

# Troubleshooting and Performance Tuning of Oracle SOA Suite 12c

# Objectives

After completing this lesson, you should be able to:

- Design a tuning methodology
- Identify top areas that require tuning
- Monitor and tune Oracle SOA Suite by collecting performance information and by using diagnostic tools
- Configure selective tracing
- Troubleshoot common problems:
  - Managed Server startup problems
  - Message failure caused by too many open files
  - Connection timeouts
  - Application-related problems



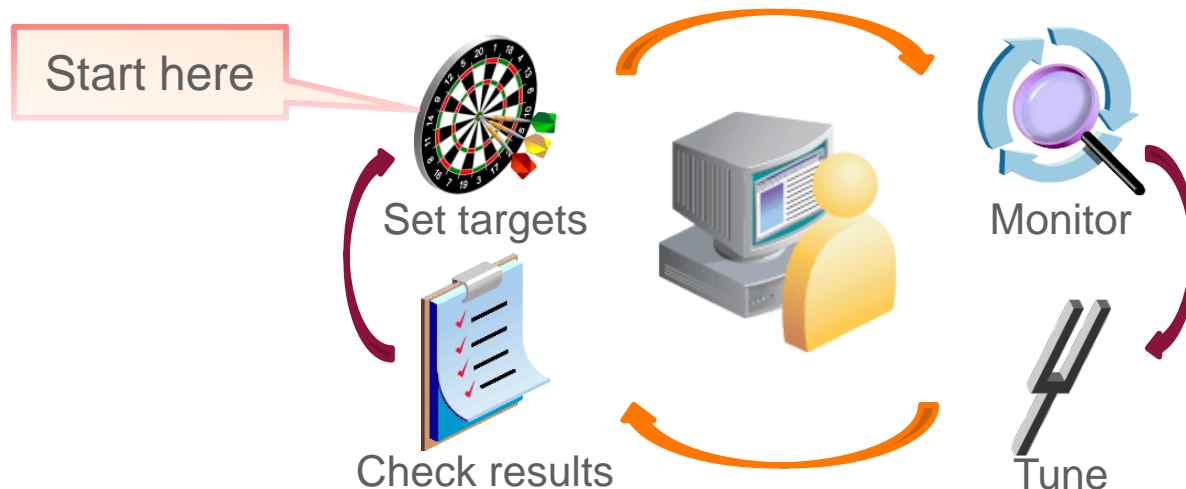
# Agenda

- Tuning methodology: Overview
- Tools for monitoring Oracle Fusion Middleware environments
- Monitoring and tuning with Oracle SOA Suite
- Troubleshooting common problems

# Performance Tuning: Overview

Performance tuning is an iterative process, where you:

- Set performance goals as measurable objectives
- Monitor environment (best done during peak periods) to obtain a baseline
- Tune based on data collected from monitoring tasks and tools
- Check results, and repeat setting goals, monitoring, tuning, and check cycle as needed



# Top Performance Tuning Areas

- Hardware resources
- Operating system
- Java Virtual Machine (JVM)
- Database
- WebLogic Server
- Database connections and data source statement caching
- Concurrency of Fusion Middleware components
- Logging levels

**Note:** Tune the lowest level first, such as the hardware resources, then the operating system, and then additional layers as required. Always tune the foundation layers first.



# Agenda

- Tuning methodology: Overview
- Tools for monitoring Oracle Fusion Middleware environments
- Monitoring and tuning with Oracle SOA Suite
- Troubleshooting common problems

# Monitoring and Diagnostic Tools: Overview

Oracle Fusion Middleware management tools include:

- Oracle Enterprise Manager Fusion Middleware Control
- Oracle WebLogic Server Administration Console
- Oracle Fusion Middleware Diagnostic Framework, which integrates with the WebLogic Diagnostic Framework (WLDF)
- WebLogic Scripting Tool (WLST)
- DMS Spy Servlet for access to the Dynamic Monitoring Service (DMS)

# Monitoring with Oracle Enterprise Manager Fusion Middleware Control

Oracle Enterprise Manager Fusion Middleware Control provides the capability to monitor:

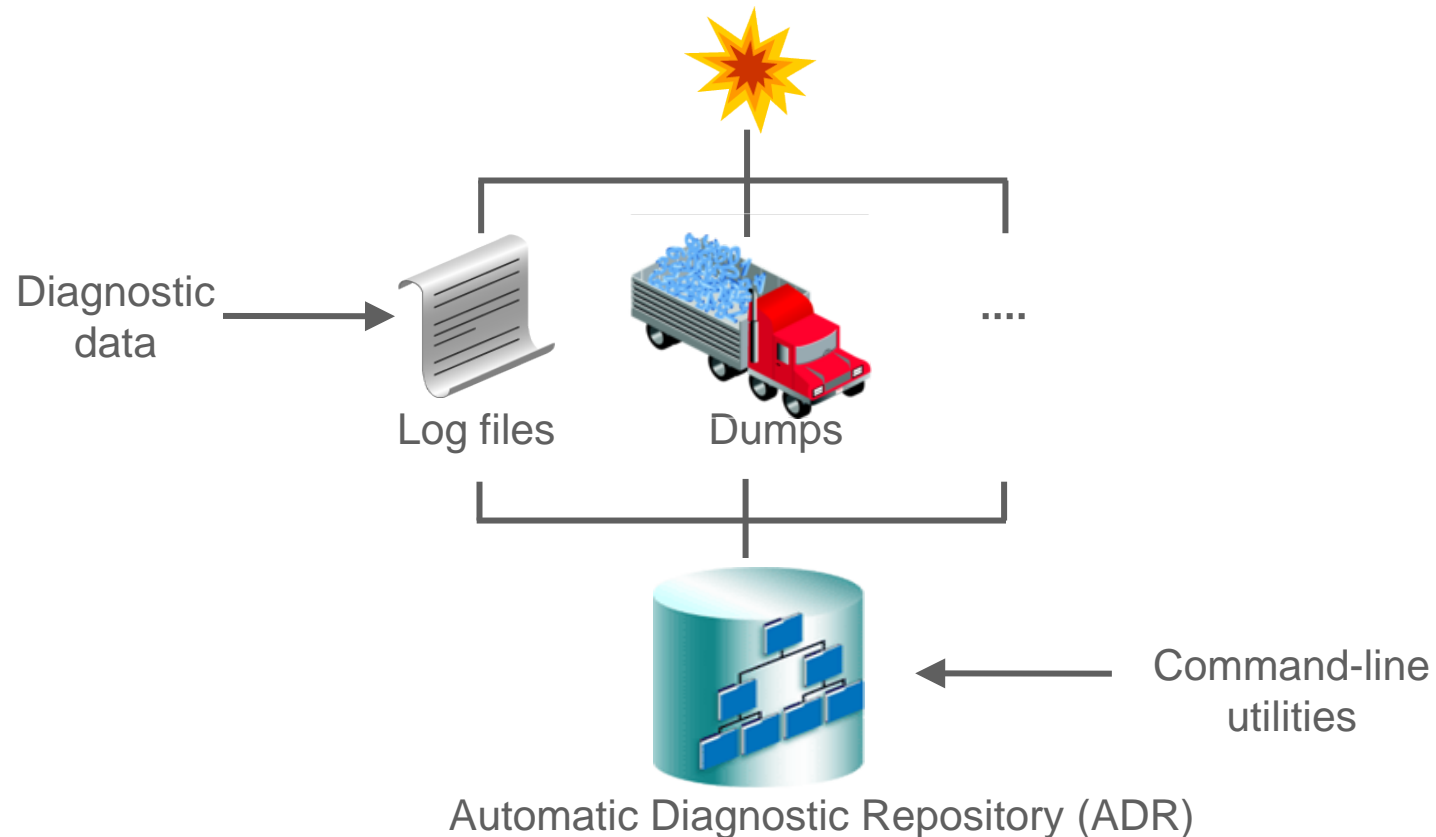
- The state and performance of each element of the farm by providing default performance metrics
- CPU usage, Work Manager, JMS servers, and JDBC and JTA usage for Oracle WebLogic Server
- JVM performance in terms of heap versus non-heap usage, garbage collection, and threads performance
- Deployed composite applications and web services
- Servlets, JSPs, and EJBs by using a wide range of application metrics , as well as faults, invocations, and violations by using web services metrics

**Note:** The performance summary page can be customized to help administrators monitor performance and diagnose problems.



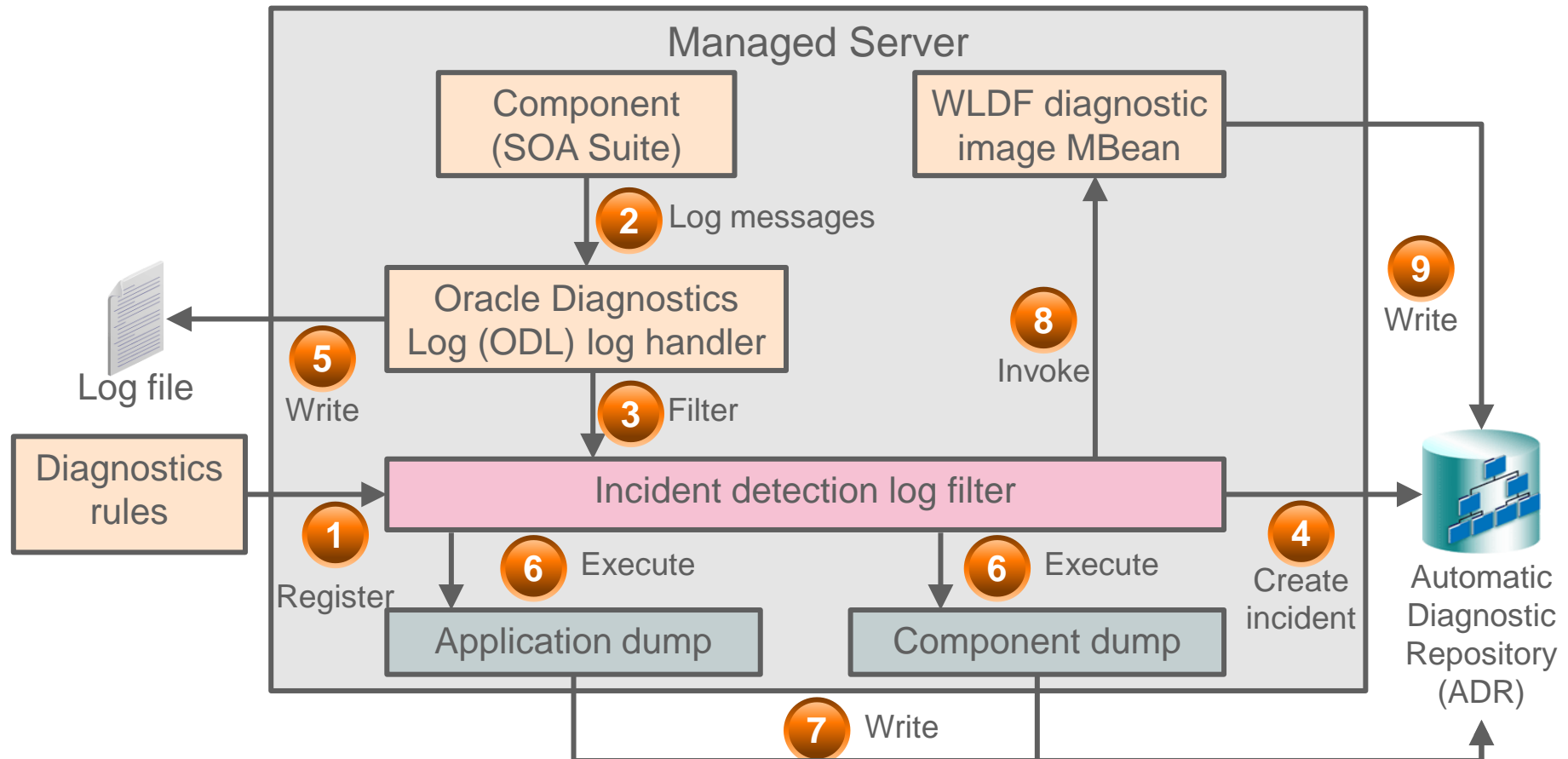
# Oracle Fusion Middleware Diagnostic Framework

Oracle Fusion Middleware Diagnostic Framework aids in detecting, diagnosing, and resolving targeted problems, and integrates with the WebLogic Diagnostic Framework (WLDF).



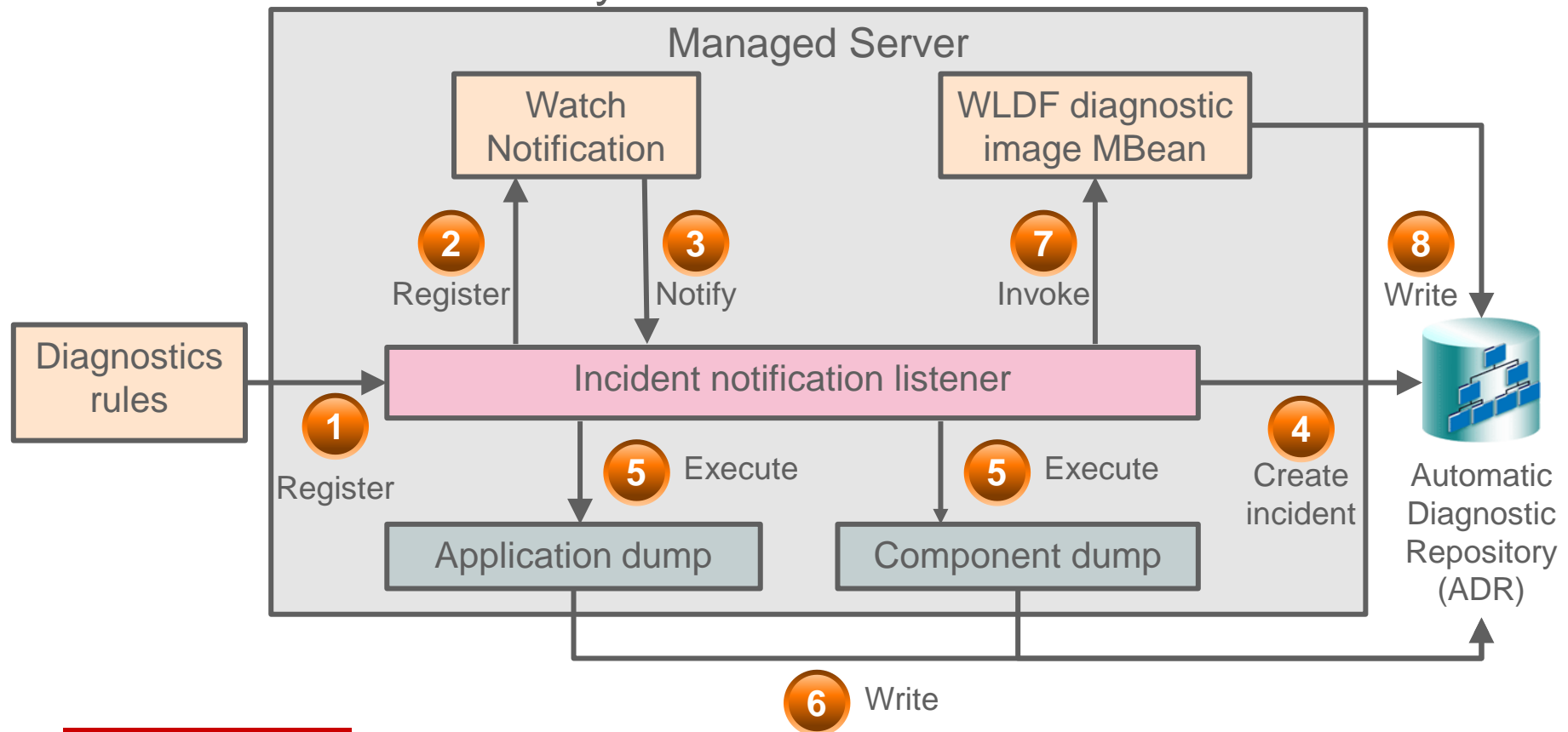
# Oracle Dynamic Monitoring Service and Architecture

The Oracle Dynamic Monitoring Service (DMS) provides data about component performance, state, and ongoing behavior.



# WLDF Watch and Notification Component

Oracle Fusion Middleware Diagnostic Framework registers a notification listener, which listens for events from the WLDF Watch and Notification system.



# Configuring Diagnostic Framework Settings

The Diagnostic Framework MBean, called DiagnosticConfig, is used to enable or disable:

- Detection of incidents through the log files
- Flood control and its parameter settings

The screenshot illustrates the navigation path to the DiagnosticConfig MBean in the Oracle WebLogic Server Administration Console. It is divided into three main sections:

- Target Navigation:** Shows the hierarchy: Application Deployments > SOA > WebLogic Domain > edg\_domain > AdminServer.
- edg\_domain Overview:** Shows the 'WebLogic Domain' selected, with the 'System MBean Browser' link highlighted in the left sidebar.
- System MBean Browser:** Displays the tree structure of MBeans. The path to the target MBean is highlighted with red boxes: Configuration MBeans > Runtime MBeans > Application Defined MBeans > oracle.dfw > Domain: edg\_domain > oracle.dfw.jmx.DiagnosticsConfig > DiagnosticConfig.

On the right, the 'Application Defined MBeans: oracle.dfw.jmx.DiagnosticsConfigMBean:DiagnosticConfig' page is shown, featuring an 'Information' tab and a table of attributes.

Name	Access	Value
1 ConfigMBean	R	true
2 DumpSamplingEnabled	R	true
3 DumpSamplingIdleWhenHealthy	RW	true
4 DumpSamplingMinimumHealthyPeriod	RW	259200000
5 eventProvider	R	true
6 eventTypes	R	jmx.attribute.ch
7 FloodControlEnabled	RW	true
8 FloodControlIncidentCount	RW	5

# Viewing DMS Metrics

DMS metrics can be viewed with:

- Oracle Enterprise Manager Fusion Middleware Control
- The Spy Servlet
- Other tools

# Viewing Metrics with Oracle Enterprise Manager Fusion Middleware Control

Fusion Middleware Control provides:

- A Routing Topology Viewer for the domain and related metrics
- Performance monitoring and metrics for different components

The image illustrates the navigation steps to view performance metrics in Oracle Enterprise Manager Fusion Middleware Control:

- a1**: In the **Target Navigation** pane, select **edg\_domain**.
- b1**: In the **edg\_domain** menu, select **Routing Topology**.
- a2**: In the **edg\_domain** menu, select **Monitoring**.
- b2**: In the **Monitoring** sub-menu, select **Performance Summary**.
- b3**: In the **Performance Summary** page, click **Show Metric Palette**.

The **Routing Topology** view shows the domain structure, including **AdminServer (9)**, **Application Deployments (7)**, **Metadata Repositories (2)**, **soa\_cluster1 (66)**, **soa\_server1 (32)**, and **soa\_server2 (32)**.

The **Performance Summary** page displays metrics for **AdminServer** over the **Past 15 minutes**. The **General** tab shows **CPU Usage (%)** and **Active Sessions**.

# Viewing Metrics with the Spy Servlet

admin.example.com:8080/dms/Spy

## Login to DMS Spy

User Name:

weblogic

Password:

.....

Login

## Metric Tables

[Logout](#)

[DMS Metrics](#)

[WebLogic Metrics](#)

[Aggregated Metrics](#)

**DMS Metrics**

[ADF](#)

[JFW\\_Dump](#)

[DFW\\_Incident](#)

## Metric Tables

[DMS Metrics](#)

[WebLogic Metrics](#)

[Aggregated Metrics](#)

**DMS Metrics**

[ADF](#)

[JFW\\_Dump](#)

[DFW\\_Incident](#)

[JDBC\\_Connection](#)

[JDBC\\_DataSource](#)

## JDBC\_Connection

Name	Host	Process	CreateNewStatement	CreateStatement	DBWaitTime	JDBC_Connection_Url	JDBC_Connection_Username
CONNECTION_10248	host01.example.com	soa_server1:8001	avg, msec	0.071	avg, msec	0	jdbc:oracle:thin:@//host02.example.com:1521/ORCL
			completed, ops	14	completed, ops	0	
			maxActive, threads	1	maxActive, threads	0	
			maxTime, msec	1	maxTime, msec	0	
			minTime, msec	0	minTime, msec	0	
			time, msec	1	time, msec	0	
						176314447	

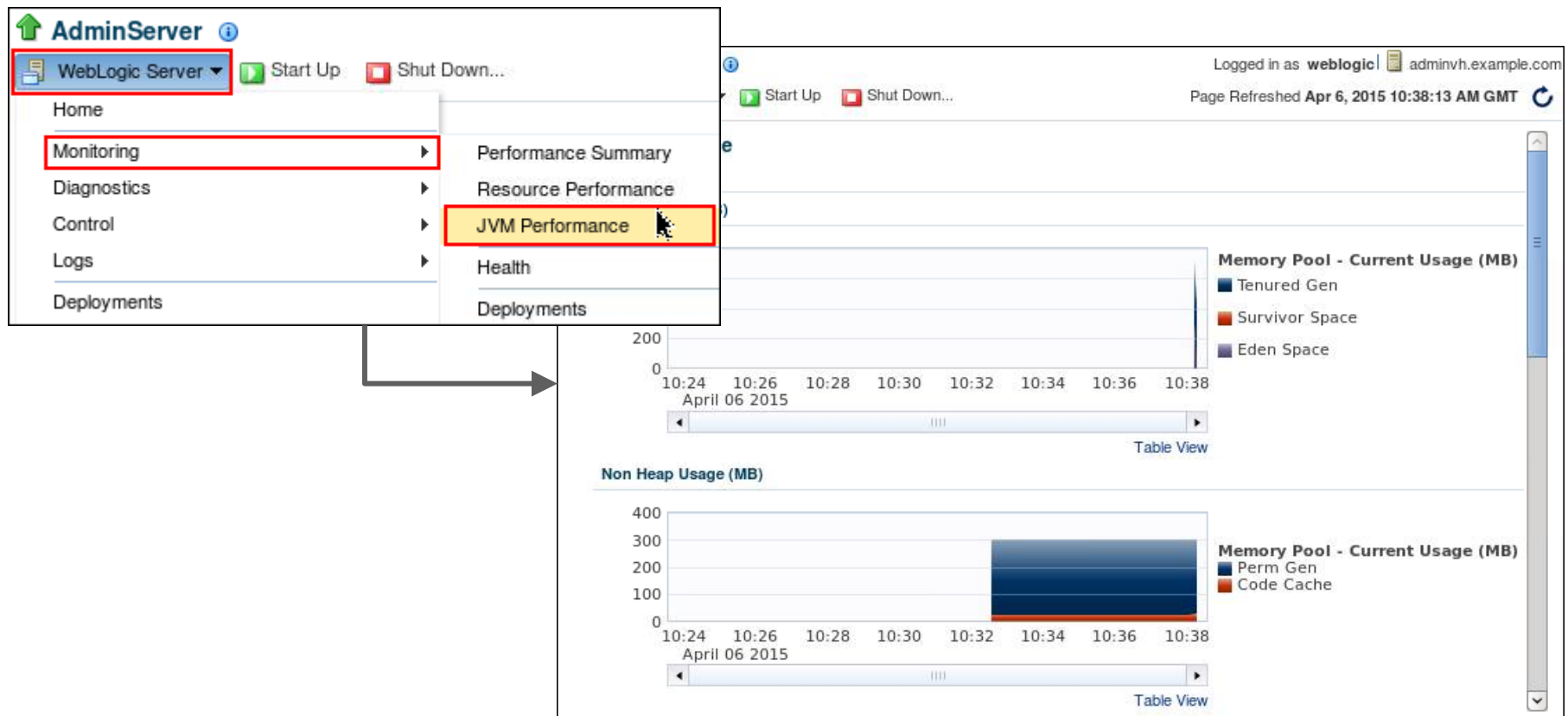
# Agenda

- Tuning methodology: Overview
- Tools for monitoring Oracle Fusion Middleware environments
- **Monitoring and tuning with Oracle SOA Suite**
- Troubleshooting common problems



# Monitoring the Java Virtual Machine Performance

To monitor Java Virtual Machine (JVM) performance, select one of the server instances (AdminServer or Managed Servers), and click WebLogic Server > Monitoring > JVM Performance.



# Tuning a Java Virtual Machine Memory

- Tuning heap size to minimize JVM garbage collection, while maximizing the number of SOA clients at a given time:
    - Ensuring that the sum of (maximum heap size multiplied by number of JVMs) is less than the physical memory (RAM)
    - Increasing the minimum size of the dynamic heap (the `-Xms` option), if out of heap memory (not due to memory leakage)
    - Decreasing maximum size of the memory pool (the `-Xmx` option), if out of native memory
  - Selecting an appropriate garbage collection scheme
- Tip:** A general rule is to set minimum heap size (`-Xms`) equal to the maximum memory pool size (`-Xmx`) to minimize garbage collection processing.



# Monitoring and Tuning the Database Size

Monitoring database performance is a task for the database administrator, who can perform some of the following tasks:

- Collecting schema statistics for MDS schemas
- Increasing redo log size
- Reclaiming disk space, such as purging data in the SOAINFRA and MDS schemas

**Note:** Use automatic purging configuration or supplied PL/SQL scripts for purging SOA instance data.

# Tuning the Oracle Database

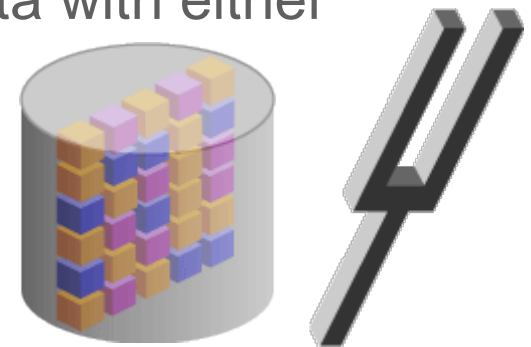
- Tuning the database initialization parameters in the `init.ora` file:

- Processes

```
SQL> SELECT value FROM v$parameter WHERE name = 'processes';  
SQL> ALTER SYSTEM SET processes=300 SCOPE=SPFILE;
```

- Buffer pool size

- Using Automatic Segment Space Management (ASSM) for permanent tablespaces
- Extending tablespaces early to avoid problems at run time
- Managing database growth by purging data with either of the following:
  - Auto purging in Oracle SOA Suite 12c
  - Manual purge scripts that run PL/SQL procedures loaded into the `SOAINFRA` database schema



# Extending Tablespaces to Avoid Problems at Run Time

If a tablespace is not extended when it reaches its size limit, runtime processing will be impacted.

Solutions:

- Enable autoextend for the tablespace.
- Manually increase the tablespace size if you receive an alert.



# Configuring Database Connections

Consider the following data source configurations when performance is an issue:

- Ensure that the connection pool has enough free connections.
- Statement caching can eliminate potential performance impacts.
- Disable unnecessary connection testing and profiling.

# Developing a Strategy for Managing Database Size

Three main strategies for reducing database schema size are:

- Purging SOA Suite instance data regularly:
  - Automatically by using the auto purge functionality
  - Manually by running PL/SQL purge procedures from SQL\*Plus
- Partitioning database tables (automatically enabled when you extend the domain with SOA Suite and select the LARGE option for database size)

# Automatically Purging Instance Data

From the SOA folder in the Navigator, right-click soa-infra and select SOA Administration > Auto Purge.

The screenshot shows the 'Auto Purge' configuration page in the SOA Administration console. The page is titled 'Auto Purge' and includes a description: 'Purge jobs must be run regularly to free up the database from older data associated with flow instances, adapter reports, and fault alerts. It is **strongly recommended** to enable Auto Purge to ensure an optimal performance of your runtime environment.' The 'Auto Purge Job' is set to 'delete\_instances\_auto\_job1', which is 'Enabled' (checked). There are 'Apply' and 'Revert' buttons. The 'Schedule' section prompts the user to 'Enter a valid calendaring expression' and shows the job schedule as 'freq=daily; byhour=0; byminute=0; bysecond=0'. The 'Configuration' section shows the 'Purge Type' as 'SINGLE', 'Retain Data' as '7 days' (with a note that data within this interval is NOT purged), and 'Maximum Flows to Purge' as '1000000'. A link 'More Auto Purge Configuration Properties...' is at the bottom.

soa-infra Logged in as **weblogic**

SOA Infrastructure

Page Refreshed **Feb 24, 2014 8:24:04 PM PST**

## Auto Purge

Purge jobs must be run regularly to free up the database from older data associated with flow instances, adapter reports, and fault alerts. **It is strongly recommended** to enable Auto Purge to ensure an optimal performance of your runtime environment.

**Auto Purge Job** delete\_instances\_auto\_job1 Enabled ☒

Apply Revert

### Schedule

Enter a valid calendaring expression.

\* Job Schedule

### Configuration

\* Purge Type SINGLE

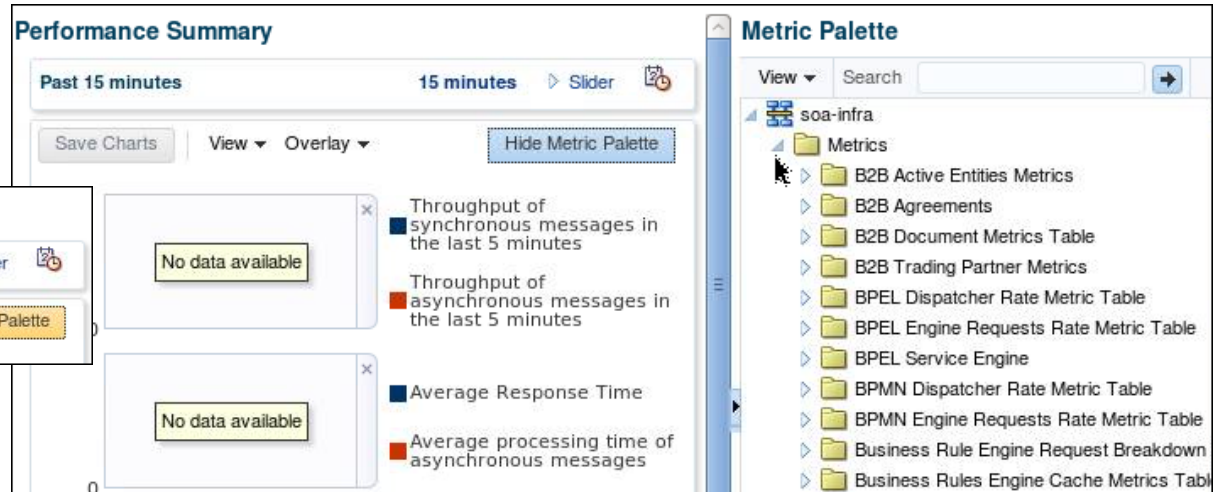
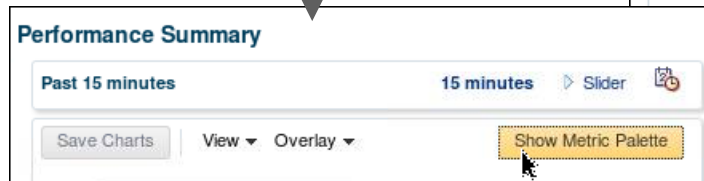
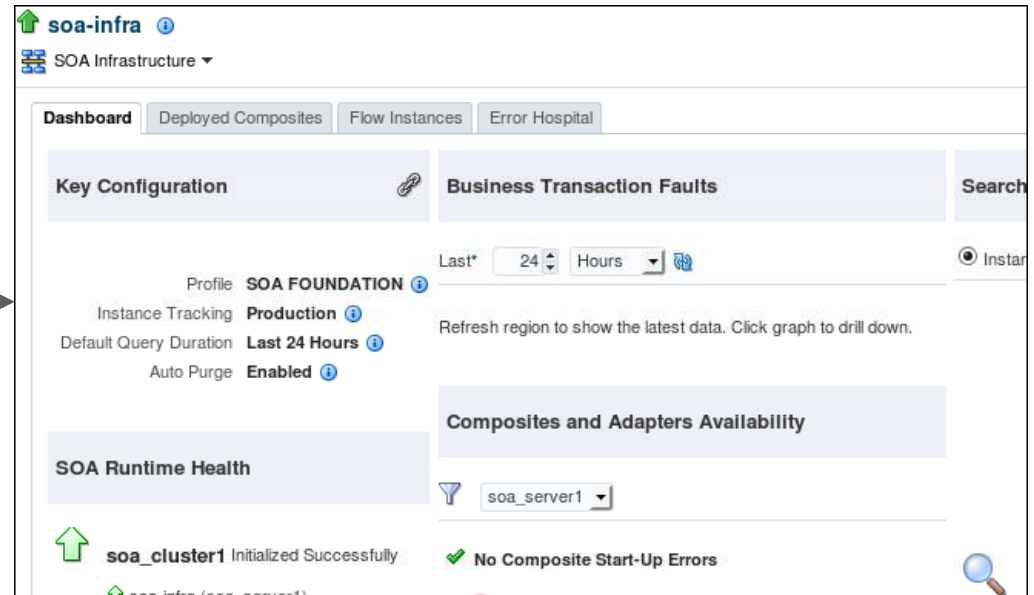
Retain Data  days  
Data within this interval is NOT purged when the job runs

Maximum Flows to Purge

[More Auto Purge Configuration Properties...](#)



# Monitoring the SOA Infrastructure Performance and Metrics Data



# Tuning SOA Infrastructure

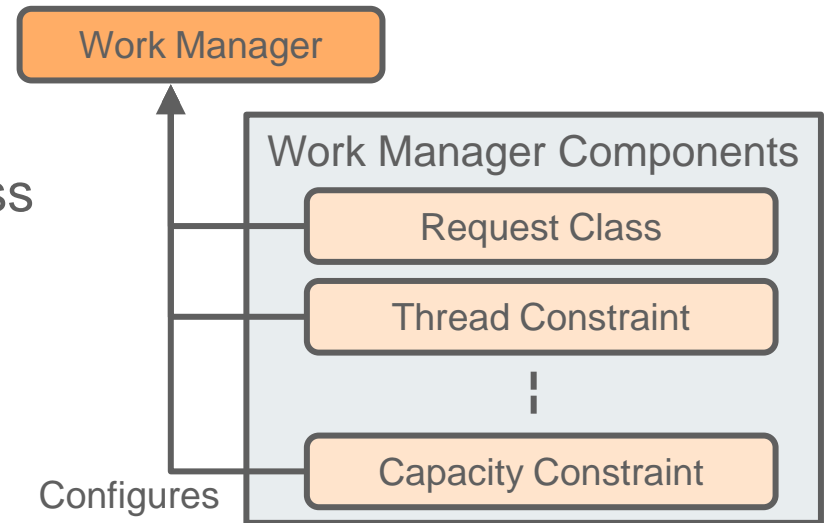
## SOA Infrastructure:

- Is a Java EE-compliant application that manages composites and their life cycle, service engines, and binding components
- Provides essential tuning parameters for:
  - Audit Level (default value: Production), which is managed from the SOA Administration > Common Properties settings page
  - Audit Purge Policy (default: everyday midnight and purges records older than 7 days), which is managed from the SOA Administration > Auto Purge properties page
- Uses Work Managers to manage most SOA-related work threads
- Manages the Modularity Profile via SOA Administration > Common Properties page



# Work Managers and Work Manager Groups

- A Work Manager is an Oracle WebLogic Server entity that represents a logical thread pool.
- A Work Manager assigns priorities for the work to be processed by defining one or more of the following components:
  - Fair Share Request Class
  - Response Time Request Class
  - Min Threads Constraint
  - Max Threads Constraint
  - Capacity Constraint
  - Context Request Class
- A Work Manager Group consists of Work Managers dedicated to processing Oracle SOA Suite work for a specific partition.

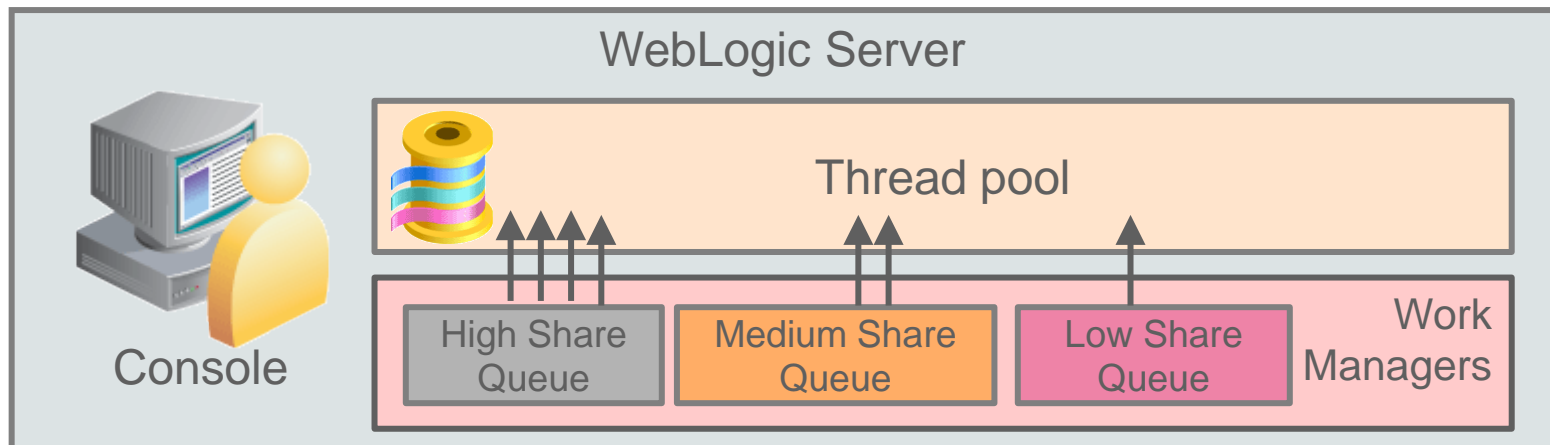


# Work Managers Types

Three types of Work Managers are provided that are characterized by its scope, its definition, and how it is used:

- Default Work Manager
- Global Work Manager
- Application-scoped Work Manager

**Note:** Configuration of Work Managers can be done through the WebLogic Server Administration Console.



# Managing Work Manager Component Configuration

Configure Work Manager components in the Oracle WebLogic Administration Console.

The screenshot illustrates the navigation path to configure Work Manager components in the Oracle WebLogic Administration Console. It shows the Domain Structure tree on the left, the Summary of Work Managers page, and the configuration page for a specific Work Manager.

**Domain Structure**

- edg\_domain
  - Environment
    - Servers
    - Clusters
    - Coherence Clusters
    - Machines
    - Virtual Hosts
    - Work Managers**
    - Startup and Shutdown Classes

**Summary of Work Managers**

Global Work Managers, Request Classes and Constraints

New Clone Delete

Name	Type
analytics_maxEventsToBuffer	Capacity Constraint
default_Adapters	Work Manager
default_Analytics	Work Manager
default_bpmnEngine	Work Manager
default_bpmnEngine_minThreads_1	Minimum Threads Constraint
<b>default_bpmnInvoke</b>	<b>Work Manager</b>

**Create a New Work Manager Component**

Back Next Finish Cancel

Select Work Manager Definition type

What type of Work Manager, Request Class or Constraint

- ☒ Work Manager
- ☐ Response Time Request Class
- ☐ Fair Share Request Class
- ☐ Context Request Class
- ☐ Maximum Threads Constraint
- ☐ Minimum Threads Constraint
- ☐ Capacity Constraint

**Settings for default\_bpmnInvoke**

Configuration Targets Notes

Save

Use this page to define the request classes and constraints for the selected global Work Manager.

Name: (No value specified)

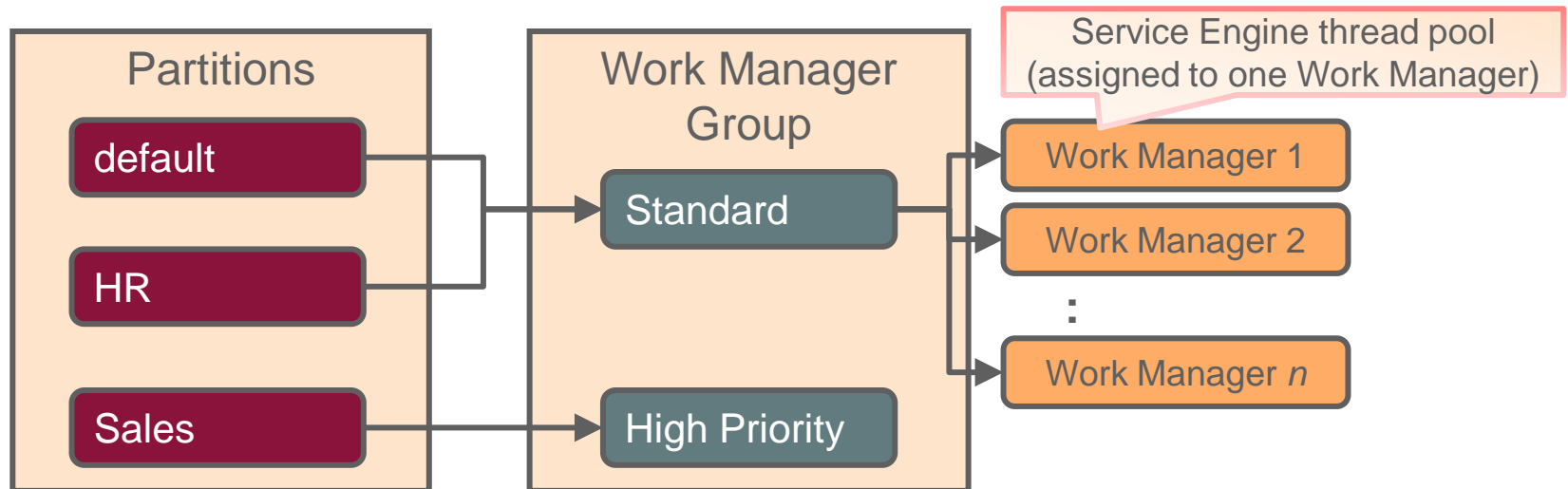
Request Class: soa\_fairShare\_20 (Fair Share Request Class) New

Minimum Threads Constraint: default\_bpmnInvoke\_minThreads\_1 New

Maximum Threads Constraint: SOAInternalProcessing\_maxThreads New

# Work Managers Groups and SOA Partitions

- A Work Manager Group:
  - Is a SOA concept that defines a group of Work Managers
  - Is associated with one or more SOA partitions
- A SOA partition is:
  - A way to groups applications for providing metadata isolation and optional thread isolation
  - Associated with one and only one Work Manager Group



# SOA Work Manager Examples

- Some non-partition Work Managers are:

Work Manager Name	Constraint Name	Description
SOA_Default_WM	None	The default Work Manager for all SOA services that do not access the soa data source such as Case Management
SOA_Notification_WM	SOAInternalProcessing_maxThreads	Notification MDB
SOA_Request_WM	SOA_Request_WM_minThreads_1 SOAIncomingRequests_maxThreads	Work Manager to handle SOA synchronous request clients: WS, Direct, Façade, ADF, Rest

- Some partition-specific Work Managers include:

Work Manager Name	Constraint Name	Description
default_Adapters	None	Adapter Work Manager, roughly included in the 30% buffer
default_dspSystem	soa_fairShare_20, minThreads_1 SOAInternalProcessing_maxThreads	BPEL System Dispatcher Work Manager
default_dsplInvoke	soa_fairShare_20, minThreads_1 SOAInternalProcessing_maxThreads	BPEL Process Invocation Dispatcher Work Manager
default_dspEngine	soa_fairShare_80, minThreads_1 SOAInternalProcessing_maxThreads	BPEL Process Engine Dispatcher Work Manager

# Configuring Work Manager Groups

Configuring a Work Manager Group is done in Oracle Enterprise Manager Fusion Middleware Control.

The screenshot illustrates the steps to configure a Work Manager Group in Oracle Enterprise Manager Fusion Middleware Control. It shows the navigation pane on the left, the main content area with the 'Work Manager Groups' table, and the 'Create Work Manager Group' dialog box.

**Step 1:** In the left navigation pane, the 'Work Manager Groups' link is highlighted. A red arrow points from this link to the 'Create ...' button in the main content area.

**Step 2:** The 'Create ...' button is highlighted. A red arrow points from this button to the 'Create Work Manager Group' dialog box.

**Table: Work Manager Groups**

Work Manager Group	Max Threads Constraint	Min Threads Constraint
default		
default_Adapters	0	0
default_Analytics	0	0
default_MediatorErrorHandling	25 [SOAInternalProcessin...	1 [default_MediatorError...
default_MediatorParallelRouting	25 [SOAInternalProcessin...	1 [default_MediatorParall...

**Callout:** Select **Metrics** from the Show drop-down list to monitor performance.

**Create Work Manager Group Dialog:**

A Work Group Manager comprises all logical thread pools associated with a given partition. Work Manager Groups are used to isolate partition configuration and request processing.

Work Manager Group Name:

Description:

Buttons: Create, Cancel



# Associating a Work Manager Group with a Partition

Associating a Work Manager Group with a partition is done when you create the partition or manage the partition configuration in Oracle Enterprise Manager Fusion Middleware Control.

The image illustrates the steps to create a new SOA Partition and associate it with a Work Manager Group in Oracle Enterprise Manager Fusion Middleware Control.

**Step 1:** In the SOA Infrastructure navigation tree, click **Manage Partitions** (indicated by a red box and a red arrow pointing to the 'Create' button in the 'Manage Partitions' dialog).

**Step 2:** In the **Create New SOA Partition** dialog, specify the details about your new partition and click **Create**. Once the partition is created, you will be able to access it in Enterprise Manager's navigation tree.

**Work Manager Group Selection:** Select or create a Work Manager Group to be used by the new partition. Work Manager Groups are used to isolate partition configuration and request processing. The **Work Manager Group** dropdown menu shows **default** (indicated by a red box and a red arrow pointing to the '+' icon).

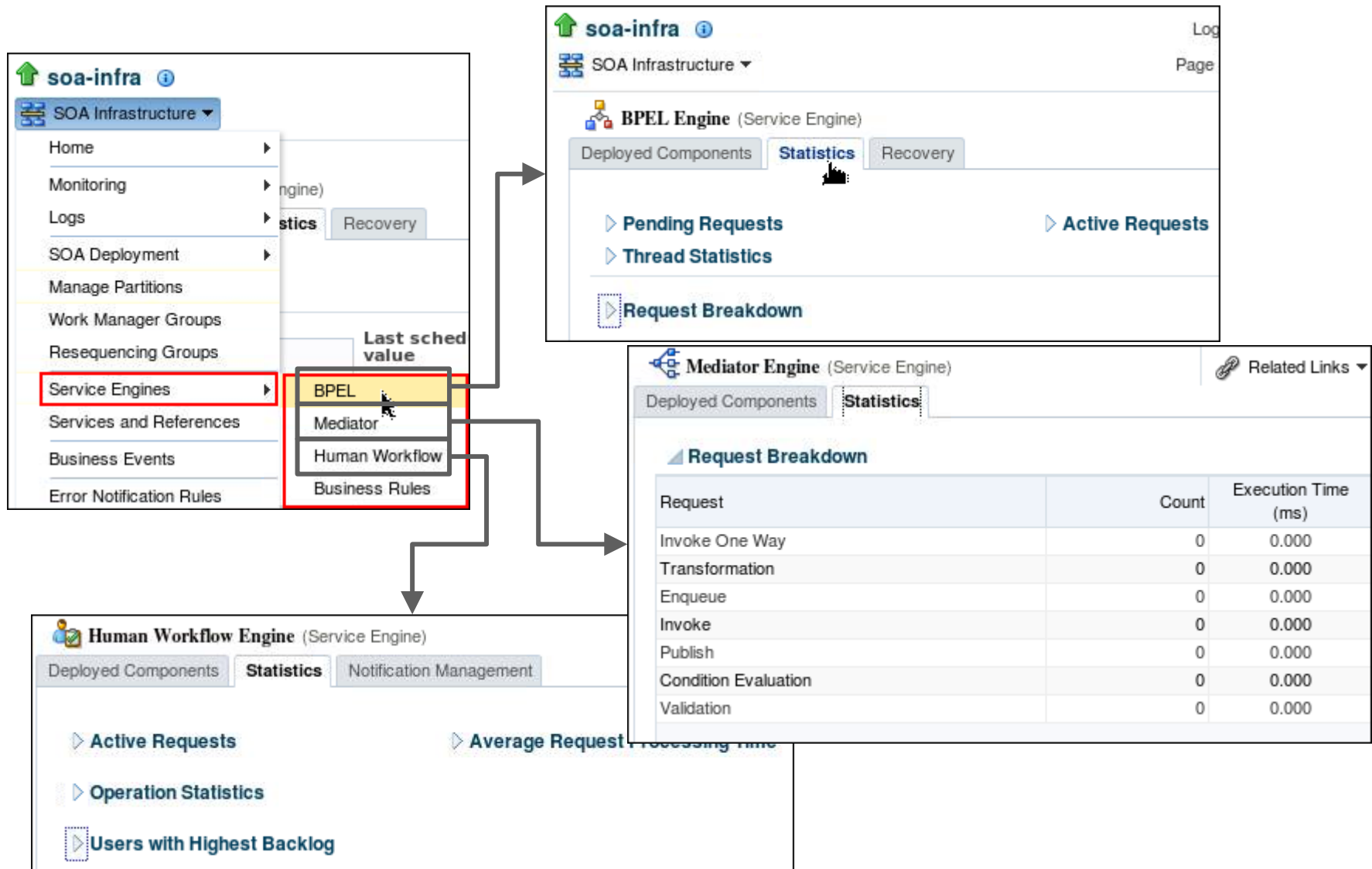
**Click to create a Work Manager Group.**

**Select a Work Manager Group from the list.**

**Click Edit to modify a selected partition.**

**Manage Partitions Dialog:** The dialog shows a table with columns **SOA Partition** and **Work Manager Group**. The table contains one row with **default** in both columns. The **Create** button is highlighted with a red box and a red arrow pointing to the 'Create' button in the 'Create New SOA Partition' dialog.

# Monitoring and Tuning Service Engines



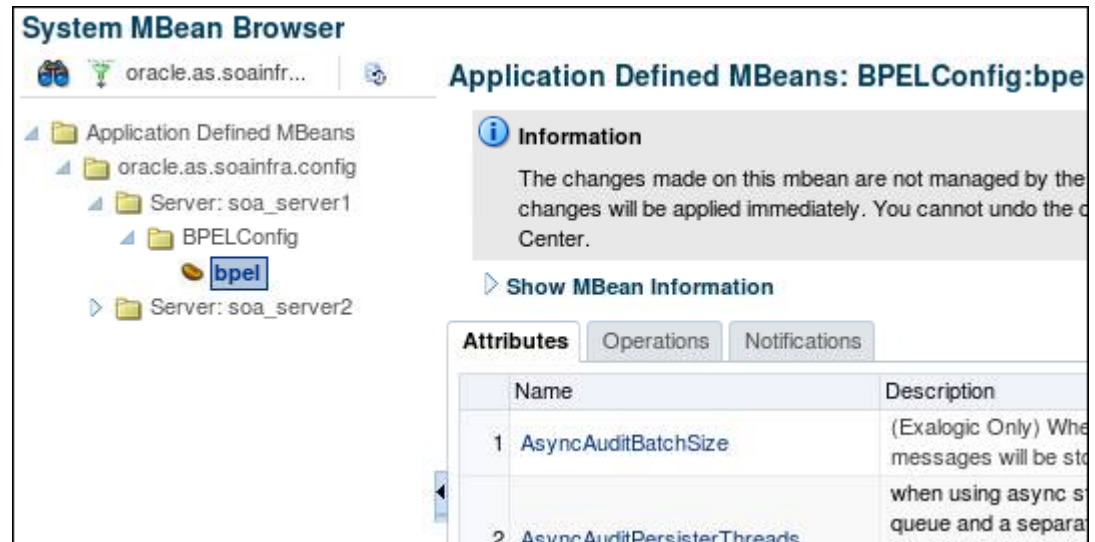
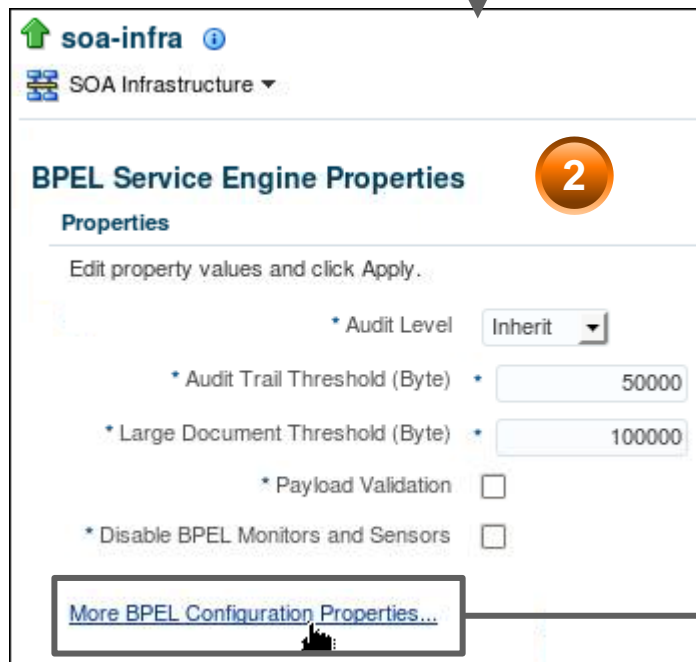
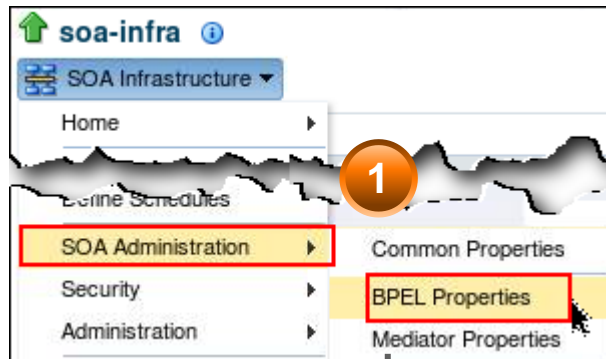
# Tuning the BPEL Engine Parameters

The BPEL Engine parameters that are *likely* or *highly likely* to improve performance are:

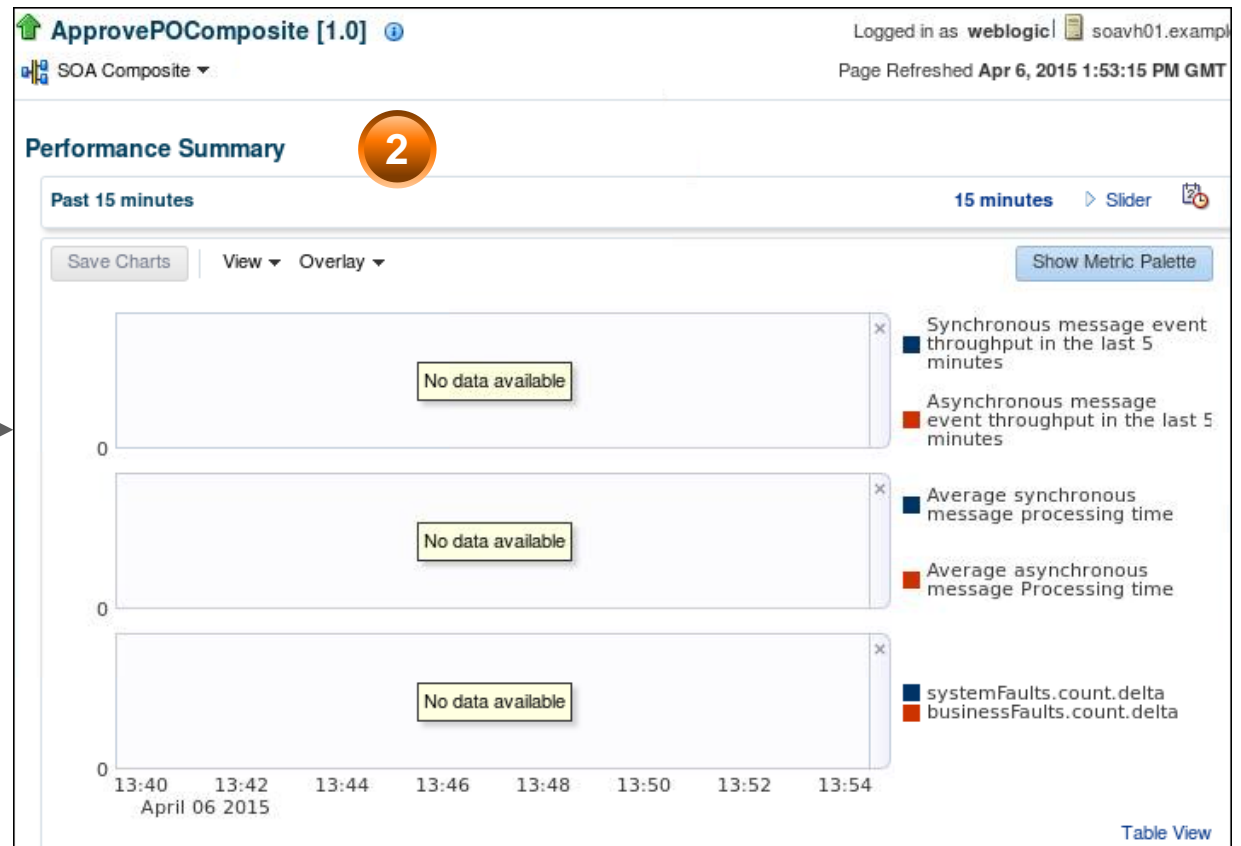
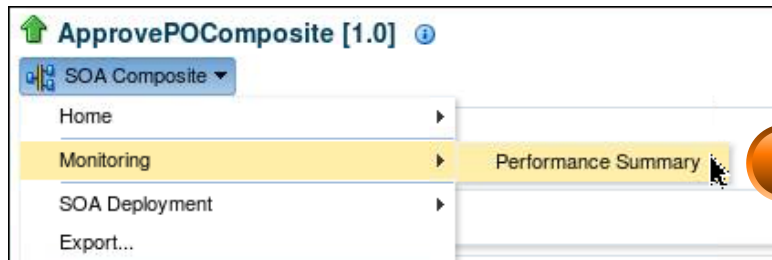
Parameter	Possible Action
auditLevel	Use the value Off to stop storing audit information.
SyncMaxWaitTime	Decrease value to improve performance.
largedocumentthreshold	Decrease value to improve performance.
validateXML	Use the default value False to improve performance.
InstanceKeyBlockSize	Increase to value greater than the number of updates to the ci_id_range table to improve performance.
Audit Level Threshold	Decrease value to improve performance.

**Note:** Always consider trade-offs for the actions described.

# Configuring BPEL Engine Parameters



# Monitoring SOA Composite Application Performance



# Monitoring an Inbound Adapter Properties

Monitoring inbound and outbound adapter properties are similar to each other. Images are for an outbound adapter example:

**Target Navigation**

View ▾

- Application Deployments
  - SOA
    - soa-infra (soa\_server1)
      - default
        - BookingSystem [1.0]
        - HelloWorld [1.0]
        - PublishStockUpdComposite [1.0]
        - SubscribeStockUpdComposite [1.0]**
        - TroubleSomeComposite [1.0]

**SubscribeStockUpdComposite [1.0]**

SOA Composite ▾

Active | Retire ... | Shut Down ... | Test ▾ | Se

**Dashboard** | Composite Definition | Flow Instances | Unit Tes

**Components**

**Services and References**

Name	Type
ExternalStoreDbRef	JCA Adapter

**ExternalStoreDbRef (Database Adapter)**

**Dashboard** | Policies | Properties | Adapter Reports

**Instances and Faults**

No data available

[Table View](#)

**ExternalStoreDbRef (Database Adapter)**

**Dashboard** | Policies | **Properties** | Adapter Reports

You can edit or delete the following binding properties. Click Add to add additional properties.

View ▾ + Add Revert...

Name (Operation or Port Type)	Value
GetActiveUnitOfWork (merge)	false
DmIType (delete)	delete
MappingsMetaDataURL (ExternalStoreDbRefSelect)	ExternalStoreDbRef-or-mappings.xml
DetectOmissions (merge)	true
DmIType (merge)	merge
DescriptorName (merge)	ExternalStoreDbRef.ExternalStore
DescriptorName (ExternalStoreDbRefSelect)	ExternalStoreDbRef.ExternalStore

# Accessing Adapter Logs

1. Access the composite application page that contains the adapter reference.
2. Click Related Links > *composite\_name* Composite Logs.
3. Enter search criteria to search for specific log information.

The screenshot displays the Oracle SOA Suite user interface. At the top, the page title is 'SubscribeStockUpdComposite [1.0]'. The user is logged in as 'weblogic' on 'soavh01.example.com'. The page was refreshed on 'Jun 2, 2015 12:53:05 AM GMT'. The 'Related Links' dropdown menu is open, showing 'SOA Infra Home' and 'SubscribeStockUpdComposite Composite Logs'. A red circle with the number '1' and an arrow points to the 'SubscribeStockUpdComposite Composite Logs' link. Below this, the 'Log Messages' section is visible. It includes a search bar with a 'Search' button and a tip: 'Tip: Enter one or more keywords separated by a comma. If keywords with \'. Example: weblogic, server\, weblogic server will search.' The search results table has columns: Time, Message Type, Message ID, and Message. The table is currently empty, displaying '(No messages matched the search criteria.)'.

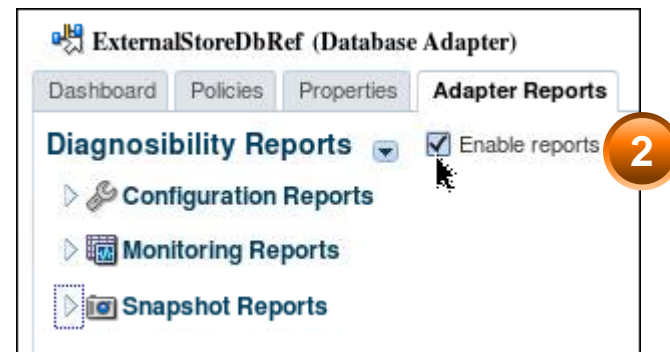
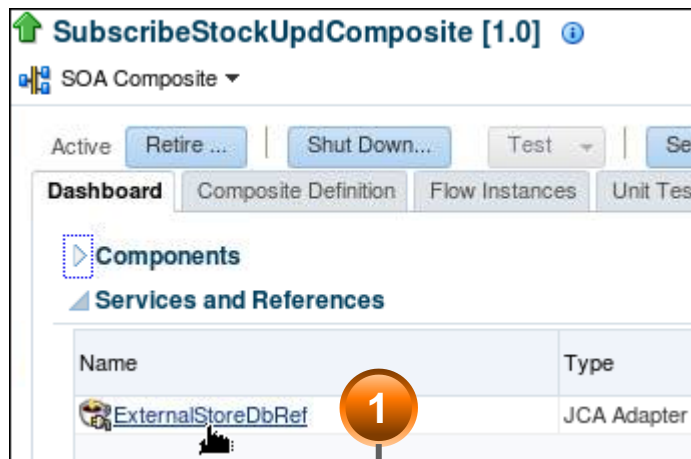
The JCA Adapter logger is called oracle.soa.adapter.



# Activating Adapter Reports

Adapter configuration through reports, such as:

- Connection Factory properties
- Service Definition properties
- Service Tuning properties
- Reference Definition properties
- Reference Tuning properties





# Selective Tracing Configuration: Introduction

Selective tracing is:

- Fine-grained logging for applications, users, request attributes
- Configured by clicking WebLogic Domain > Logs > Selective Tracing, for a domain, or by using WLST commands

**Target Navigation**

View ▾

- Application Deployments
- SOA
- WebLogic Domain
  - edg\_domain** 1
  - AdminServer

**edg\_domain** ⓘ

WebLogic Domain ▾

- Home
- Monitoring
- Diagnostics
- Control
- Logs**
- Deployments

View Log Messages

**Selective Tracing** 2

**Selective Tracing**

Use this page to configure the selective tracing for the selected logger.

**Information**

Selective Tracing should be used only for critical diagnostics purposes. This feature adds more diagnostic logging messages and could be a performance impact. Enable Selective Tracing only for the needed tracing options.

**Tracing Options** Active Traces And Tracing History

Use this page for configuring selective tracing options. The configuration settings done on this page will be applied to all the Weblogic servers of the Weblogic domain.

**Tracing Options**

Application Name: FtpAdapter

Level: **TRACE:32 (FINEST)** 3

Description:

Duration (minutes): 30

☒ Generate A New Unique Trace ID

☐ Use A Custom

Trace ID:

**Start Tracing** 4

Search for logger name to trace specific logger messages.

Logger Name: com.oracle.webservices

TRACE:32 (FINEST)

NOTIFICATION:16 (CONFIG)

TRACE:1 (FINE)

TRACE:16 (FINER)

TRACE:32 (FINEST)

# Viewing and Disabling Selective Tracing

On the domain > Logs > Selective Tracing > Active Traces and Tracing History tab page, you can:

- View:
  - The currently active selective traces
  - The history of selective traces
- Disable a selected trace entry in the table (if any)

The screenshot shows the 'Selective Tracing' interface with the 'Active Traces And Tracing History' tab selected. The interface includes a 'Tracing Options' sidebar and a main content area. The main content area has a description: 'Use this page to see the active traces and the disabled traces (tracing history)'. Below this, there are two sections: 'Active Traces' and 'Tracing History'. The 'Active Traces' section contains a table with columns 'Trace ID', 'Option Name', 'Option Value', and 'Desc'. The table is currently empty, displaying 'No active traces'. Above the table are buttons for 'Disable' and 'Disable and Create Incident'. The 'Tracing History' section contains a description: 'The table below shows a list of all the disabled traces.' and a 'Clear History' button. The table for tracing history is also empty.

**Selective Tracing**

Tracing Options **Active Traces And Tracing History**

Use this page to see the active traces and the disabled traces (tracing history).

**Active Traces**

The table below shows a list of all the active traces. An active trace can be disabled. A disabled trace will show up in the Tracing History table below.

**Disable** **Disable and Create Incident**

Trace ID	Option Name	Option Value	Desc
No active traces			

**Tracing History**

The table below shows a list of all the disabled traces.

**Clear History**

Trace ID	Option Name	Option Value	Desc
----------	-------------	--------------	------

# Agenda

- Tuning methodology: Overview
- Tools for monitoring Oracle Fusion Middleware environments
- Monitoring and tuning with Oracle SOA Suite
- Troubleshooting common problems

# Common Problems and Solutions

- Resolving message failure caused by too many open files
- Resolving connection timeouts
- Resolving common application-related problems

# Troubleshooting SOA Server Startup Failures

- Problem:  
You receive `ClassNotFoundException` errors when the SOA Managed Server is started by a Node Manager.
- Solution:  
Node Manager must be started with the property `StartScriptEnabled=true`, so that the Node Manager can use the start scripts that contain the environment information of the Managed Server to start the Managed Server.

# Resolving Message Failure Caused by Too Many Open Files

- Problem: You receive the following error at run time or compilation time:

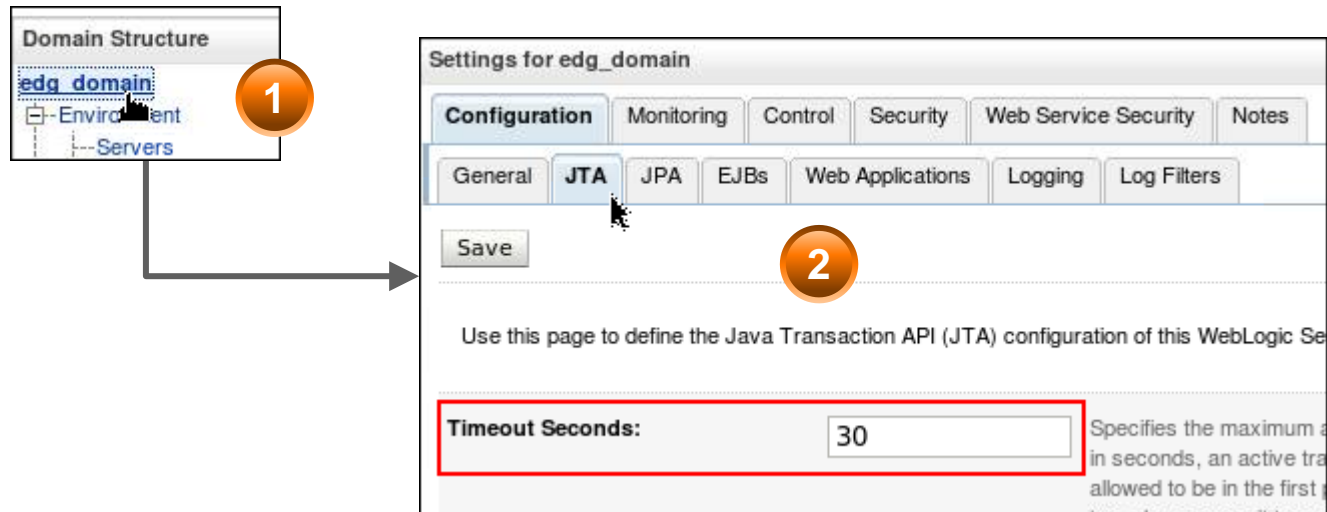
Message send failed: Too many open files

- Solution: (UNIX) Edit the `/etc/security/limits.conf` file to increase the value for file descriptors to at least 4096. For example:

```
#<domain>    <type>    <item>      <value>
#*           soft     core        0
#ftp         hard     nproc       0
...
# End of file
@svrgroup    soft     memlock     500000
@svrgroup    hard     memlock     500000
*            soft     nfile       4096
*            hard     nfile       4096
```

# Resolving Connection Timeouts

- Problem: You receive a connection timeout error under circumstances such as large payload or large number of message files.
- Solution: Increase the transaction timeout property in Oracle WebLogic Administration Console.



# Resolving Common Application-Related Problems

- Problem:
  - Incorrect XML messages structures
    - Incorrect namespace in the input data stream (for File Adapter or soap-initiated requests)
  - Incorrect configuration plans used with deployment
    - Incorrect URLs for service references
  - Missing configuration of runtime resources
    - Incorrect DbAdapter runtime resource configuration
    - Incorrect JMS Adapter runtime resource configuration
- Solution: Fix the application.



# Summary

In this lesson, you should have learned how to:

- Design a tuning methodology
- Identify top areas that require tuning
- Monitor and tune Oracle SOA Suite by collecting performance information and by using diagnostic tools
- Configure selective tracing
- Troubleshoot common problems:
  - Managed Server startup problems
  - Message failure caused by too many open files
  - Connection timeouts
  - Application-related problems



# Practice 13: Overview

This practice covers the following topics

- Part 1: Performance Tuning
  - Practice 13-1: Deploy an EAR application
  - Practice 13-2: Examine Server Information in the Routing Topology
  - Practice 13-3: Observe WebLogic Server Instance JVM Performance
  - Practice 13-4: Explore Metrics for Oracle SOA Suite Components
- Part 2: Troubleshooting
  - Practice 13-5: Deploy a Troublesome Composite Application
  - Practice 13-6: Testing the TroubleSomeComposite Interfaces