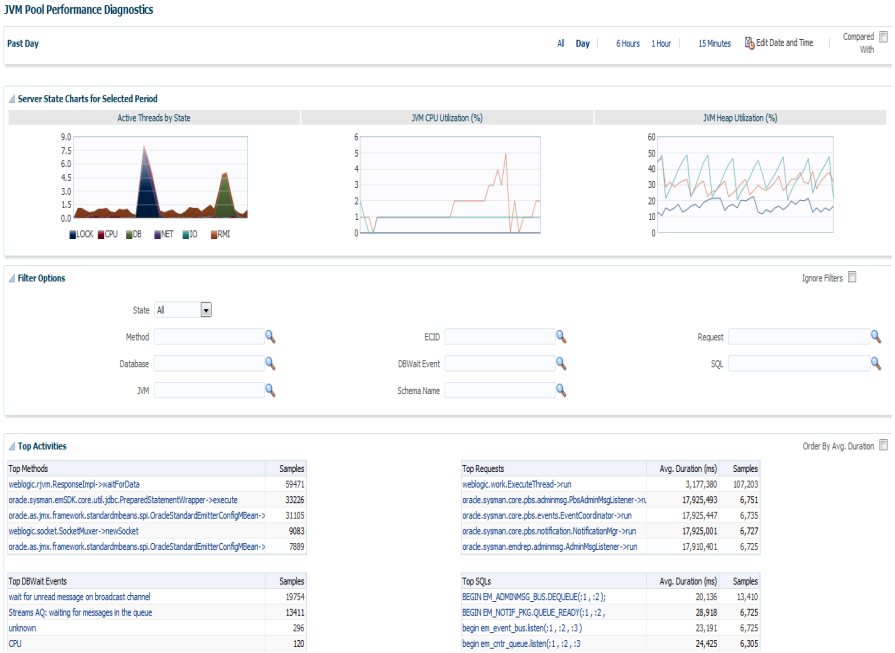
5. Take note of the following:

   - In the **Summary** section, Poll indicates if polling of that JVM Pool has been enabled. Polling Interval denotes the time interval (in milli seconds) at which the JVM Pool is polled.
   - In the **Availability** section, the availability status (in the last 24 hours) of all the JVMs is listed.
   - In the **Realtime Thread States** section, view the overall state of the threads including locks.
   - In the **Incident** section, the servers in the domain whose target status is Down are listed. The incidents will be automatically cleared when the underlying issues are resolved.

   a. From the **Java Virtual Machine Pool** menu, choose **JVM Performance Diagnostics**.

      The JVM Pool Performance Diagnostics page displays. This page provides information about all the JVMs in the pool for the selected time period. You can also modify the time period using Edit Date and Time option at the top of the page.

      The following figure shows the JVM Pool Performance Diagnostics page for the domain:



b. View the various charts:

Server State Charts For Selected Period section

This section displays the Active Threads, CPU Utilization, and Heap Utilization of IO, CPU, lock, and network resources during the selected time. The number of Java threads (daemon and non-daemon) that are currently running in the virtual machine for this Oracle WebLogic Domain:

Active Threads by State: This chart displays the number of Java threads that are currently running in the server. It is color-coded by thread state.

CPU Utilization (%): This chart shows the CPU utilization across the JVM.

Heap Utilization (%): This chart shows the heap utilization across the JVM.

Garbage Collections (Invocations/min): This chart shows the number of times the JVM garbage collector was invoked in the time period per minute. It includes both major and minor garbage collections.

Filter Options section

This section displays the options by which you can filter the JVMs in the pool:

- State
- Method
- Database
- JVM
- ECID
- DBWait Event
- Schema Name
- Request
- SQL.

Top Activities section

This section displays the top activities in the pool:

Top Methods: This chart shows the most expensive Java methods in the selected time period.

Top Requests: This chart shows the top page requests in the selected time period.

Top DBWait Events: This chart shows the cross-tier correlation with the database.

Top SQLs: This chart displays the list of SQL calls ordered by their cost (the number of samples). The SQL calls can also be ordered by average duration by selecting this option in the top right corner of the section.

Top Databases: This chart shows the top impacted databases by the JVMs in this pool.

JVMs In Pool section

This section displays the Active Threads, CPU Utilization, and JVM Heap Utilization of IO, CPU, lock, and network resources during the selected time.

Active Threads: This chart displays the number of active Java threads that are currently running in the domain. It is color-coded by thread state.

CPU Utilization (%): This chart shows the CPU utilization across the JVMs in the pool.

JVM Heap Utilization (%): This chart shows the heap utilization across the JVMs in the pool.

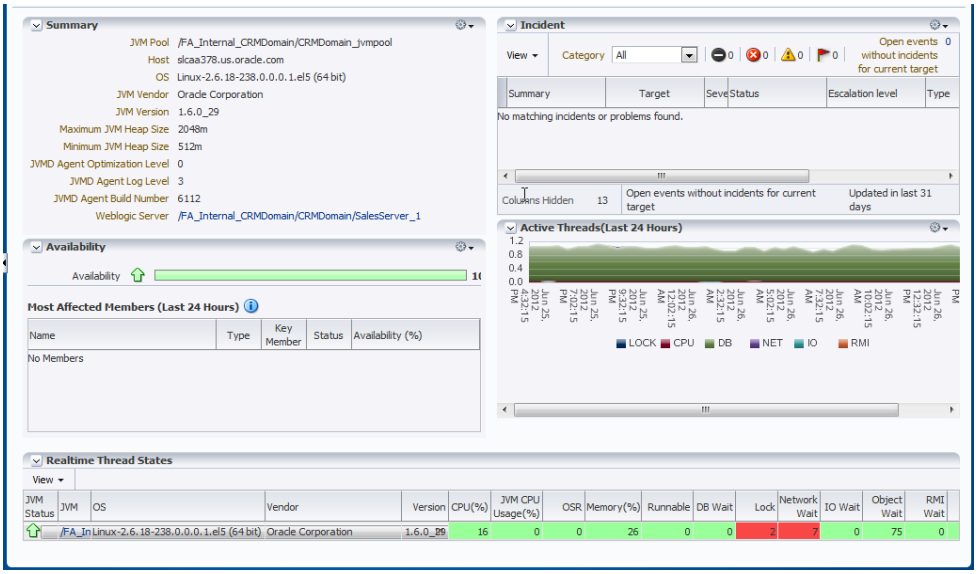
Garbage Collections (Invocations/min): This chart shows the number of times the JVM garbage collector was invoked in the time period. It includes both major and minor garbage collections.

Note: The **Compared with** feature enables you to compare the diagnostics across two specified periods of time.

---

6. When you select a WebLogic Server, select **Java Virtual Machine Home** from the **Java Virtual Machine menu**.

   The JVM Summary page displays.



7. Take note of the following:

   a. From the **Java Virtual Machine** menu, choose **JVM Performance Diagnostics**.

      The JVM Performance Diagnostics page displays. This page provides information for a single JVM for the selected time period. You can also modify the time period using Edit Date and Time option at the top of the page.

      To see the JVM Performance Diagnostics for multiple JVMs, navigate to **Target -> Middleware -> Diagnostic Snaptshots** and click **Create**. Fill all the mandatory fields and add all the JVM targets and create the snapshot. The snapshot shows up in the table. You can select and download the snapshot.

      The following figure shows the JVM Performance Diagnostics page for the JVM:

The **Thread Transition Graph** is displayed as ADF Bar Graph. By default, the data is shown for 30 minutes and the maximum time window is one hour.

From the Create Diagnostics Snaptshots dialog, the administrator can select a time window where performance symptoms occurred and store the data separately for arbitrary use. This data allows the administrator to export the snapshot out of the system.

b. View the various charts:

Server State Charts For Selected Period section

This section displays the Active Threads, CPU Utilization, and Heap Utilization of IO, CPU, lock, and network resources during the selected time. The number of Java threads (daemon and non-daemon) that are currently running in the virtual machine for this Oracle WebLogic Server:

Active Threads by State: This chart displays the number of Java threads that are currently running in the server. It is color-coded by thread state.

CPU Utilization (%): This chart shows the CPU utilization for the JVM.

Heap Utilization (%): This chart shows the heap utilization for the JVM.

Garbage Collections (Invocations/min): This chart shows the number of times the JVM garbage collector was invoked in the time period. It includes both major and minor garbage collections.

Filter Options section
This section displays the options by which you can filter the data:

- State
- Method
- Database
- JVM
- ECID
- DBWait Event
- Schema Name
- Request
- SQL.

General tab
This tab displays data from the JVM:

Active Threads by State: This chart displays the number of Java threads that are currently running in server. It is color-coded by thread state.

Top Requests: This chart shows the top page requests in the selected time period.

Threads tab
This tab displays data from the JVM:

Threads State Transition: This chart shows how the threads have transitioned from one state to the other in the selected period. You can change the time interval and move it to a different time period by using the quick time selection control at the top of the page. You can hover over the colored bars to see the thread state transition changes from one state to another, for example from Runnable to Not Active or Runnable to DB Wait. Click a bar graph in the State column to view a detailed analysis on the state of the thread. This feature allows you to analyze each sample (JVM snapshot at a specific time) in the monitored data.

Metric By Active States: This chart shows how long each of the threads have been in the various states.

Note: The **Compared with** feature enables you to compare the diagnostics across two specified periods of time.

4.2.4 Finding the Top Java Methods

If you have a slow-running application, locate the Java method causing the potential issue.

To find the top Java methods with Cloud Control:

1. From the **Targets** menu, choose **Targets** > **Middleware**.

   The Middleware target home page displays.

2. Search for the Oracle WebLogic Server domains:

a. Select an area under the **Area** drop-down list. For example, Oracle WebLogic Domains, Clusters, and Servers.

b. Enter appropriate information in the fields **Comment**, **Contact**, **Cost Center**, **Customer Support Identifier**, **Department**, **Deployment Type**, **Lifecycle Status**, **Line of Business**, **Location**, **Status**, and **Target Version**.

c. Click **Search**.

3. Click a domain or one of its servers which has JVM Diagnostics agents deployed. For more information about installing JVM Diagnostics, see the *Oracle Enterprise Manager Cloud Control Basic Installation Guide*.

   The WebLogic Server Domain Home page or WebLogic Server Home page displays.

4. From the **WebLogic Domain** menu or **WebLogic Server** menu, choose **Diagnostics** > **JVM Performance Diagnostics**.

   The JVM Pool Performance Diagnostics page displays for an Oracle WebLogic Domain and the JVM Performance Diagnostics page displays for a Managed Server.

5. In the **Top Methods** section, review the list of methods ordered by their cost (the number of samples).

6. In the **Top Methods** section, click a method to add it as a filter.

7. In the Method Stack Tree dialog, click the plus sign next to the method to investigate further.

8. The Method Stack Tree dialog displays an aggregation of all the call stacks with that method as the current call.

9. In the Method Stack Tree dialog, select the method and click **OK**.

   The **Filter Options** section auto-fills the information on the method and the charts update to reflect that method, for example:

   - Heavy use of IO, CPU, lock, and network resources (**Active Threads** chart)
   - Request causing it to be invoked (**Top Requests** chart)
   - SQL or database state it causes (**Top DBStates** and **Top SQLs** charts)

10. After you are done with viewing the method, in **Filter Options** section, clear out the **Method** text box filter and click anywhere to remove the filter.

4.2.5 Finding the Top SQL Queries

If you suspect a slow SQL query is causing a network requests or IO issue, find the slowest SQL queries.

To find the top SQL calls using JVM diagnostics with Cloud Control:

1. From the **Targets** menu, choose **Targets** > **Middleware**.

   The Middleware target home page displays.

2. Search for the Oracle WebLogic Server domains:

   a. Select an area under the **Area** drop-down list. For example, Oracle WebLogic Domains, Clusters, and Servers.

b. Enter appropriate information in the fields **Comment**, **Contact**, **Cost Center**, **Customer Support Identifier**, **Department**, **Deployment Type**, **Lifecycle Status**, **Line of Business**, **Location**, **Status**, and **Target Version**.

c. Click **Search**.

3. Click a domain or one of its servers which has JVM Diagnostics agents deployed. For more information about installing JVM Diagnostics, see the *Oracle Enterprise Manager Cloud Control Basic Installation Guide*.

   The WebLogic Server Domain Home page or WebLogic Server Home page displays.

4. From the **WebLogic Domain** menu or **WebLogic Server** menu, choose **Diagnostics** > **JVM Performance Diagnostics**.

   The JVM Pool Performance Diagnostics page displays for an Oracle WebLogic Domain and the JVM Performance Diagnostics page displays for a Managed Server.

5. In the **Top SQLs** section, review the list of SQL calls ordered by their cost (the number of samples). If the target type is WebLogic Domain or JVM Pool, the SQL calls can also be ordered by average duration by selecting this option in the top right corner of the section.

6. In the **Top SQLs** section, click a SQL call to view the charts for that call.

   The **Filter Options** section auto-fills the information on the method and the charts update to reflect that method. Adding the statement as a filter enables you to see everything related to that SQL call, for example:

   - Methods that invoke it (**Top Methods** chart)
   - Request causing it to be invoked (**Top Requests** chart)
   - Database state it causes (**Top Databases** chart)

7. After you are done with viewing the method, in **Filter Options** section, clear out the **SQL** field and click anywhere to remove the filter.

4.2.6 Analyzing Live Stuck Threads

If application users report a spinning status indication after clicking in the application, investigate the live stuck threads. Follow the procedure given below to find and analyze the live stuck threads.

To find and analyze the live stuck threads:

1. From the **Targets** menu, choose **Targets** > **Middleware**.

   The Middleware target home page displays.

2. Search for the Oracle WebLogic Server domains:

   a. Select an area under the **Area** drop-down list. For example, Oracle WebLogic Domains, Clusters, and Servers.

   b. Enter appropriate information in the fields **Comment**, **Contact**, **Cost Center**, **Customer Support Identifier**, **Department**, **Deployment Type**, **Lifecycle Status**, **Line of Business**, **Location**, **Status**, and **Target Version**.

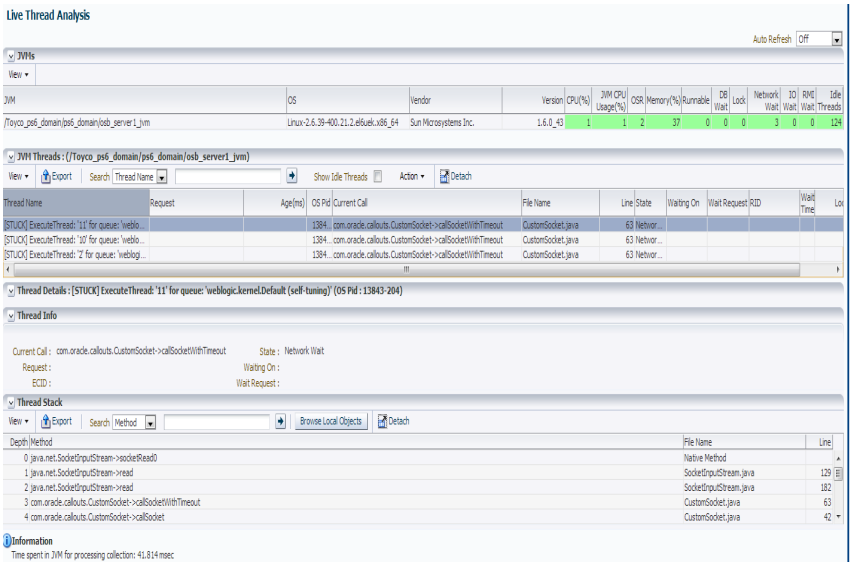
   c. Click **Search**.

3. Click a domain or one of its servers which has JVM Diagnostics agents deployed. For more information about installing JVM Diagnostics, see the *Oracle Enterprise Manager Cloud Control Basic Installation Guide*.

   The WebLogic Server Domain Home page or WebLogic Server Home page displays.

4. From the **WebLogic Domain** menu, choose **Diagnostics** > **Live Thread Analysis**.

   The Live Thread Analysis page displays.



5. In the **JVMs** section, select a thread to display details in the **JVM Threads** section.
6. In the **JVM Threads** section, look for a thread having the prefix **[STUCK]** and click it.
7. In the **Thread Info** and **Thread Stack** sections, look at the **Current Call**, **File Name**, **Line**, and **State** for the thread.

   This information provides you with the key information on how to locate the code that is causing the problem:

   - **Current Call:** This field displays the name of the method call where the code is stuck.
   - **File Name:** This column identifies the file with the problem.
   - **Line:** This column identifies the line number in the file where the problematic code exists.
   - **State:** This column displays the state of the thread (for example, CPU, IO, Network, DB Wait, Lock, and so on).

8. Look for the **Lock Held** in the **Thread Info** section.
9. If the stuck thread is in the **DB Wait** state, then click the link and go directly to the database session to see what that thread is doing in the database, or use the technique described in Section 4.2.8.

4.2.7 Analyzing Stuck Threads from Specific Time Periods

The procedure listed in this section allows you to find and analyze the stuck threads in the historical context.

To find the stuck threads in the historical context:

1. In **Target Navigation** area, expand **Java Virtual Machine Pools** > *domain_name_* **jvmpool**.
2. Select a Java Virtual Machine.
3. From the **Java Virtual Machine** menu, choose **JVM Performance Diagnostics**.

4. In the **Filter Options**, enter **[STUCK]%** in the **Thread Name** field.
5. In the **Threads** sub-tab, drag the pointer to select the timeline.

> **Note:** Scroll left and right to select the appropriate date. After you select the date, select the required duration.

4.2.8 Drilling Down from JVM Diagnostics to SQL Instances

If you have a webpage that does not return the intended results, and when you diagnose that page using the JVMD to notice that an SQL query that was issued did not execute, you must analyze the SQL statement.

To analyze SQL from Cloud Control:

1. From the **Targets** menu, choose **Targets** > **Middleware**.

   The Middleware target home page displays.
2. Search for the Java Virtual Machines:
   a. From the **Search** area, select **JVMs** under the **Area** drop-down list.

b. Enter appropriate information for **Comment**, **Contact**, **Cost Center**, **Customer Support Identifier**, **Department**, **Deployment Type**, **Lifecycle Status**, **Line of Business**, **Location**, **Status**, **Target Version**.

c. Click **Search**.

3. Select any JVM Pool.

   The JVM Pool Home page displays.

4. Click **JVM Performance Diagnostics**.

5. In the **Top SQLs** table under **Top Activities**, click any SQL statement to view the SQL Details.

6. Click the Database and login with appropriate credentials to view the SQL Details.

7. Click the search icon next to SQL field. Set the resulted SQL Query as filter by selecting it.

8. Analyze the SQL.

4.2.9 Analyzing Potential Memory Leaks

To find and analyze memory leaks, you can use Cloud Control to take and analyze snapshots of the heap.

Analyzing heap requires a large amount of free space in the Oracle Database tablespace being used. As a standard practice, ensure you have five times the size of heap dump file being loaded in the tablespace. Since you know the size of your dump file, make sure that there is adequate space to accommodate the dump file before it is loaded into the database.

To use heap snapshot to analyze memory leaks:

- Task 1, "Review Metrics for a Potential Memory Leak"
- Task 2, "Perform a Live Heap Analysis"
- Task 3, "Take a Heap Snapshot"

Task 1 Review Metrics for a Potential Memory Leak

To create a snapshot of the heap for later loading and examination for leaks:

1. From the **Targets** menu, choose **Targets** > **Middleware**.

   The Middleware target home page displays.

2. Search for the Oracle WebLogic Server domains:

   a. From the **Search** area, select **Oracle WebLogic Servers** under the **Area** drop-down list.

   b. Enter appropriate information for **Comment**, **Contact**, **Cost Center**, **Customer Support Identifier**, **Department**, **Deployment Type**, **Lifecycle Status**, **Line of Business**, **Location**, **Status**, **Target Version**.

   c. Click **Search**.

3. Click a Java Virtual Machine with JVM Diagnostics agents deployed. For more information about installing JVM Diagnostics, see the *Oracle Enterprise Manager Cloud Control Basic Installation Guide*.

   The Java Virtual Machine Home page displays.

4. From the **Java Virtual Machine** menu, choose **Performance Summary**.

The Performance Summary page displays.

5. Review the following metrics for any periods of time where the **Warning Thresholds** or **Critical Thresholds** were reached:

JVM GC Overhead	This metric is useful in understanding the efficiency of the JVM when it executes the application logic. If the GC overhead is too high (10% could be considered as high usage), it indicates that the JVM is spending substantial time on GC while blocking the service threads from responding to other requests.
JVM Heap Usage (%)	This metric shows the heap utilization for the JVM. This metric provides an indicator of heap size as they fluctuate between garbage collections.
JVM Heap Used After GC	This metric is the most relevant indicator of an application's memory leak. This metric can show a trend of growth in heap consumption and, in combination with other load/demand metric, can provide very useful prediction about the rate of the leak. For example, if the chart trending up while the application load is stable, then it is possible there is a leak.

If any of the metrics exceed the **Warning Thresholds** or **Critical Thresholds**, it could indicate memory is a factor in the JVM performance and availability. It could mean there is a memory leak or that the JVM heap configuration is too small for the application load. If the heap configuration is correct, assume there is a leak and investigate the cause.

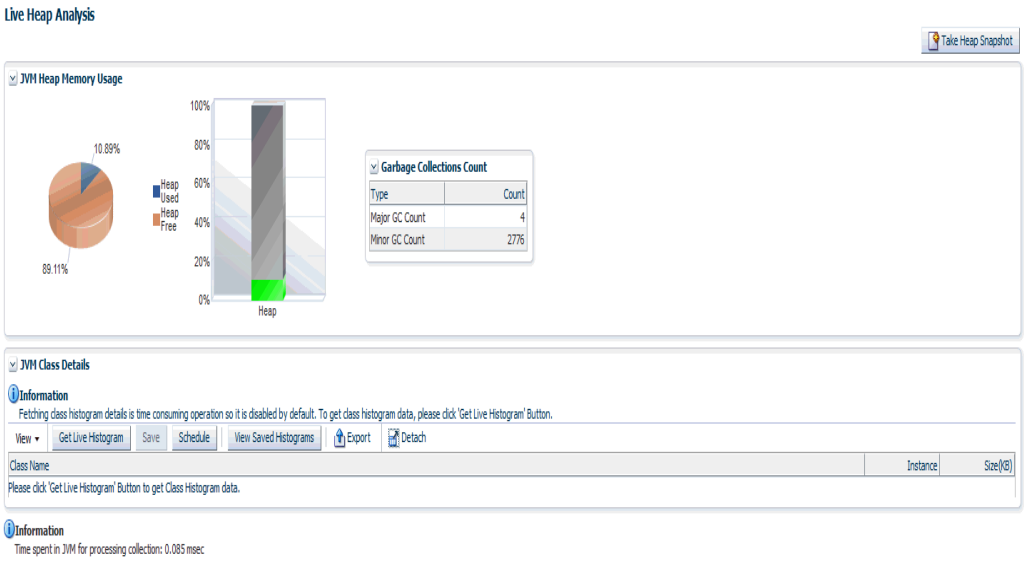
6. If any of the metrics exceed the **Warning Thresholds** or **Critical Thresholds**, proceed to Task 2.

Task 2 Perform a Live Heap Analysis

To create a snapshot of the heap for later loading and examination for leaks:

1. From the **Java Virtual Machine** menu, choose **Live Heap Analysis**.

   The Live Heap Analysis page displays.



2. Review the top portion of the page to the analyze the heap and the number of objects added to the garbage collector; review the **JVM Class Details** table to review the largest-size objects in the heap.

   For more information about using this page, see the topic "Viewing the Real-Time Heap Data" in the Cloud Control online help.

Task 3 Take a Heap Snapshot

To take a snapshot of the heap for later loading and examination of leaks:

1. On the Live Heap Analysis page, click **Take Heap Snapshot**.

   The Load Heap Snapshot page displays.

2. Under the **Heap Snapshot Formats and Analysis Options** section:

   1. Notice that the **Heap Snapshot Type** option enables you to pick either JVMD Format (txt) or a HPROF Format (binary) for use with other tools.

   2. The **Heap Analysis Options** drop-down list enables to select the appropriate type of heap analysis to be done. Leave this field empty to perform the heap dump only. Select at least one of the following options to perform heap analysis and load heap data:

      a. **All** - performs heap analysis

      b. **Load Heap Data to Repository** - loads the heap data to the repository

      c. **Memory Leak Report** - generates a memory leak report

   For more information about using this page, see the topic "Taking a Heap Snapshot" in the Cloud Control online help.

3. When you select any heap analysis option, the following sections are enabled:

   a. **Heap Snapshot Time**

      This option enables you to either take a snapshot at the current time or schedule it for a later time.

   b. **JVM Diagnostics Agent Host Credentials**

      Select either **Preferred** (Normal or Privilege), **Named** (credentials saved earlier), or **New** (enter new credentials) credentials to login to the JVM Diagnostics Agent Host.

   c. **Heap Analysis Host**

      – Select a saved heap analysis host.

      – Add a host for heap analysis

      – Edit a saved heap analysis host

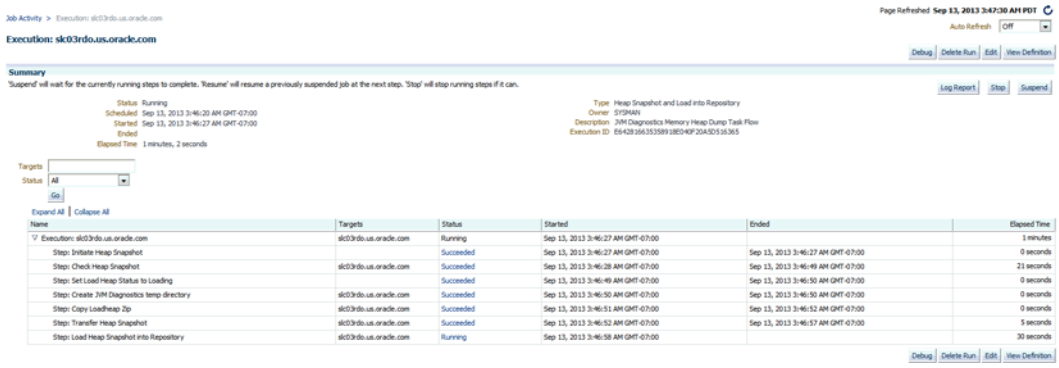
      – Remove a saved heap analysis host.

   d. **Enter Heap Analysis Host Credentials**

      Specify credentials for host where JVM Diagnostics Heap Analysis and Loading is running. Select either **Preferred** (Normal or Privilege), **Named** (credentials saved earlier), or **New** (enter new credentials) credentials to login to the Heap Analysis Host.

4. Click **Submit**. A confirmation window to load the heap appears.

5. When you select **Yes** on the confirmation dialog, Heap Analysis Job page appears from where you can see the JVM DIagnostics Memory Heap Dump Task Flow.



   a. Click **View Definition** to see the details of the job.

   b. Click **Edit** to modify the parameters for running the job.

   c. Click **Delete** Run to delete the memory heap dump job.

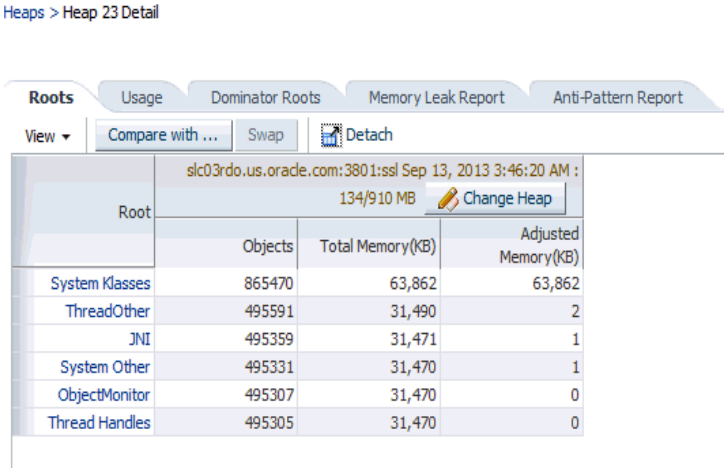
   d. Click **Log Report** to view the report of the memory heap dump job.

6. From the **Java Virtual Machine** menu, choose **Heap Snapshots and Class Histograms**.

   You can see that the saved snapshots are listed under the **Available Heap Snapshots** section.

7. Select the heap you require from the table and click **Detail**.

   The Heaps > Roots page displays. The **Roots** tab displays the objects reachable by roots, which are objects that are directly reachable from the JVM itself.



8. Click a root name to drill down and view the objects that consume a lot of memory.

   The Top 40 Objects page displays.

9. In the **Heaps > Details** page, click **Compare with** to compare the current heap with another heap dump taken previously.

   When comparing heaps, select the bigger one first. Otherwise you may see negative deltas.

10. In the Select a heap record dialog, select the second heap, and then click **OK**.

11. Compare both the heaps. Compare the number of objects (**Objects**) and the occupied memory size (**Adjusted Memory**) in each heap dump. This measure indicates the objects that are growing over the period when the snapshots were taken.

12. Drill down into the root which had the largest delta to find the biggest memory leak.

4.3 Working with Class Histograms

A class histogram is displayed in the form of a table when the optimization level of the jamagent is 0. The histogram displays the top 300 data rows sorted by the size. This feature allows you to monitor histograms for the analysis of the Virtual Machine.

4.3.1 Viewing Class Histogram

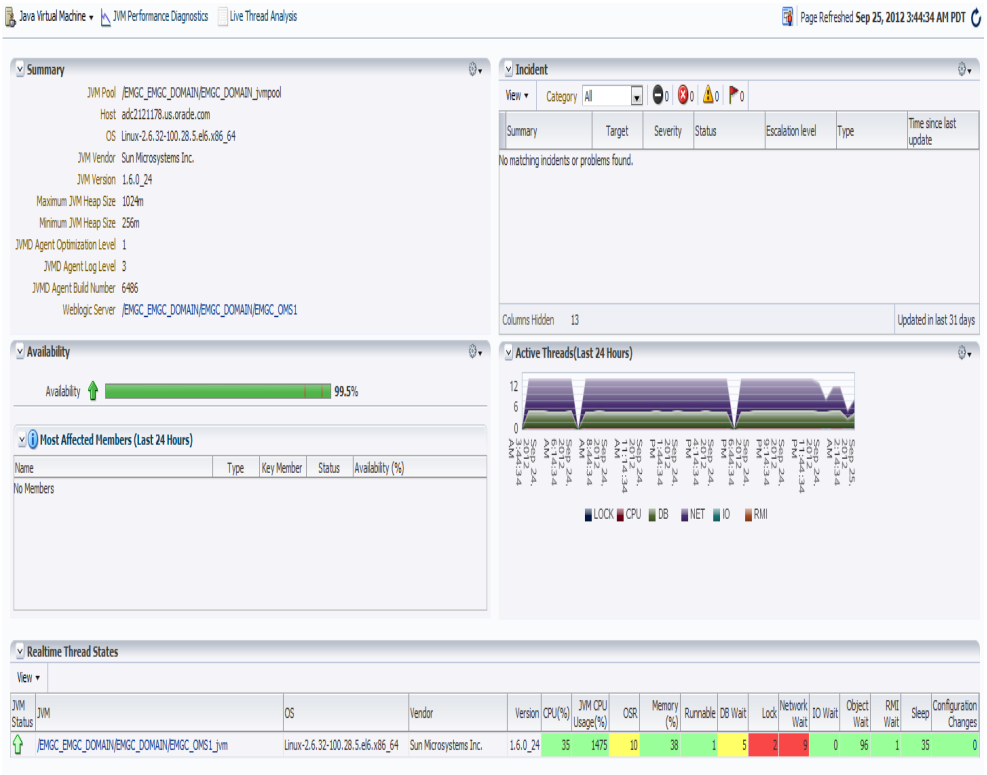
To get started with viewing the Class Histogram:

1. From the **Targets** menu, choose **Targets** > **Middleware**.

   The Middleware target home page displays.

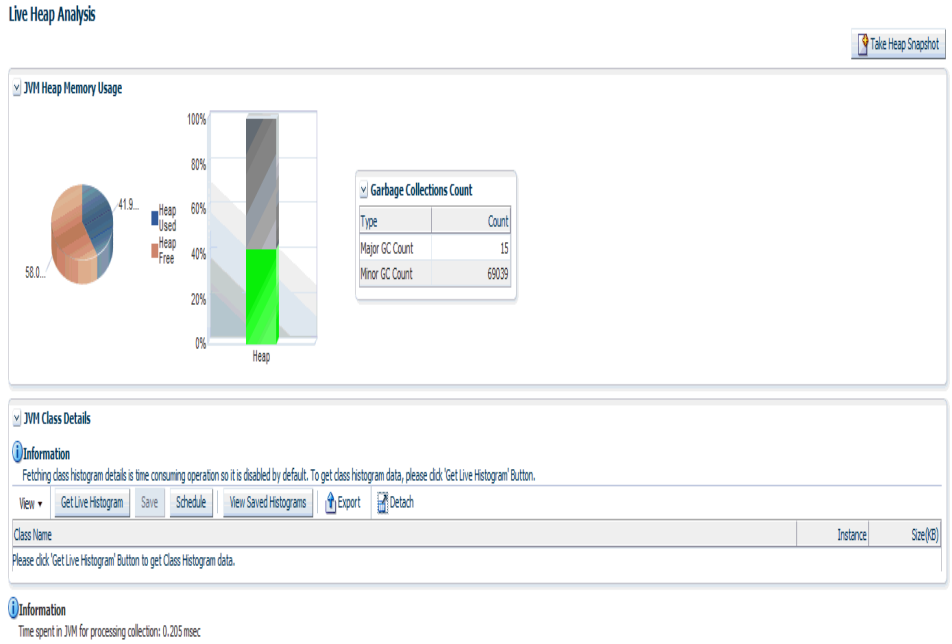
2. Select a Java Virtual Machine which is up and running.

   The Java Virtual Machine Home page displays.

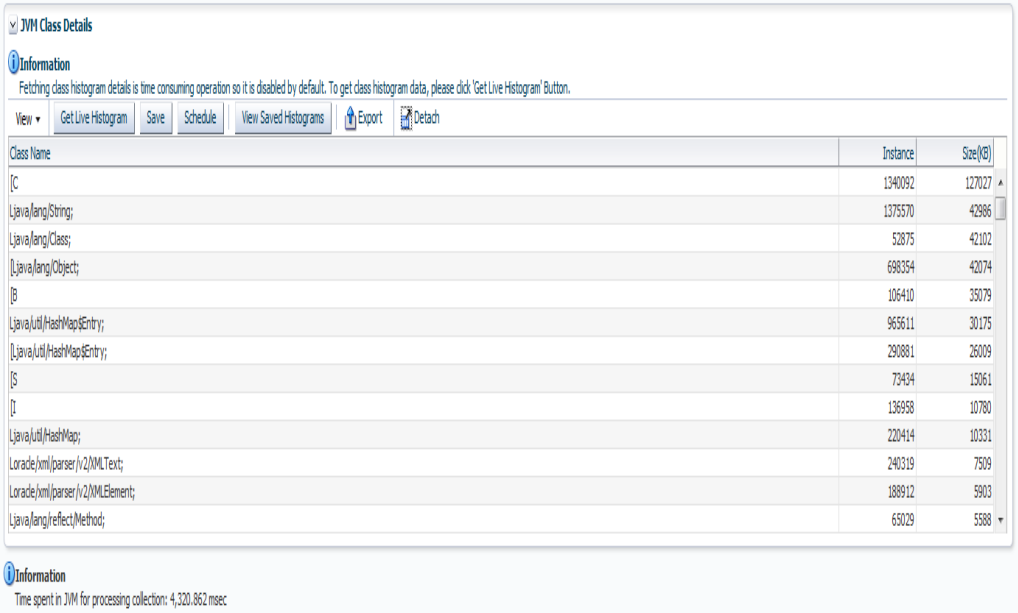


3. Select **Live Heap Analysis** from the **Java Virtual Machine** menu.

   The Live Heap Analysis summary page displays. The JVM Class Details are displayed in the form of a table on this page.



4. Click **Get Live Histogram** to get class histogram data.



4.3.2 Saving Class Histogram

To get started with saving the Class Histogram:

1. From the **Targets** menu, choose **Targets** > **Middleware.**

   The Middleware target home page displays.

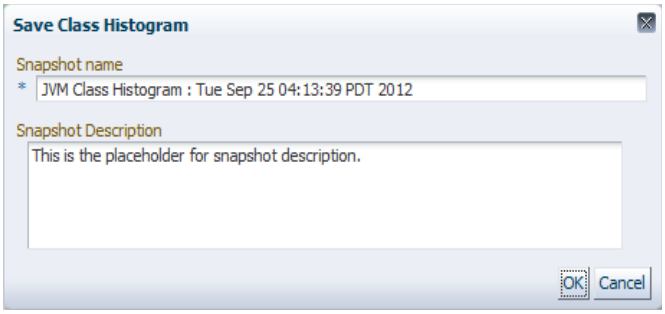
2. Select a Java Virtual Machine which is up and running.

   The Java Virtual Machine home page displays.

3. Select **Live Heap Analysis** from the **Java Virtual Machine** menu.

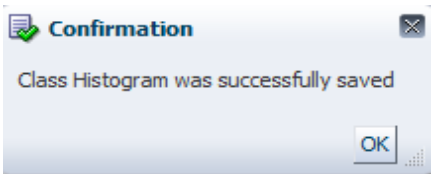
   The JVM Class Details are displayed in a tabular form.

4. Click **Save**.

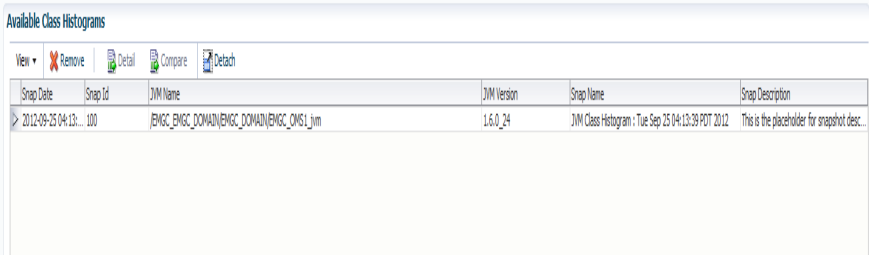


5. Enter a name for the Snapshot in the corresponding field.
6. Provide a description for the snapshot.

7. A confirmation window appears.



8. Click **View Saved Histograms** to view a list of available Class Histograms.



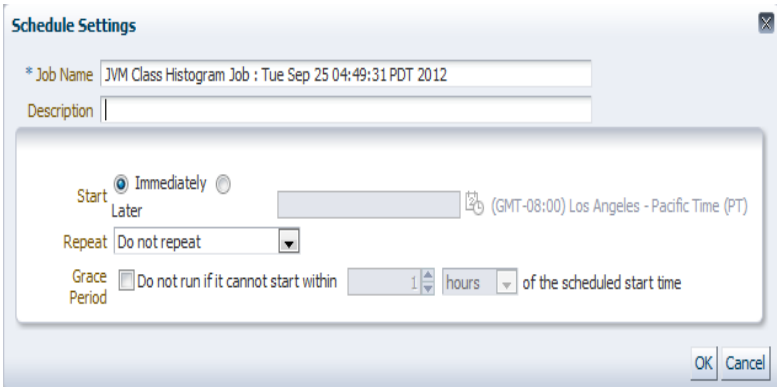
4.3.3 Scheduling Histogram Jobs

Scheduling will allow you to insert JVM Class Histogram data into the repository by running the job at the defined time.

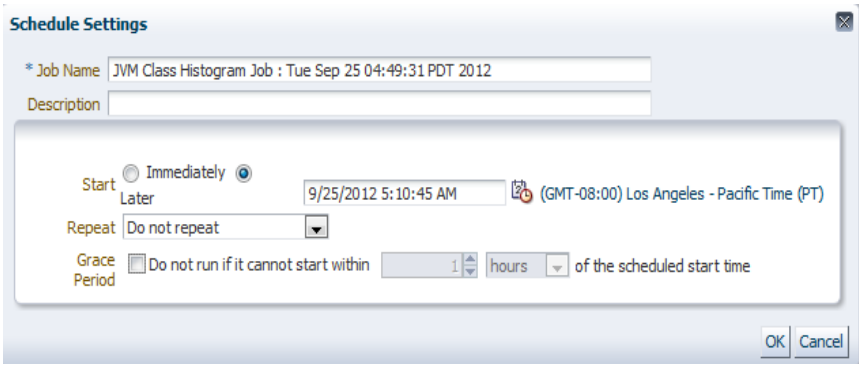
To get started with scheduling the running of histogram jobs:

1. Navigate to Live Heap Analysis summary page.
2. Click **Schedule** under the **JVM Class Details**.

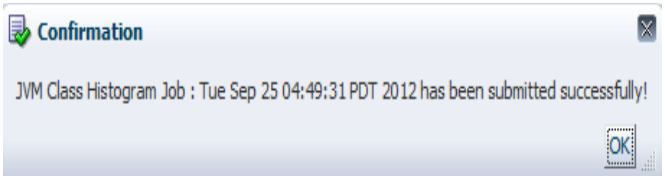
   The Schedule Settings window appears.



3. Enter a name and description for the job to be scheduled.
4. Select when to start the job.
   - If you want to schedule the job with immediate effect, select **Immediately**.
   - If you want to save the job to be run in future, select **Later**. Enter a date and time for the job.



5. Select the frequency at which you want to repeat the job from the **Repeat** drop-down list.
6. Select the option for the **Grace Period**. If you select the grace period, the job will remain active and run within the specified grace period.
7. Click **OK** to schedule the histogram job.
8. A confirmation window appears.



4.3.4 Comparing Class Histograms

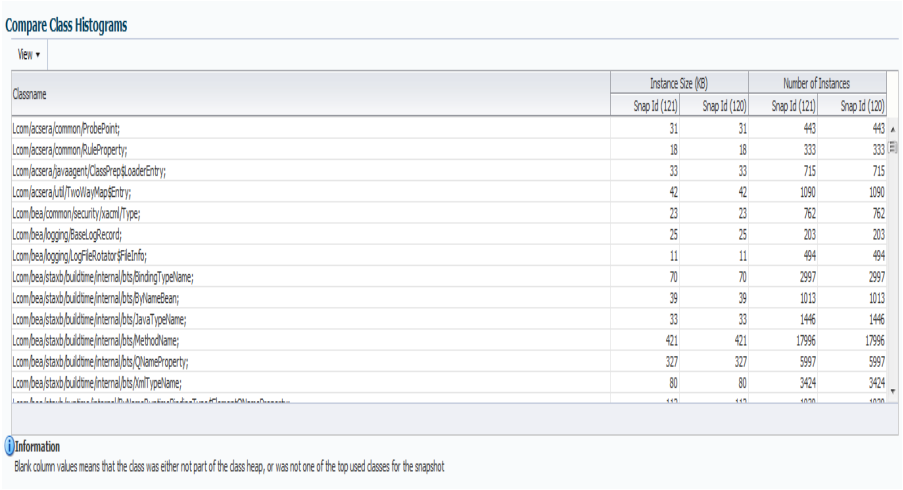
The compare functionality allows you to compare any two class histogram snapshots listed in the table.

To get started with comparing the class histogram snapshots:

1. Navigate to Live Heap Analysis page.
2. Click **View Saved Histograms** under **JVM Class Details**.

3. Select any two class histograms (using the *Ctrl* key) under **Available Class Histograms** and click **Compare**.

   The Compare Class Histograms page appears.



   The comparison table displays the following details:

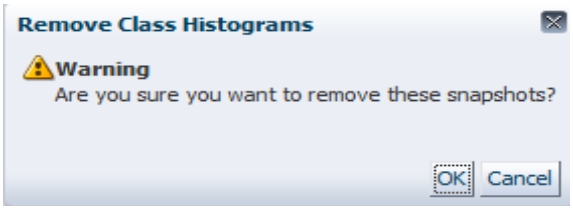
   - Class name
   - Instance size for both the snapshots
   - Number of instances for both the snapshots.

4.3.5 Deleting Class Histograms

The remove functionality allows you to delete the saved class histograms from the repository.

To get started with deleting class histograms:

1. Navigate to the Live Heap Analysis summary page.
2. Click **View Saved Histograms** under **JVM Class Details**.
3. Select the class histogram you want to delete from **Available Class Histograms**.
4. Click **Remove**.
5. Select **OK** in the confirmation window that appears.



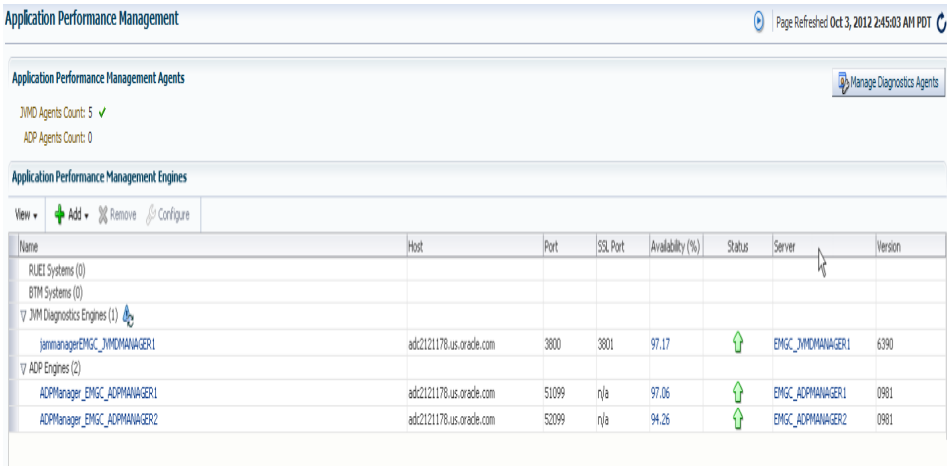
4.4 Deploying/Upgrading/Removing Bulk Diagnostics Agents

Bulk deployment/upgradation of JVMD Agents overcomes the limitation of deploying/upgrading a JVMD agent on a single domain. This feature facilitates the management of deploying/upgrading JVMD agents on all discovered WebLogic domains using a single wizard.

To get started with deploying bulk diagnostic agents:

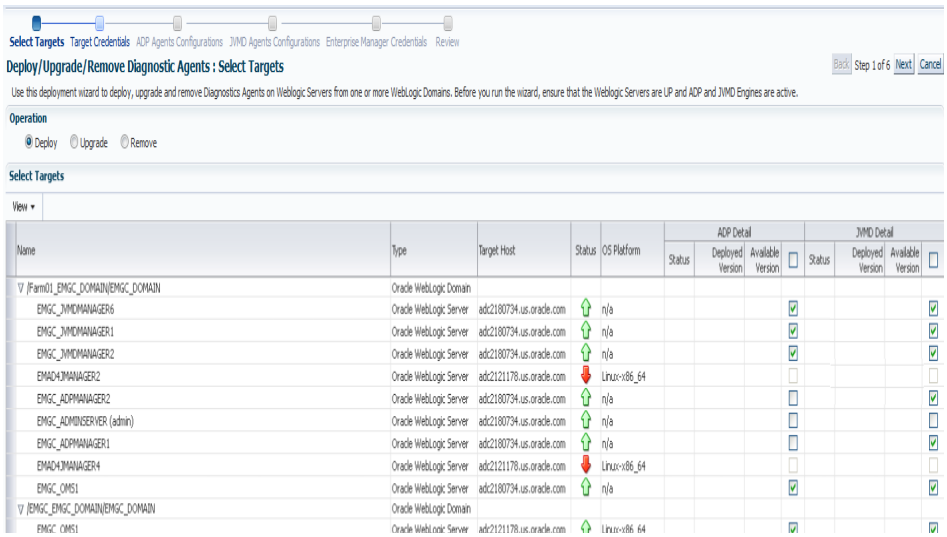
1. From the **Setup** menu, select **Middleware Management > Application Performance Management**.

   The Application Performance Management page appears. The number of JVMD Agents and ADP Agents deployed is displayed under **Application Performance Management Agents**.



2. Click **Manage Diagnostic Agents** to deploy diagnostic agents.

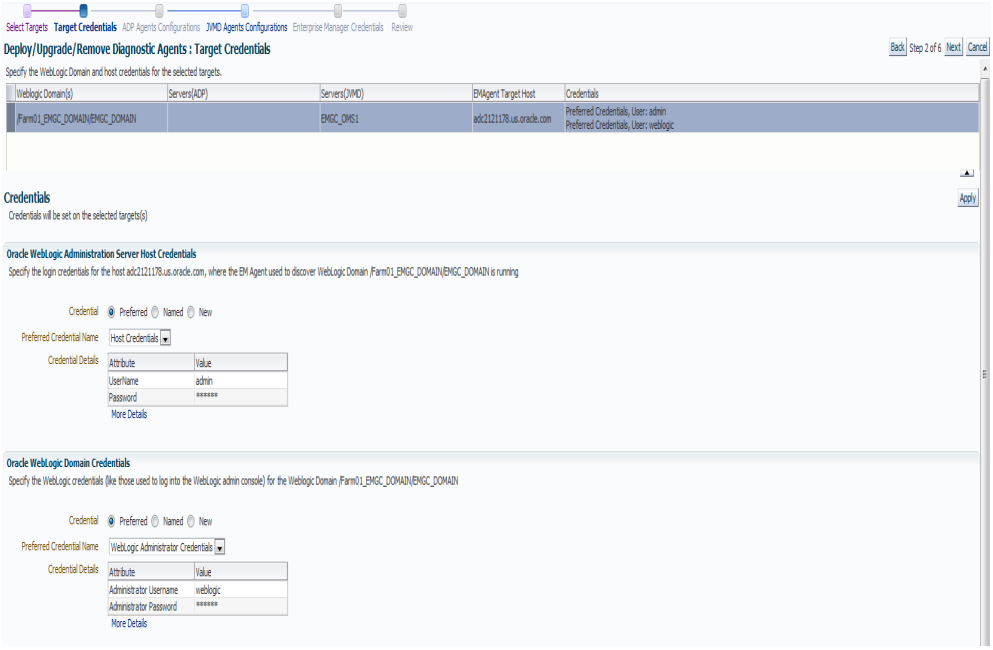
   The Select Targets page appears.



3. Select **Deploy** under **Operation**.

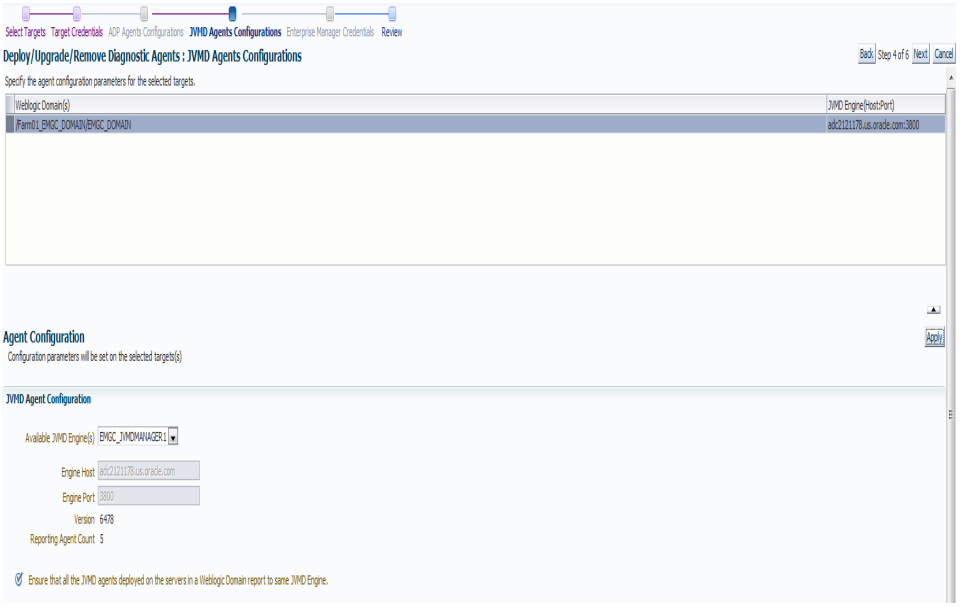
4. Select the targets on which you want to deploy ADP and JVMD agents and click **Next**.

   The Target Credentials page appears.

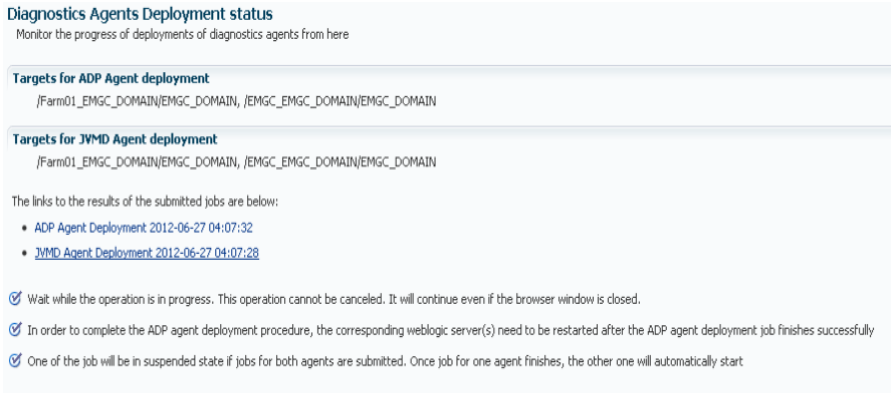


5. Select each target and specify the WebLogic Administration Server Host Credentials and Weblogic Domain Credentials.
6. Click **Apply** to save the changes made to the target credentials and click **Next**.
7. Select each domain and specify ADP Agent Configuration details. Click **Apply** to apply the changes.
8. Click **Next**.

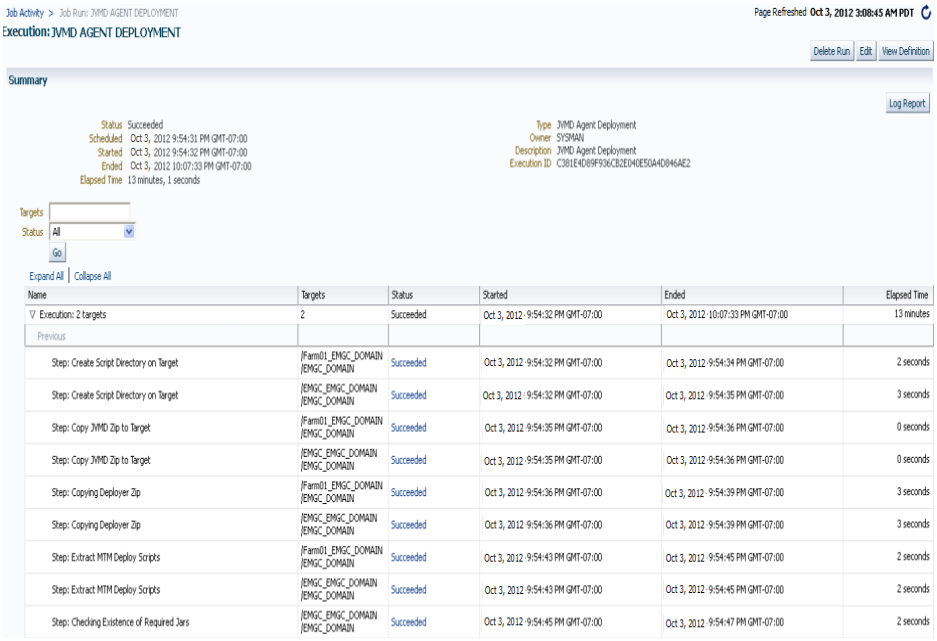
   The JVMD Agents Configurations page appears.



9. Select each domain and specify the JVMD Agent Configuration details. Click **Apply**.
10. Click **Next**.
11. Specify the Host and WebLogic domain credentials for Enterprise Manager OMS.
12. Click **Save** to apply the changes made to the credentials and click **Next**.
13. Review all the target details, credentials, and engine details and click **Deploy**.
14. Click **JVMD Agent Deployment Job** on Diagnostics Agents Deployment Status to view the progress of the job.



15. A summary of the JVMD Agent Job progress appears.



To get started with upgradation of ADP and JVMD Agents:

1. Select **Upgrade** under **Operation** on Select Targets page to upgrade ADP and JVMD Agents to the latest version. Click **Next**.

> **Note:** You can upgrade agents only for WebLogic Server targets.

2. Specify Target Credentials. Click **Next**.

3. Specify Agent Configuration details. Click **Next**.
4. Review the target details, credentials, and engine details. Click **Upgrade**.

   You can view the progress of the agent deployment job.

To remove ADP and JVMD Agents:

1. Select **Remove** under **Operation** on Select Targets page to delete the deployed ADP and JVMD Agents. Click **Next**.
2. Specify Target Credentials. Click **Next**.
3. Specify Agent Configuration details. Click **Next**.
4. Review the target details, credentials, and engine details. Click **Remove**.

   You can view the progress of removal of diagnostics agents on the Diagnostics Agents Removal Status page.

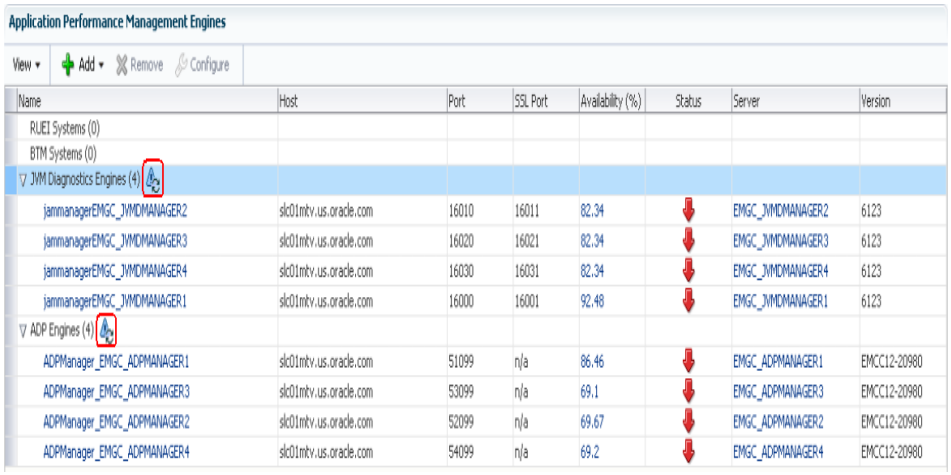
4.5 Upgrading/Redeploying/Removing ADP and JVMD Engines

You can upgrade/redeploy/remove the selected ADP/JVMD Engines on local or existing remote WebLogic Servers to the latest available version. When the upgrade is completed, the upgraded engine(s) gets redeployed.

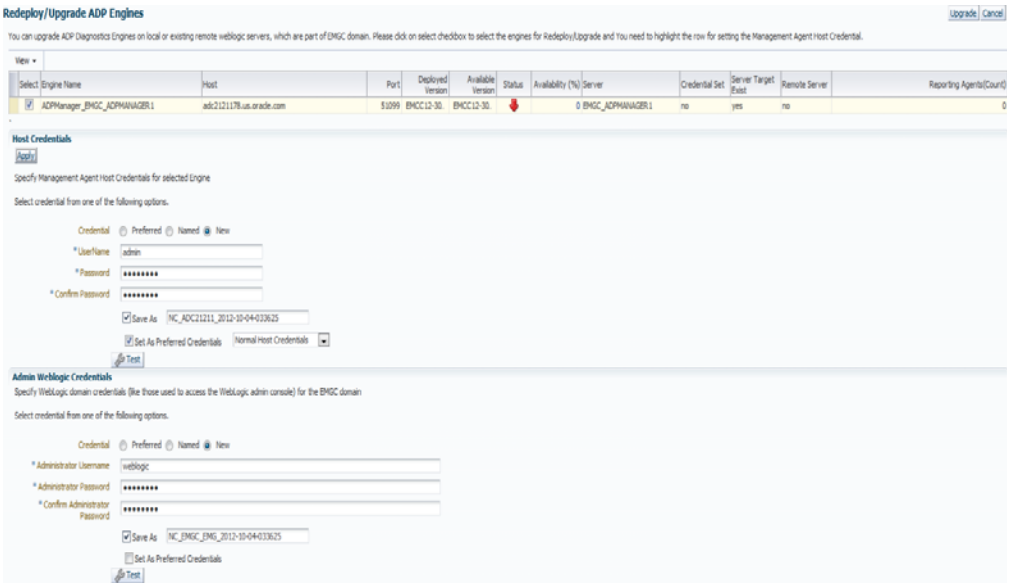
To get started with upgrading the selected ADP/JVMD Engines:

1. From the **Setup** menu, select **Middleware Management > Application Performance Management**.

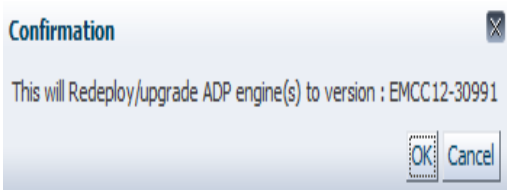
   The Application Performance Management page appears. The available JVMD Engines and ADP Engines are listed under **Application Performance Management Engines**.



2. Click the **Upgrade** icon next to JVMD Engines or ADP Engines to upgrade the corresponding engine(s) or click the **Upgrade** button.
3. Select the engine you want to upgrade.

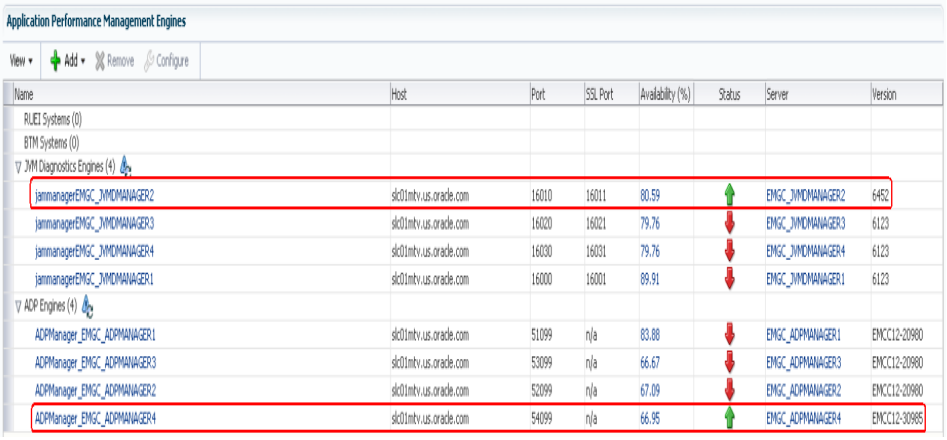


4. Specify the Management Agent Host and WebLogic domain credentials for the selected engine(s). Click **Apply** and then **Upgrade**.
5. Click **OK** on the confirmation window to redeploy the selected engine.



6. Click the job link to view the upgrade job progress.

   The summary of job execution is displayed.
7. When you navigate back to Application Performance Management page, you can see that the selected engines have been upgraded to the latest available version.



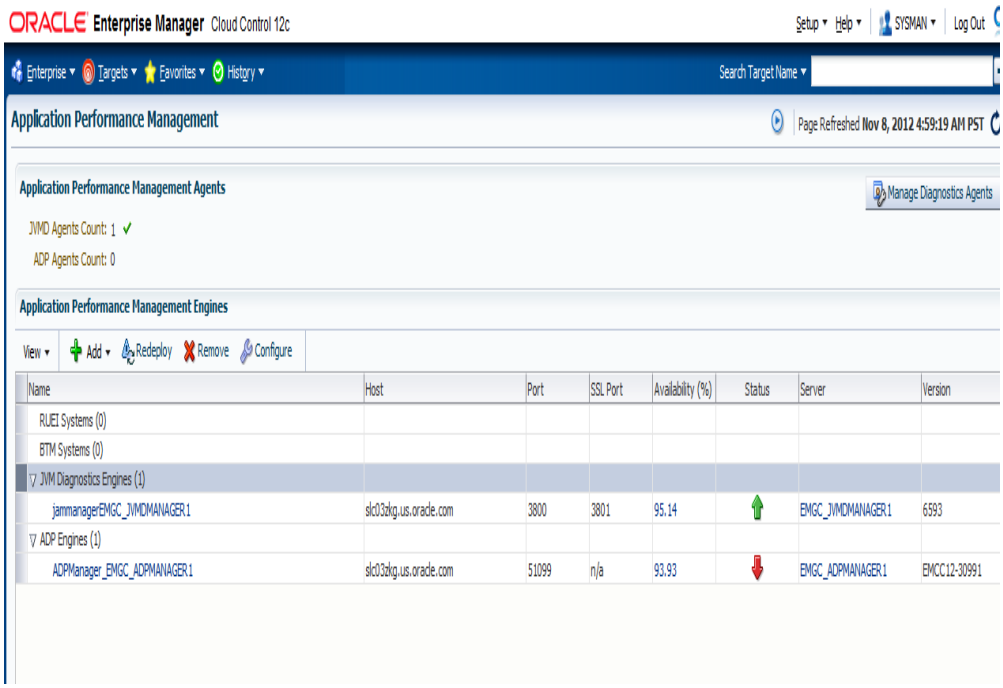
When there is no newer version available for upgrade, you have the option to redeploy the same engine.

To redeploy the selected ADP/JVMD Engines:

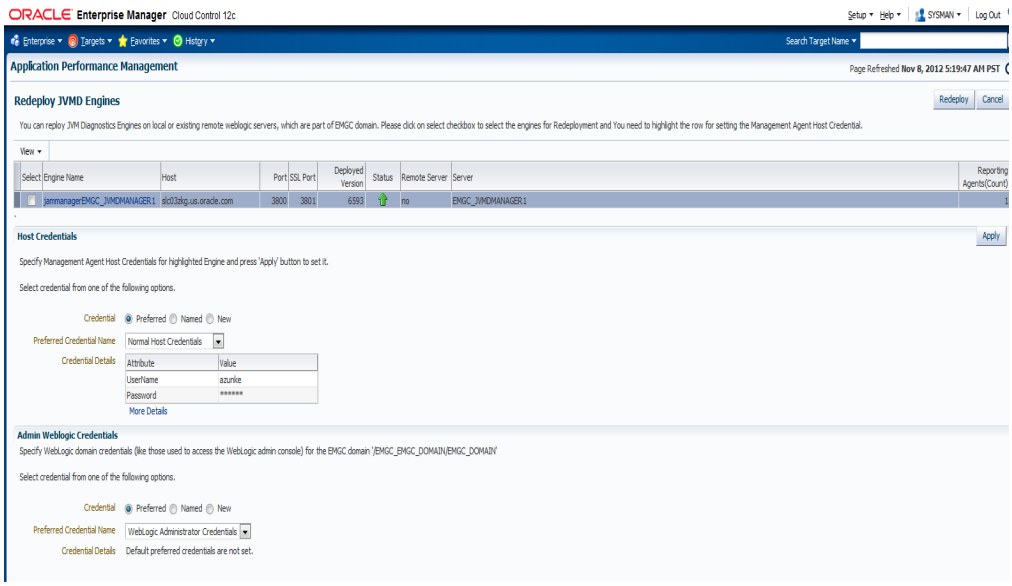
1. From the **Setup** menu, select **Middleware Management > Application Performance Management**.

   The Application Performance Management page appears. The available JVMD Engines and ADP Engines are listed under **Application Performance Management Engines**.

2. Click the **Redeploy** button to redeploy the corresponding engine(s).



3. Select the engine you want to redeploy.



4. Specify the Management Agent Host and WebLogic domain credentials for the selected engine(s). Click **Apply** and then **Redeploy**. The selected engine is redeployed.

5. Click the job link to view the job progress.

Figure 4–1 Progress Page for JVMD Engine Redeployment

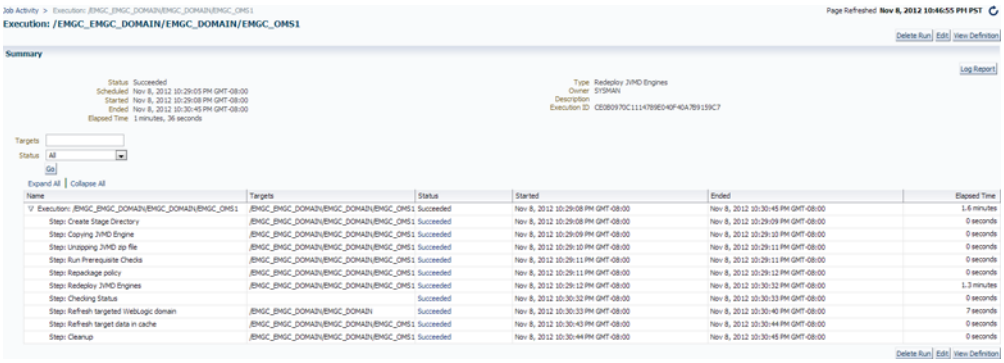
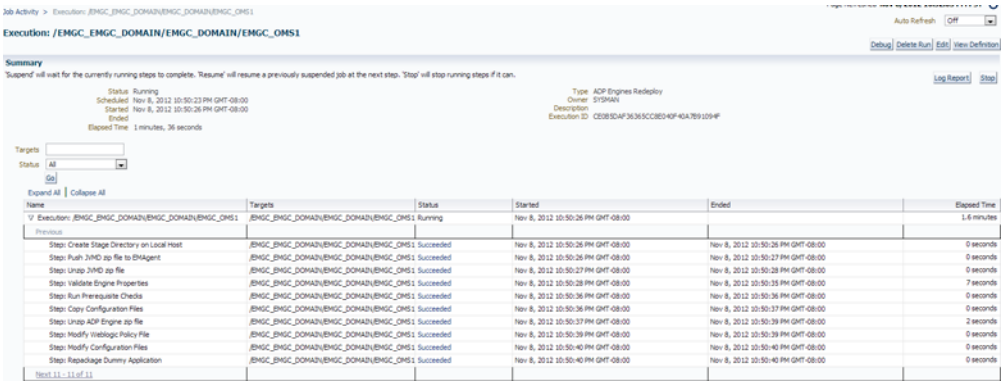


Figure 4–2 Progress Page for ADP Engine Redeployment



To remove the selected ADP/JVMD Engines:

1. From the **Setup** menu, select **Middleware Management > Application Performance Management**.

   The Application Performance Management page appears. The available JVMD Engines and ADP Engines are listed under **Application Performance Management Engines**.

2. Select the required engine by selecting the corresponding row. Click the **Remove** button to delete the corresponding engine(s).

3. Select the engine(s) you want to remove. Select Remove WebLogic Managed Server to remove the managed weblogic server along with the selected engine.

4. Select the appropriate Admin WebLogic Host Credentials and Admin WebLogic Credentials.

5. Click **Remove**.

6. Select **Yes** on the Edit Lock Warning dialog, if you want to proceed with the removal of selected engine.

7. Check the details on the Confirmation dialog and click **OK**.

8. Click the link in the Engine remove job status dialog to view the progress of the job.

9. Click OK in the Engine remove job status dialog to complete the removal process.

4.6 Working with Pool Threshold Violations

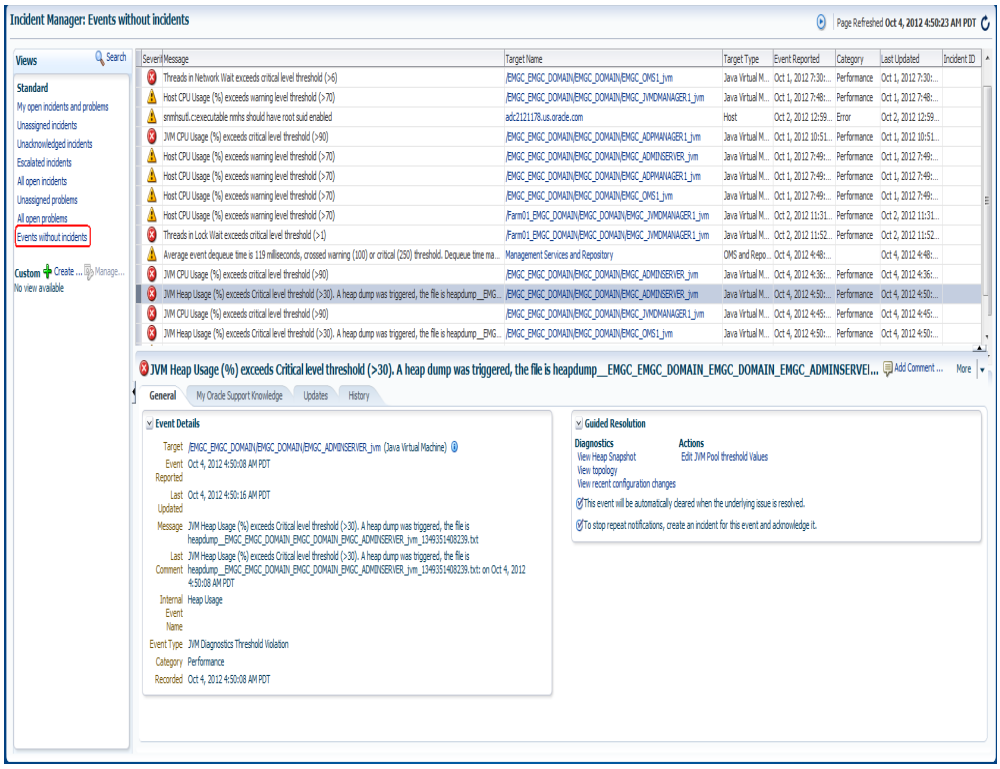
This feature allows you to monitor the Pool Threshold Violations and take corrective actions for the violations. The Threshold Violations and the corresponding Corrective Actions are displayed on the EM Event page. You have the option to setup the required corrective action for each of the threshold violations.

4.6.1 Viewing Threshold Violations

To get started with viewing the threshold violations:

1. From the **Enterprise** menu, select **Monitoring** > **Incident Manager**.
2. Select **Events without incidents** from the **Views** navigation pane.

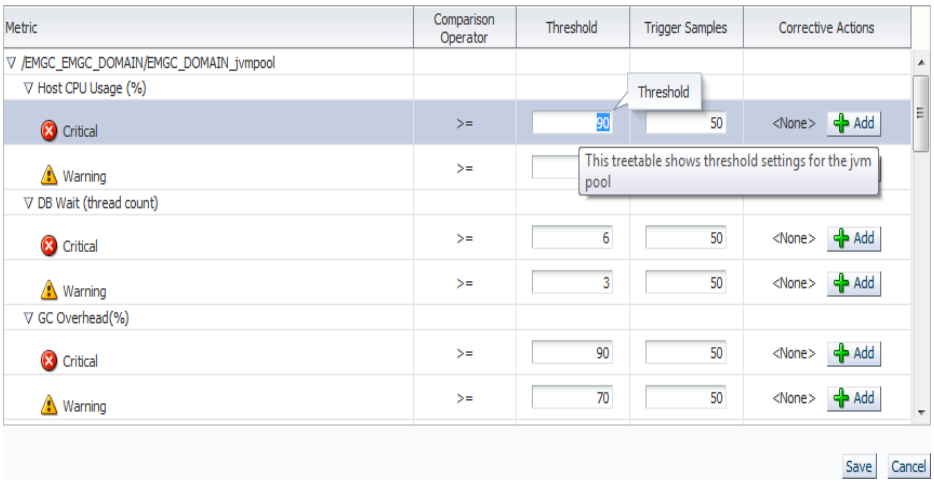
   The Incident Manager: Events without incidents page appears.



3. Select any message, for which the target Type is Java Virtual Machine, to view its details.
4. Select **Edit JVM Pool threshold values** under **Guided Resolutions > Actions** within the **General** tab to setup corrective actions for the selected threshold violation.

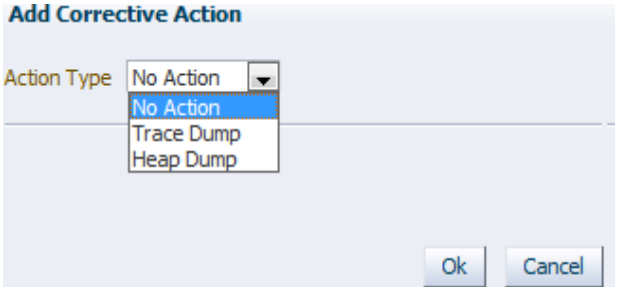
   The JVM Pool threshold value page appears. You can also navigate to this page through the **Configure JVM Pool** option in the context menu.

   The pool threshold tree table contains information of threshold values and corresponding corrective actions for a Java Virtual Machine Pool target. Values of Threshold and Trigger Samples values can be changed inline on the table.

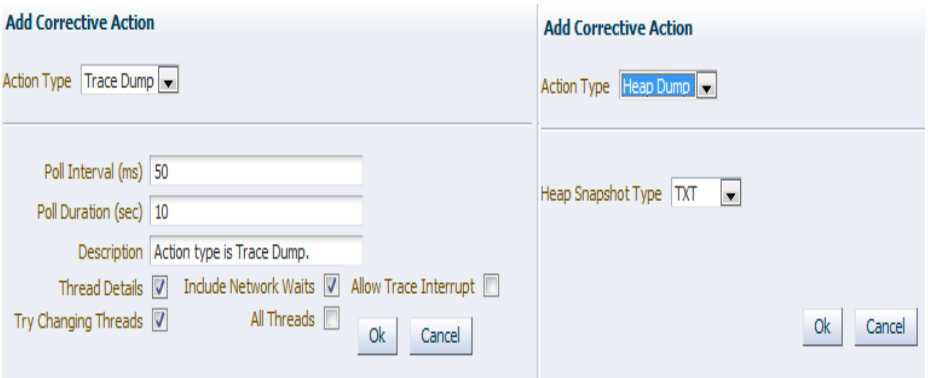


5. Click **Add** to add a corrective action for the corresponding threshold value.

   The Add Corrective Action window appears. Corrective actions are of three types: *No Action*, *TraceDump*, *Heap Dump*.



6. Select the action type you want to add.



   Provide the necessary details for the corresponding action type and click **Ok**.

7. Click **Save** to commit the changes made to the threshold violations.

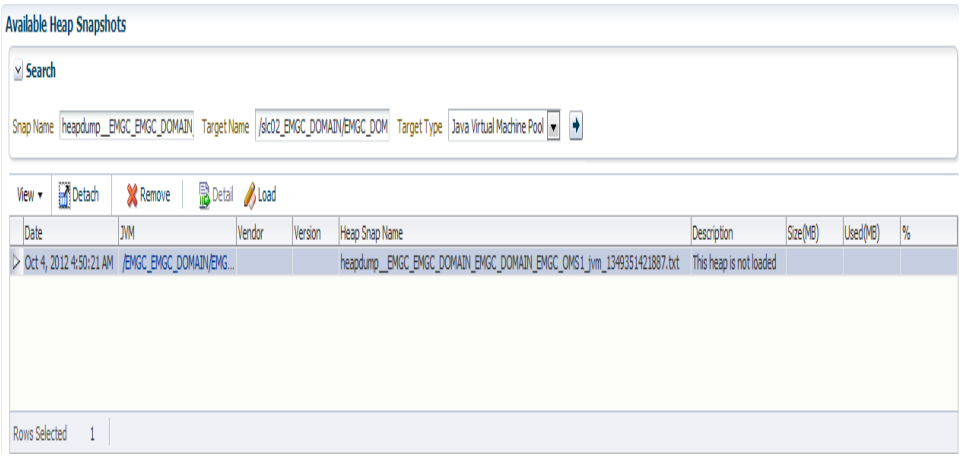
Note: If a corrective action was triggered by the threshold violation, the trace/heap dump files will be displayed under **Event Details**. A link to navigate to the trace/heap snapshot page will also be available.

4.6.2 Loading the Heaps

You can trigger a corrective action to perform the heap analysis. The heap created on the snapshot page is not loaded by default.

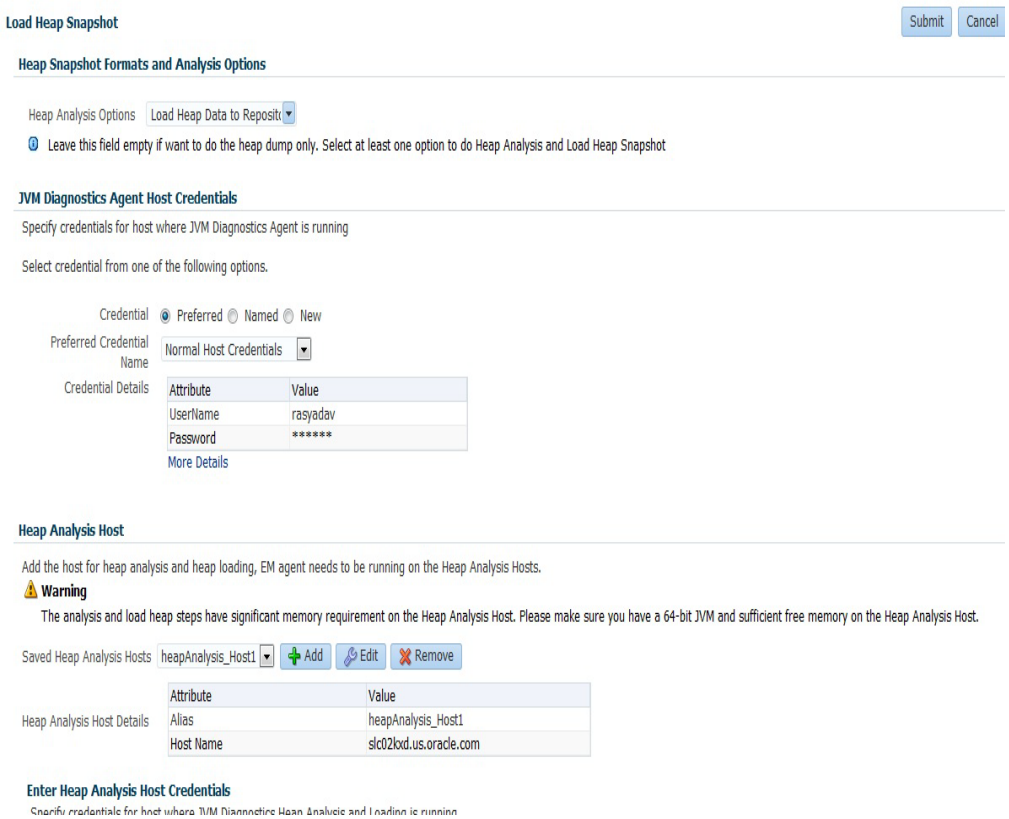
To get started with loading the heaps:

1. Navigate to the Java Virtual Machine Pool Home page.
2. Select **Heap Snapshots and Class Histograms** from the **Java Virtual Machine Pool** menu.
3. Select the heap you want to load to the repository.



4. Click **Load**.

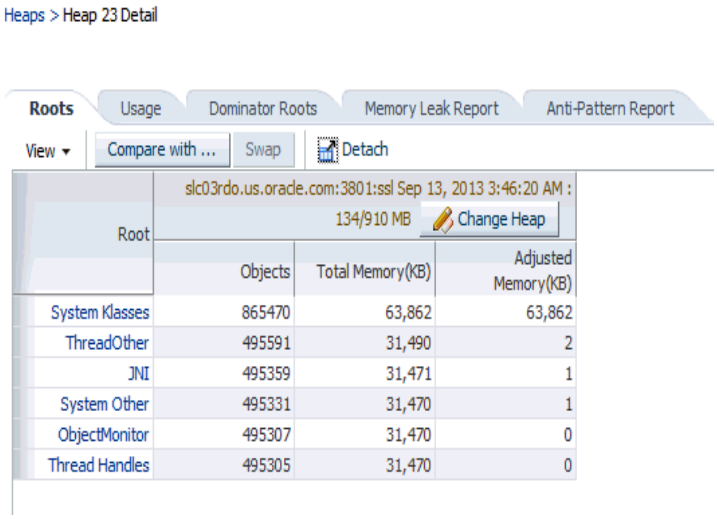
   The Load Heap Snapshot page appears.



5. Configure the following details before you load the heap dump:
   - JVM Diagnostics Agent Host credentials
   - Host and path of the selected Heap loader
   - Credentials for Heap Loader Host and JVM Diagnostics DB User.
6. Click **Load Heap** on the Load Heap Snapshot page.
7. Click **Yes** on the confirmation window to load the selected heap.

   The selected heap is loaded to the repository.
8. Navigate to the Available Heap Snapshots page.
9. Select the loaded heap from the Heap Snapshots table and click **Detail** to analyze the heap dump.

   The heap details are displayed.



Glossary

abstract role

An enterprise role that can be associated with any user, irrespective or his or her job or duties. Typical examples of this role are Employee, Manager, Customer, and Supplier. Several job roles are provisioned with each Oracle Fusion application. An abstract role *must* inherit at least a **duty role**.

Active Session History (ASH)

See **ASH**.

ADDM

Automatic Database Diagnostic Monitor. Analyzes statistics to provide automatic diagnosis of major performance problems.

Administration Server

Part of the **Oracle WebLogic Server domain** and runs the processes that manage Oracle Business Intelligence components. The Administration Server includes the Oracle WebLogic Server Administration Console, Oracle Fusion Middleware Control, and JMX MBeans. For a Simple Install type, the Administration Server also includes Java components for Oracle Business Intelligence such as Oracle BI Publisher and Oracle Real-Time Decisions.

ADR

Automatic Diagnostic Repository. A file-based repository that stores database diagnostic data such as trace files, the alert log, and Health Monitor reports. Its enables customers and Oracle Support to correlate and analyze diagnostic data across multiple Oracle instances, components, and products. ADR is located outside the database, which enables Oracle Database to access and manage ADR when the physical database is unavailable. An instance can create ADR before a database has been created.

adrci

Automatic Diagnostic Repository Command Interpreter. A standard incident administration tool, you can use it for viewing and packaging incidents. It is provided with each instance of the Automatic Diagnostic Repository (ADR) in Oracle Fusion Applications, Oracle Fusion Middleware, and Oracle Database.

Advanced Queuing (AQ)

See **AQ**.

aggregate storage option (ASO)

See **ASO**.

API

Application Programming Interface. A set of exposed data structures and functions that an application can use to invoke services on an application object, such as a portlet.

application programming interface (API)

See **API**.

application role

A configuring role which is a collection of users, enterprise roles, and application roles, and is stored in the domain policy store.

AQ

Advanced Queuing. AQ provides database-integrated message queuing functionality. It leverages the functions of the Oracle database so that messages can be stored persistently, propagated between queues on different machines and databases, and transmitted using Oracle Net Services, HTTP (S), and SMTP.

ASH

Active Session History. Active Session History (ASH) statistics are samples of session activity in the database. The database samples active sessions every second and stores them in a circular buffer in the System Global Area (SGA). Any session that is connected to the database and using CPU, or is waiting for an event that does not belong to the idle wait class, is considered an active session

ASO

Aggregate Storage Option. The option to use Aggregate Storage as opposed to Block Storage.

automatic memory management

A database memory-management mode whereby the database dynamically tunes the sizes of the individual SGA components and the sizes of the individual **Program Global Area (PGA)**.

automatic PGA memory management

A database memory-management mode whereby you set a target size for the instance PGA. The database then tunes the size of the instance **PGA** to your target, and dynamically tunes the sizes of individual PGAs.

automatic shared memory management

A database memory-management mode whereby you set target and maximum sizes for the SGA. Oracle Database then tunes the total size of the SGA to your designated target, and dynamically tunes the sizes of all SGA components. In this memory management mode, you also implicitly enable automatic PGA memory management.

Automatic Database Diagnostic Monitor (ADDM)

See **ADDM**.

Automatic Diagnostic Repository (ADR)

See **ADR**.

Automatic Diagnostic Repository Command Interpreter (adrci)

See adrci.

Automatic Workload Repository (AWR)

See **AWR**.

AWR

Automatic Workload Repository. Automates database statistics gathering by collecting, processing, and maintaining performance statistics for database problem detection and self-tuning purposes.

AWR snapshot

A set of data for a specific time that is used for performance comparisons.

BHD

Bounce Handling Daemon. A daemon that tracks e-mail messages that cannot be delivered, parses the returned e-mail messages, and records the cause of the e-mail bounce.

bounce handling daemon (BHD)

See **BHD**.

BPEL

Business Process Execution Language. An XML-based markup language for composing a set of discrete web services into an end-to-end process flow.

BSO

Block storage option. The option to use block storage as opposed to aggregate storage.

Business Process Execution Language (BPEL)

See **BPEL**.

business role

See **job role**.

CA

Certificate Authority. A trusted third party that issues, renews, and revokes digital certificates. The CA essentially vouches for a entity's identity, and may delegate the verification of an applicant to a Registration Authority (RA). Some well known Certificate Authorities (CAs) include Digital Signature Trust, Thawte, and VeriSign.

Certificate Authority (CA)

See CA.

Click -Through Daemon (CTD)

See CTD.

Common Unix Printing Service (CUPS)

See CUPS.

CRM

Customer relationship management. A portfolio of CRM solutions that addresses customer touch-points and provides functionality to support an organization's business needs.

CSF

Credential store framework. A set of APIs that applications can use to create, read, update, and manage credentials securely. A typical use of the credential store is to store credentials (user name and password) to access some external system, such as a database or an LDAP-base repository.

CSR

Customer Service Representative. Customer service representatives resolve low risk customer issues originating from customer calls. CSRs have limited access to OAAM Admin.

credential store framework (CSF)

See **CSF**.

CTD

Click -Through Daemon. It passes details about the HTTP requests it serviced (tracked URLs, Forward to a Friend and so on) to the Marketing Server through SOAP protocol. The communication between the CTD and the Marketing Server requires you to define a way for the SOAP messages top ass through the inner DMZ firewall.

CUPS

Common Unix Printing Service. A modular printing system for Unix-like computer operating systems which allows a computer to act as a print server.

customer relationship management (CRM)

See **CRM**.

customer service representative (CSR)

See CSR

DAL

Data Access Layer. A layer which provides simplified access to data stored in persistent storage, such as a database.

Data Access Layer (DAL)

See DAL

data role

An enterprise role used exclusively in data security policies. It can inherit job roles, duty roles, and abstract roles. Several data roles are provisioned with each Oracle Fusion application.

data security policy

An Oracle Fusion application policy which includes a condition that identifies a row or a set of rows in a business object, and the privileges the data security grants are for *only* the data that meets the condition. Data security policies are stored in the transactional database.

Database Administrator (DBA)

See **DBA**.

DBA

Database Administrator. A user belonging to the DBA group. By default, members in the DBA group have access to all Oracle Portal product pages, and have the Manage privilege for all pages, page groups, database providers, and administration.

diagnostic administrator job role

A **job role** which can use all diagnostic testing functionality provided for Oracle Fusion applications, including purging test results from the database and canceling test runs started by other users.

diagnostic advanced user job role

A **job role** which can schedule and execute diagnostic test runs, view diagnostic test results, attach test results to application incidents for Oracle Fusion applications, and cancel diagnostic test runs that were started by the current user. In general, this job role is recommended for running Oracle Fusion Applications diagnostic tests, since its added capabilities allow users to work with administrators more flexibly during troubleshooting.

diagnostic regular User job role

A **job role** which can schedule and execute diagnostic test runs, view diagnostic test results for Oracle Fusion applications, and cancel diagnostic test runs that were started by the current user.

diagnostic viewer job role

A **job role** which can view and analyze diagnostic test results for Oracle Fusion applications.

Distinguished Names (DN)

See **DN**.

distributed

The term for any environment that is not a non-distributed environment.

DMS

Dynamic Monitoring Service. Enables Oracle Fusion Middleware components to provide administration tools, such as Oracle Enterprise Manager, with data regarding the component's performance, state and on-going behavior. Fusion Middle components push data to DMS and in turn DMS publishes that data through a range of different components. Specifically, DMS is used by Oracle HTTP Server, Oracle Application Development Framework, WebLogic Diagnostic Framework (WLDF), and JDBC. DMS measures and reports metrics, trace events and system performance and provides a context correlation service for these components.

DN

Distinguished Names. Common entries between the source and destination directories.

duty role

An application role that corresponds with the duties of the job.

Dynamic Monitoring Service (DMS)

See **DMS**.

EAR

Enterprise Archive file. A Java EE archive file that is used in deploying applications on a Java EE application server. WebCenter Portal applications are deployed using both a generic EAR file, which contains the application and the respective runtime customization, and a targeted EAR file, which contains only the application for deployment to the application server. EAR files simplify application deployment by reducing the possibility of errors when moving an application from development to test, and test to production.

ECID

Execution Context Identifier. A global unique identifier and a sequence number of the thread of execution that the originating component participates in. The identifier can be used to correlate messages from several components that may be involved in the same thread of execution.

EDN

Event Delivery Network. A type of queue used for event Propagation. EDN uses Advanced Queueing (AQ) database queues.

EJB

Enterprise Java Beans. It is the server-side component architecture for Java Platform, Enterprise Edition (Java EE). EJB technology enables rapid and simplified development of distributed, transaction, secure and portable applications based on Java Technology.

Email Sending Daemon (ESD)

See **ESD**.

Enterprise Archive file (EAR)

See **EAR**.

enterprise group

The term used for enterprise roles in the Oracle WebLogic Administration Console.

Enterprise Java Beans (EJB)

See **EJB**.

enterprise role

A configuring role which is a collection of users and other enterprise roles. It is stored in the domain identity store.

Enterprise Scheduler Application (ESSAPP)

See **ESSAPP**.

ESD

Email Sending Daemon. Assembles each outbound email message for a campaign using the email template (HTML or text) and the recipient list, and then sends each message to your company's outbound MTAs for delivery.

ESSAPP

Enterprise Scheduler Application. The Oracle Enterprise Scheduler Application manages job requests for the Oracle Fusion Financials product family.

ETL

Extract, Transform, and Load. ETL refers to the methods involved in accessing and manipulating source data and loading it into a data warehouse. The order in which these processes are performed varies. Note that ETT (extraction, transformation, transportation) and ETM (extraction, transformation, move) are sometimes used instead of ETL.

Event Delivery Network (EDN)

See EDN.

exclusive binding mode

A method of binding a work assignment to a request processor in which job requests specialized to the work assignment are processed exclusively by that work assignment when it is active. These job requests are excluded from all other work assignments, including the default work assignment. If the work assignment does not have an active workshift, then the job request can be processed by another work assignment.

Execution Context Identifier (ECID)

See **ECID**.

External roles

The term used for enterprise roles in the Oracle Authorization Policy Manager environment.

extract, transform, and load (ETL)

See **ETL**.

farm

A collection of components managed by Fusion Applications control. It can contain an Oracle WebLogic Server Domain, one Administration Server, one or more Managed Servers, clusters, and the Oracle Fusion Middleware components that are installed, configured, and running in the domain.

FAST

Find, assess, secure, test. A methodology required during data masking.

Find, assess, secure, test (FAST)

See **FAST**.

FND Grant

Foundation Grant. A data security policy, which ties a data role or job role to a specific set of data.

Foundation Grant (FND Grant)

See **FND Grant**.

full-movement scenario

A scenario that may occur when moving Oracle Fusion Applications components from one environment to another. In a full-movement scenario, the target environment does

not exist. First, the source environment is created, configured, customized, and tested. Then, the target environment is created by moving all the components along with their configurations from the source environment.

functional policy

An Oracle Fusion application policy which does *not* include a condition. Instead, a function policy assigns permission to resources or code artifacts (such as task flows, pages, Java methods, or UI components) and grants a specific set of actions on each resource. Functional policies are stored in the domain policy store.

global policy attachment (GPA)

See **GPA**.

global unique identifier (GUID)

See **GUID**.

GPA

Global Policy Attachment. A policy set in which policies are attached globally to a range of endpoints of the same type. By using GPA, you can ensure that all subjects are secured by default.

GUID

Global unique identifier. An identifier generated by the system and inserted into an entry when the entry is added to the directory. In a multimaster replicated environment, the GUID, not the DN, uniquely identifies an entry. The GUID of an entry cannot be modified by a user.

IDE

Integrated Development Environment. A visual application development tool containing editors, debuggers, screen painters, object browsers, and the like. Oracle JDeveloper is an example of an IDE.

initialization parameter file

A text file that can be read by the Oracle instance, but it is not written to by the instance. You can change a text initialization parameter file with a text editor, but changes do not take effect until you restart the Oracle instance. When you start the instance with this type of file, you can still change many initialization parameters dynamically with Database Control, but only for the current instance. Unless you also edit the text initialization parameter file and make the same change, the change is lost when you restart the database instance.

Integrated Development Environment (IDE)

See **IDE**.

Internet Printing Protocol (IPP)

See **IPP**

IPP

Internet Printing Protocol. A standard network protocol for remote printing as well as for managing print jobs, media size, resolution, and so forth.

Java EE

Also known as Java EE 5. Java Enterprise Edition 5 Platform. A platform that enables application developers to develop, deploy, and manage multitier, server-centric, enterprise-level applications. The Java EE platform offers a multi-tiered distributed application model, integrated XML-based data interchange, a unified security model, and flexible transaction control. You can build you own Java EE portlets and expose them through web producers.

Java EE applications

Applications that run in Java Enterprise Edition (Java EE).

Java Message Service (JMS)

See **JMS**.

Java Required Files (JRF)

See **JRF**.

Java Virtual Machine (JVM)

See **JVM**.

JKS keystore

The default JDK implementation of Java keystores.

JMS

Java Message Service. A Java API that allows applications to create, send, receive, and read messages using reliable, asynchronous, loosely coupled communication.

job role

An enterprise role that corresponds with a job or business occupation. A Job role *must* inherit at least a Duty role.

job sets

A collection of job requests.

JRF

Java Required Files. The JRF include the Oracle ADF libraries, several other components shared by the product suites included in a given installation, and components such as Oracle Metadata Service (MDS) and Oracle Platform Security Services (OPSS).

JVM

Java Virtual Machine. A virtual machine that can execute Java bytecode.

LBR

Load-balancing router. A very fast network device that distributes Web requests to a large number of servers. It provides portal users with a single published address, without their having to send each request to a specific middle tier server.

LDAP

Lightweight Directory Access Protocol. A standard, extensible directory access protocol. It is a common language that LDAP clients and servers use to communicate. The framework of design conventions supporting industry-standard directory products, such as the Oracle Internet Directory.

LDAP Data Interchange Format (LDIF)

See **LDIF**.

LDIF

LDAP Data Interchange Format. The set of standards for formatting an input file for any of the LDAP command-line utilities.

Lightweight Directory Access Protocol (LDAP)

See **LDAP**

load-balancing router (LBR)

See **LBR**.

Local Policy Attachment (LPA)

See **LPA**.

LPA

Local Policy Attachment. The act of directly attaching one or more policies to a policy subject.

Managed Server

In a production environment, a Managed Server hosts applications and the resources needed by those applications. A domain, which is a logically related group of Oracle WebLogic Server resources, can have any number of managed Servers. An Administration Server manages these servers.

manual shared memory management

A database memory-management mode whereby you set the sizes of several individual **System Global Area (SGA)** components, thereby determining the overall SGA size. You then manually tune these individual SGA components on an ongoing basis. Manual shared memory management mode is intended for experienced DBAs only. Note that in this mode, automatic PGA memory management remains enabled.

move plan

A single XML file, derived from multiple configuration archives, that provides the properties required to copy content from one installation to another.

NAT

Network Address Translation. The process of modifying IP address information in IP packet headers while in transit across a traffic routing device.

Network Address Translation (NAT)

See **NAT**.

Network File System (NFS)

See **NFS**.

NFS

Network File System. A distributed file system protocol which allows you to access files over a network in a manner similar to how local storage is accessed.

Node Manager

An Oracle WebLogic Server utility that enables you to start, shut down, and restart **Administration Server** and **Managed Server** instances from a remote location. Although Node Manager is optional, it is recommended if your Oracle WebLogic Server environment hosts applications with high availability requirements.

non-distributed

The term for an environment in which the Administration Servers of all the domains and all the system component instances run on the same host.

non-XA transaction

A non-XA transaction always involves just one resource and generally cannot participate in a global transaction.

ODL

Oracle Diagnostic Logging. A method of logging which attempts to harmonize the different logging standards that are used by Oracle products.

ODV

Oracle Database Vault. ODV establishes limitations on the power of privileged users to access sensitive data through segregation of duties policies on DBA roles and by securely consolidating application data in the database. These limitations prevent DBAs and other privileged users from overriding the protections placed on sensitive data by the Virtual Private Database (VPD). Oracle Fusion Applications deploys with the ODV enabled when it is installed.

offline backup

A mode of backup for your Oracle Fusion Applications Environment in which you must first shut down the environment. The Administration Server, all Managed Servers in the domain, and all system components in the Oracle instances should be shut down.

OLTP

Online Transaction Processing. OLTP systems are one of the most common data processing systems in today's enterprises. They are primarily characterized through a specific data usage that is different from data warehousing environments, yet some of the characters, such as having large volumes of data and lifecycle-related data usage and importance, are identical. Main characteristics of OLTP are short response time, small transactions, data maintenance operations, large user populations, high concurrency, large data volumes, high availability, and lifecycle related data usage.

online backup

A mode of backup for your Oracle Fusion Applications environment in which you do not shut down the environment before backing up the files. To avoid an inconsistent backup, do not make any configuration changes until the backup is completed. To ensure that no changes are made in the Oracle WebLogic Server Domain, lock the WebLogic Server configuration.

Online Transaction Processing (OLTP)

See **OLTP**.

OPMN

Oracle Process Manager and Notification Server. A process management tool that manages all system components (server processes), and supports both local and distributed process management, automatic process recycling and the communication of process state (up, down, starting, stopping). OPMN detects process unavailability and automatically restarts processes).

Oracle Database Vault (ODV)

See **ODV**.

Oracle Diagnostic Logging (ODL)

See **ODL**.

Oracle Process Manager and Notification Server (OPMN)

See **OPMN**.

Oracle Real Application Clusters (RAC)

See **RAC**.

Oracle WebLogic Server domain

The basic administration unit for WebLogic Server instances. A domain consists of one or more WebLogic Server instances (and their associated resources) that you manage with a single **Administration Server**. You can define multiple domains based on different system administrators' responsibilities, application boundaries, or geographical locations of servers. Conversely, you can use a single domain to centralize all WebLogic Server administration activities.

Oracle WebLogic Scripting Tool (WLST)

See **WLST**.

PEM

Privacy-enhanced Electronic Mail. An encryption technique that provides encryption, authentication, message integrity, and key management.

personally Identifiable Information (PII)

See **PII**

PGA

Program Global Area. A memory buffer that contains data and control information for a server process. A PGA is created when a server process is started.

PII

Personally identifiable information. Information that can be used to uniquely identify, contact, or locate a single person or can be used with other sources to uniquely identify an individual.

post-processing function (PPF)

See **PPF**.

PPF

Post-processing function. This is a special case of a user-defined function. A post processing function is called after the mask value is generated using the specified

format. The function takes the generated mask value and further modifies it to produce the actual mask value.

Privacy-enhanced Electronic Mail (PEM)

See **PEM**.

privilege

Entitlement (or permission set).

product family

Combinations of one or more Java EE applications specific to Oracle Fusion Applications.

product offering

Groups of features within an installation of Oracle Fusion Applications which represent the highest-level collection of functionality that you can license and implement.

Program Global Area (PGA)

See **PGA**.

provisioning

The entire set of operations required to install, configure, and deploy applications product offerings from a system point of view. It performs these operations:

- Installation provides the operations related to laying down all the component needed to create an Oracle Fusion Applications environment.
- Configuration tailors components based on the applications topology, the creation of Oracle WebLogic Server Managed Servers and clusters, and the updating of endpoints and virtual hosts.
- Deployment starts the Managed Servers and clusters and facilitates the actual use of product offerings.

purge policies

Allow the scheduling service to remove scheduled jobs according to specified criteria. For example, a purge policy might specify the retention of all Java type job requests using a particular job definition submitted executed by a given application for three days. Another purge policy might retain a particular type of job request, say, all SQL job requests in a successful state, for only one day. You can also specify the frequency at which the purge policy is to run.

query optimizer

Determines the best and most efficient way to retrieve the results of an SQL statement. It compares the cost of all possible approaches and chooses the approach with the least cost. The query optimizer is responsible for generating an execution plan.

RAC

Real Application Clusters. Oracle RAC allows the Oracle Database to run any packaged or custom application across a set of clustered servers. This capability provides the highest levels of availability and the most flexible scalability. If a clustered server fails, Oracle Database continues running on the surviving servers. When more processing power is needed, you can add another server without interrupting access to data.

RCU

Repository Creation Utility. Oracle Fusion Applications RCU is a utility that loads applications and middleware content into the database. The process of running Applications RCU creates the applications and middleware schemas, loads seed data, and creates the tablespaces, as well as other required packages.

RDA

Remote Diagnostic Agent. Provides a suite of data collection and diagnostic scripts that aid in the analysis and support of Oracle products.

Recovery Manager (RMAN)

See **RMAN**.

Remote Diagnostic Agent (RDA)

See **RDA**.

Repository Creation Utility (RCU)

See **RCU**.

RMAN

Recovery Manager. You can use Oracle Recovery Manager to recover database-based metadata repositories and Oracle Fusion Applications Databases.

SAML

Security Assertion Markup Language. Defines a framework for exchanging security information between online business partners.

SBC

Session Border Controller. A piece of equipment that is deployed at the border of the network and controls the signaling and media across network address translation (NAT) devices and firewall boundaries. In addition, it prevents unauthorized access and denial of service (DoS) attacks.

Security Assertion Markup Language (SAML)

See **SAML**.

Secure Socket Layer (SSL)

See **SSL**.

server parameter file

A binary file that can be written to and read by the database. It *must not* be edited manually. It is stored on the host system on which Oracle Database is running. Changes are made when you use Database Control to modify one or more initialization parameters, or when Oracle Database itself makes changes for self-tuning purposes. Any changes to it persist across database shutdown and startup operations.

Service-Oriented Architecture (SOA)

See **SOA**.

Session Border Controller (SBC)

See **SBC**.

SGA

System Global Area. A group of shared memory structures that contains data and control information for one Oracle Database instance. If multiple users are concurrently connected to the same instance, the data in the instance SGA is shared among the users. Consequently, the SGA is sometimes referred to as the shared global area. The combination of the background processes and memory buffers is called an Oracle instance.

Simple Object Access Protocol (SOAP)

See **SOAP**.

snapshot (in Content Movement)

An image of a Fusion Applications instance at a moment in time, which can be used as the source for a content movement.Used in the Enterprise Manager Cloud Control interface.

SOA

Service-Oriented Architecture. A design methodology aimed at maximizing the reuse of application services.

SOAP

Simple Object Access Protocol. A protocol for sending and receiving requests and responses across the Internet. It is based on XML and simple transport protocols, so it is not blocked by firewalls. SOAP is independent of operating system, implementation language, or any single object model.

SSL

Secure Socket Layer. An industry standard protocol designed by Netscape Communications Corporation for securing network connections. SSL provides authentication, encryption, and data integrity using public key infrastructure (PKI).

standard binding mode

The default mode of binding a work assignment to a request processor. Standard binding mode allows the request processor to process job requests as defined by the specialization rules when an active workshift is defined. If a job request is specialized to two different work assignments, either of those work assignments or the default work assignment can process the job request.

System Global Area (SGA)

See **SGA**.

task role

See **duty role**.

TDE

Transparent Data Encryption. TDE prevents access to PII in the file system or on backups or disk. It also protects confidential data, such as credit card and social security numbers. TDE also encrypts sensitive table data stored in data files at the tablespace level.

Thread ID (TID)

See **TID**

TID

Thread ID. A unique identifier for the thread within the java process where the message was generated.

Transparent Data Encryption (TDE)

See **TDE**.

UDF

User-Defined Function. Takes the original value, the row id, and column name to generate the mask value. A single column format can be a combination of one or more formats including UDF.

User-Defined Function (UDF)

See **UDF**.

Virtual Private Database (VPD)

See **VPD**.

VPD

Virtual Private Database. A feature for ASPs that want to leverage the Oracle database to host their customers. Essentially, it uses one physical database instance for all customers, but to each customer it looks as if they have their own database. Users cannot see any information that is not meant for them and complete customer isolation is achieved. It requires little to no changes in the core application to take effect as most of the work is done at the database level. Implementing VPD basically requires two key steps: adding a context column (for example, company name) to all the database tables, and implementing a policy to restrict queries on each table based on the context of the logged in user. VPD provides highly secure, full subscriber isolation using this method.

WLST

Oracle WebLogic Scripting Tool. A command line tool for managing Oracle Fusion Middleware components, such as Oracle WebCenter.

work allocation

The process of defining constraints on where and when jobs run as well as the amount of resources that can be used to process the jobs. This process includes creating a work assignment, and the binding the work assignment to a request processor.

work assignments

Rules that govern the execution and binding of jobs to a particular server and request processor.

workshift

Temporal window of time during which jobs can processed and what resources can be used during those periods. The resources defined in a workshift include threads, which are a local resource of the request processor, and asynchronous workers, a global resource. The number of asynchronous workers can be specified to throttle the use of a shared global resource, such as database jobs.

XA transaction

An XA transaction is a global transaction that may span multiple resources.