

## **Oracle® Enterprise Manager**

Getting Started with Oracle Fusion Middleware Management

11g Release 1 (11.1.0.1)

**E17558-05**

April 2011

Oracle Enterprise Manager Getting Started with Oracle Fusion Middleware Management, 11g Release 1 (11.1.0.1)

E17558-05

Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

Contributor: Venkat Audinarayan, Neelima Bawa, Jacqueline Gosselin, Maheshwari Hatte, Glen Hawkins, Arivalagan Kaliyaperumal, Shiraz Kanga, James Kao, Pushpa Raghavachar, Rishi Saraswat, Karthik Somasundaram

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

---

---

# Contents

<b>Preface</b> .....	xv
Audience .....	xv
Documentation Accessibility .....	xv
Related Documents .....	xvi
Terminology .....	xvi
Conventions .....	xvii
<b>Part I Managing Oracle Fusion Middleware</b>	
<b>1 Introduction to Middleware Management</b>	
1.1 Middleware Management in Enterprise Manager .....	1-1
1.2 Using Oracle Enterprise Manager Grid Control to Manage the Oracle Fusion Middleware Environment 1-2	
1.3 Oracle Fusion Middleware Control versus Oracle Enterprise Manager Grid Control ....	1-2
1.3.1 Managing Oracle Fusion Middleware with Fusion Middleware Control .....	1-2
1.3.2 Managing Oracle Fusion Middleware with Oracle Enterprise Manager 11g Grid Control 1-3	
<b>2 Discovering Middleware Targets</b>	
2.1 Discovering a WebLogic 9.x or 10.x Domain Via Grid Control .....	2-1
2.2 Discovering a WebLogic Server 7.x or 8.x Domain Via Grid Control.....	2-3
2.3 Discovering Multiple WebLogic Domains Via EMCLI.....	2-4
2.4 Enabling Automatic Discovery of New Domain Members.....	2-4
<b>3 Monitoring Middleware Targets</b>	
3.1 Monitoring Middleware Targets in Oracle Enterprise Manager Grid Control .....	3-1
3.1.1 Oracle Fusion Middleware Components .....	3-2
3.1.2 Oracle Application Server Components.....	3-4
3.1.3 Non-Oracle Middleware Components .....	3-4
3.2 Out-of-Box Monitoring of Middleware Targets .....	3-4
3.2.1 Target Home Page .....	3-4
3.2.2 Out-of-box Metrics.....	3-6
3.2.3 Analyzing Historical Performance.....	3-7
3.2.4 Setting Metric Thresholds for Alert Notifications.....	3-8
3.2.5 Monitoring Templates.....	3-8

3.2.6	Managing and Creating Blackouts .....	3-8
3.2.7	Extend Monitoring for Applications Deployed to WebLogic Server.....	3-9
3.2.8	Request Monitoring .....	3-9
3.2.8.1	Defining and Managing Transactions Being Monitored .....	3-10
3.3	Diagnosing Performance Problems .....	3-12
3.3.1	Using Home Pages to Diagnose Performance Issues .....	3-12
3.4	Administering Middleware Targets.....	3-13
3.4.1	Process Control .....	3-14
3.4.2	Creating WebLogic Domain Groups .....	3-14
3.5	Managing Configurations.....	3-14
3.5.1	Policy Violations .....	3-15
3.6	Managing Service Levels .....	3-15
3.7	Automating Operations .....	3-16
3.7.1	Job System.....	3-16
3.7.1.1	Log File Rotation.....	3-16
3.7.2	Operations Automation .....	3-17
3.7.2.1	Cloning from Test to Production Environments .....	3-17
3.7.2.2	Scaling Out Domains .....	3-18

## Part II Managing Oracle SOA

### 4 Overview of Oracle SOA Management Pack Enterprise Edition

### 5 Discovering and Monitoring Oracle BPEL Process Manager

5.1	Supported Versions .....	5-1
5.2	Understanding the Discovery Mechanism.....	5-2
5.3	Understanding the Discovery Process .....	5-3
5.4	Setting Up Oracle Software Library .....	5-4
5.5	Downloading One-Off Patches .....	5-5
5.6	Discovering BPEL Process Manager .....	5-6
5.6.1	Discovering BPEL Process Manager Deployed to Oracle Application Server .....	5-6
5.6.2	Discovering BPEL Process Manager Deployed to Oracle WebLogic Managed Server.....	5-7
5.6.2.1	Discovering Oracle WebLogic Managed Server .....	5-7
5.6.2.2	Discovering BPEL Process Manager Deployed to Oracle WebLogic Managed Server	5-8
5.6.3	Discovering BPEL Process Manager Deployed to IBM WebSphere Application Server..	5-9
5.6.3.1	Discovering IBM WebSphere Application Server .....	5-10
5.6.3.2	Discovering BPEL Process Manager Deployed to IBM WebSphere Application Server	5-10
5.7	Configuring BPEL Process Manager.....	5-12
5.7.1	Specifying Details for Monitoring BPEL Process Manager .....	5-12
5.7.2	Adding BPEL JAR Files To Agent CLASSPATH .....	5-13
5.7.3	Registering BPEL Process Manager Credentials and Host Credentials .....	5-14
5.8	Troubleshooting .....	5-14
5.8.1	404 Not Found Errors.....	5-14

5.8.2	Discovery Errors on Target Details Page (Adding BPEL Deployed to Oracle WebLogic Managed Server)	5-15
5.8.3	Discovery Errors on Review Page (Adding BPEL Deployed to Oracle WebLogic Managed Server)	5-15
5.8.4	Discovery Errors on Review Page (Adding BPEL Deployed to IBM WebSphere Application Server)	5-16
5.8.5	Display Errors on Processes Page.....	5-17
5.8.5.1	No Credentials Specified for Monitoring BPEL Process Manager.....	5-17
5.8.5.2	Required BPEL JAR Files Not Added To Agent CLASSPATH.....	5-18
5.8.6	Retrieving the OPMN Port.....	5-18
5.8.7	java.lang.NoClassDefFoundError Error.....	5-19
5.8.8	javax.naming.NameNotFoundException Error.....	5-19
5.8.9	javax.naming.NamingException Error.....	5-20
5.8.10	javax.naming.NoInitialContextException Error.....	5-21
5.8.11	Error While Creating BPEL Infrastructure Services.....	5-21
5.8.12	Metric Collection Errors for BPEL Process Manager Partner Link Metrics.....	5-22

## 6 Discovering and Monitoring Oracle Service Bus

6.1	Supported Versions.....	6-1
6.2	Understanding the Discovery Mechanism.....	6-1
6.3	Understanding the Discovery Process.....	6-2
6.4	Downloading One-Off Patches.....	6-3
6.5	Discovering Oracle Service Bus.....	6-3
6.5.1	Discovering OSB Deployed to Oracle WebLogic Managed Server That Is Not Monitored in Grid Control	6-4
6.5.2	Discovering OSB Deployed to Oracle WebLogic Managed Server That Is Already Monitored in Grid Control	6-4
6.6	Enabling Management Packs.....	6-5
6.7	Troubleshooting.....	6-5
6.7.1	Required Patches Missing.....	6-6
6.7.2	System and Service.....	6-6

## 7 Discovering and Monitoring SOA Suite 11g

7.1	New Features in This Release.....	7-1
7.2	Supported Versions.....	7-1
7.3	Understanding the Discovery Process.....	7-2
7.4	Discovering SOA Suite 11g.....	7-2
7.4.1	Discovering the SOA Suite 11g Using a Remote Agent.....	7-2
7.5	Post Discovery Steps.....	7-3
7.5.1	Configuring Instance Tracing.....	7-3
7.5.2	Viewing Application Dependency and Performance (ADP) Metrics.....	7-4
7.6	Creating Reports.....	7-4
7.7	Provisioning SOA Artifacts and Composites.....	7-4
7.8	Troubleshooting.....	7-5
7.8.1	Discovery.....	7-5
7.8.2	Monitoring.....	7-5
7.8.3	Instance Tracing.....	7-5

7.8.4	Faults .....	7-6
7.8.5	Application Dependency and Performance Integration.....	7-6
7.8.6	Reports.....	7-6
7.8.7	Systems and Services.....	7-7
7.8.8	BPEL Recovery .....	7-7

## Part III Installing and Configuring Performance Management Components

### 8 Installing and Configuring Performance Management Components

#### 9 Configuring JVM Diagnostics

9.1	JVM Diagnostics Architecture.....	9-1
9.2	Prerequisites .....	9-2
9.3	Deploying the JVM Diagnostics Manager.....	9-2
9.3.1	Running the DeployAD4Jmanager.sh Script.....	9-2
9.3.2	Deploying the JVM Diagnostics Agent on 64-bit JVMs .....	9-6
9.4	Scaling Out by Deploying Multiple JVM Diagnostics Managers .....	9-6
9.5	Deploying the JVM Diagnostics Agent.....	9-7
9.5.1	Automated Deployment Procedure.....	9-7
9.5.2	Manual Deployment Procedure .....	9-8
9.5.2.1	Installing the JVM Diagnostics Agent on Standalone Java Applications.....	9-8
9.5.2.1.1	Original Java Call .....	9-9
9.5.2.1.2	Modified Java Call with JVM Diagnostics Agent.....	9-9
9.5.2.1.3	Sample Configuration.....	9-9
9.5.2.1.4	Adding the JVM Diagnostics Agent Libraries, Classes to Classpath .....	9-10
9.5.2.1.5	Target Parameters .....	9-10
9.5.2.2	Customizing the JVM Diagnostics Agent .....	9-10
9.5.2.2.1	Web.XML Contents.....	9-11
9.6	Undeploying the JVM Diagnostics Agent.....	9-12
9.7	Deploying the Database Agent .....	9-12
9.8	Creating a Special Repository User for Loading JVM Diagnostics Heaps .....	9-14

#### 10 Installing and Configuring Application Dependency and Performance

10.1	Quick Steps for Installing and Configuring the ADP Manager .....	10-1
10.2	Pre-Installation Requirements.....	10-5
10.2.1	Environment Considerations .....	10-5
10.2.1.1	Access and Connectivity.....	10-6
10.2.1.1.1	Access to Monitored Application Platform .....	10-6
10.2.1.1.2	Network Connectivity .....	10-6
10.2.1.1.3	DHCP .....	10-6
10.2.1.2	ADP Database .....	10-6
10.2.2	Clustering and Application Domains.....	10-7
10.2.2.1	WebLogic Clustering.....	10-7
10.2.2.1.1	Clusters .....	10-7
10.2.2.1.2	Node Manager .....	10-7
10.2.3	Statistics and Privileges To Obtain From Various Administrators .....	10-7

10.2.3.1	Database Administrator .....	10-7
10.2.3.2	WebLogic Administrator.....	10-8
10.2.3.3	WebSphere Administrator .....	10-8
10.2.3.4	OC4J Administrator .....	10-8
10.3	Installation Process .....	10-8
10.3.1	Installing ADP - Windows Installation .....	10-8
10.3.2	Installing ADP - Solaris, Linux, and AIX Installations.....	10-9
10.3.3	Setting Up ADP Data Repository .....	10-10
10.3.4	Deploying ADP Components .....	10-10
10.3.4.1	Configuring ADP.....	10-10
10.3.4.2	Deploying ADP Agents on the WebLogic Platform.....	10-11
10.3.4.2.1	HttpDeployer.ear Procedure .....	10-11
10.3.4.2.2	Deploying the ADP Java Agent.....	10-11
10.3.4.3	Deploying ADP Agents on the WebSphere Platform .....	10-11
10.3.4.3.1	Required IBM WebSphere Libraries.....	10-11
10.3.4.3.2	Automatic ADP Agent Deployment for WebSphere Platform.....	10-12
10.3.4.3.3	Deploying ADP Java Agent for Oracle Platform.....	10-12
10.3.4.4	Running Multiple ADP Instances .....	10-12
10.3.4.4.1	Front-End Configuration.....	10-13
10.3.5	Installing and Configuring ADP Data Repository.....	10-13
10.3.5.1	Configuring Oracle DBMS for ADP.....	10-13
10.3.5.2	Configuring MySQL DBMS for ADP .....	10-14
10.3.6	Backing Up the ADP Database .....	10-14
10.4	Post-Installation Requirements.....	10-14
10.4.1	IBM WebSphere Post-Deployment Requirements.....	10-15
10.4.1.1	Configuring ADP for WebSphere Application Server 6.1 Secured Connections.....	10-15
10.4.1.2	Configuring WebSphere 5.1 for Secure Connectivity .....	10-15
10.4.2	Configuring Oracle SOA Suite for Secure Connectivity .....	10-17
10.4.3	Configuring Oracle WebLogic Server or Oracle WebLogic Portal (WLP) for Secure Connectivity .....	10-17
10.4.4	Importing a Certificate into the Manager's Keystore .....	10-17
10.4.5	Configuring the ADP Agent When WebLogic Is Installed As a Windows Service.....	10-19

## 11 Configuring Request Monitoring

11.1	Request Monitoring Architecture.....	11-1
11.2	Prerequisites .....	11-2
11.3	Configuring Request Monitoring .....	11-2

## Part IV JVM Diagnostics

### 12 Introduction to JVM Diagnostics

12.1	Overview .....	12-1
12.1.1	Java Activity Monitoring and Diagnostics with Low Overhead .....	12-2
12.1.2	In-depth Visibility of JVM Activity.....	12-2

12.1.3	Real Time Transaction Tracing .....	12-2
12.1.4	Cross-Tier Correlation with Oracle Databases .....	12-2
12.1.5	Memory Leak Detection and Analysis .....	12-2
12.1.6	JVM Pooling.....	12-3
12.1.7	Real-time and Historical Diagnostics.....	12-3
12.2	New Features in this Release.....	12-3
12.2.1	One-off Patch Enhancements .....	12-3
12.3	Supported Platforms and JVMs .....	12-4
12.4	User Roles.....	12-4

## 13 Using JVM Diagnostics

13.1	Accessing the JVM Diagnostics Pages .....	13-1
13.2	Viewing the Domain (Pool) Summary and Details Pages .....	13-2
13.3	Viewing the Domain (Pool) Real Time Thread Analysis Page.....	13-3
13.4	Viewing the JVM Summary and Details Pages .....	13-4
13.5	Viewing the JVM Real Time Thread Analysis Page .....	13-5
13.6	Viewing the JVM Real Time Memory Analysis Page.....	13-6
13.7	Using the Setup Pages.....	13-6
13.7.1	Console Setup Page .....	13-7
13.7.2	JVM Pool Info .....	13-7
13.7.2.1	Updating a JVM Pool .....	13-7
13.7.2.2	Update Pool Thresholds .....	13-7
13.7.3	Databases Page.....	13-9
13.7.4	JVMs and Managers Page.....	13-9
13.7.5	Monitoring.....	13-9
13.7.6	Trace Files Page.....	13-9
13.7.7	Download Agent Page.....	13-9
13.7.7.1	Downloading the Java Agent.....	13-10
13.8	Analyzing the JVM Heaps .....	13-10
13.8.1	All Heaps Page .....	13-10
13.8.2	Heap Usage Page .....	13-10
13.8.3	Heap Usage by Roots .....	13-10
13.8.4	Top 40 Objects Page.....	13-10
13.8.5	Heap Object Information Page .....	13-11
13.8.6	Compare Heaps Page.....	13-11
13.9	Using the Database Pages.....	13-11
13.10	Tracing JVM Threads.....	13-12
13.10.1	Traces Active Threads Page .....	13-12
13.10.2	Trace Summary Page.....	13-12
13.10.3	Available Threads Page .....	13-13
13.10.4	Thread Summary Page.....	13-13

## 14 Troubleshooting JVM Diagnostics

14.1	Agent Automated Deployment Errors .....	14-1
14.1.1	DeployAd4jAgentOnTarget Step Errors .....	14-2
14.1.2	CreateTempDir Step Errors .....	14-3
14.1.3	Active State Error.....	14-3



14.2	Secure Communication Errors .....	14-4
14.3	Cross Tier Functionality Errors.....	14-5
14.4	Trace Errors.....	14-6
14.5	Deployment Script Execution Errors .....	14-6
14.5.1	Deployment on 64-bit JVMs.....	14-7
14.5.2	JVM Diagnostics Manager Automated Deployment.....	14-8
14.6	LoadHeap Errors.....	14-8
14.7	Errors on JVM Diagnostics UI Pages.....	14-9
14.8	Frequently Asked Questions .....	14-9
14.8.1	Location of the JVM Diagnostics Logs.....	14-10
14.8.2	JVM Diagnostics Manager Status .....	14-10
14.8.3	JVM Diagnostics Agent Status .....	14-10
14.8.4	Monitoring Status .....	14-11
14.8.5	Running the create_jvm_diagnostic_db_user.sh Script .....	14-11
14.8.6	Usage of the Try Changing Threads Parameter .....	14-11
14.8.7	Significance of Optimization Levels .....	14-11
14.8.8	Manually Deploying the JVM Diagnostics Agent .....	14-11
14.8.9	Log Manager Level.....	14-12
14.8.10	Repository Space Requirements .....	14-12

## Part V Using Application Dependency and Performance

### 15 Introduction to Application Dependency and Performance

15.1	Overview .....	15-1
15.1.1	Managing Complex Java EE, SOA, and Portal Applications .....	15-1
15.1.2	Delivering a Service-Oriented View Across Environments .....	15-2
15.1.3	Avoiding Involvement from Java EE, SOA, Portal, and Application Experts.....	15-3
15.1.4	Eliminating Repetitive Do-It-Yourself (DIY) Manual Processes.....	15-3
15.1.5	ADP Solution.....	15-4
15.2	Architecture .....	15-4
15.2.1	ADP Java Agents .....	15-5
15.2.2	ADP Manager.....	15-6
15.2.2.1	ADP Manager and High Availability .....	15-6
15.2.3	ADP Database .....	15-6
15.2.4	ADP User Interface.....	15-6

### 16 Exploring Application Dependency and Performance

16.1	Exploring the User Interface.....	16-1
16.1.1	Accessing ADP .....	16-2
16.1.2	General ADP UI Elements .....	16-2
16.1.3	Drill Down in Operational Dashboard .....	16-2
16.1.4	Configuring Service Level Objectives (SLOs).....	16-3
16.1.4.1	Create New SLO .....	16-3
16.1.4.2	Defining SLO Parameters .....	16-3
16.1.4.2.1	Propagating Threshold Violation Events.....	16-4
16.1.4.2.2	Types of SLOs .....	16-4

16.1.4.2.3	SLO Events Viewer .....	16-4
16.1.5	Configuring SLO Blackouts.....	16-4
16.1.6	Time Frame.....	16-6
16.1.7	Display Interval.....	16-6
16.1.7.1	Time Frame.....	16-6
16.1.7.2	Interval Context .....	16-6
16.1.7.3	Turning Off Time Frame Limitation.....	16-7
16.1.8	Graphs and Data Items .....	16-7
16.1.9	Right-Click Operations on Tables and Graphs.....	16-8
16.1.10	Comparative View.....	16-8
16.1.11	Save as PDF.....	16-9
16.1.12	Easy Scroller .....	16-9
16.1.13	Zoom In and Zoom Out Toolbar .....	16-9
16.1.14	Custom Metrics .....	16-9
16.1.15	Functional View .....	16-10
16.1.16	Topology View.....	16-10
16.1.17	Architecture View.....	16-11
16.1.17.1	Accessing the Architecture View .....	16-12
16.1.17.1.1	Arrows in Architecture Views.....	16-12
16.1.17.1.2	Architecture View Summary .....	16-12
16.1.18	Metric Types .....	16-13
16.2	Exploring the Monitoring Tab .....	16-13
16.2.1	Oracle WebLogic Portals .....	16-14
16.2.1.1	Desktops.....	16-16
16.2.1.1.1	Display Portal Desktop - Desktop Structure Viewer .....	16-16
16.2.1.2	Portlet Drill Down .....	16-17
16.2.1.3	Pageflow Viewer.....	16-17
16.2.1.4	Books .....	16-17
16.2.1.5	Pages.....	16-18
16.2.1.6	Portlets.....	16-18
16.2.2	WebSphere Portals.....	16-19
16.2.2.1	Virtual Portals .....	16-20
16.2.2.1.1	Display Virtual Portal - Structure Viewer .....	16-20
16.2.2.2	Pages.....	16-21
16.2.2.3	Portlets.....	16-21
16.2.3	Oracle BPEL Processes .....	16-22
16.2.3.1	Delay Analysis View .....	16-23
16.2.3.2	Metadata View .....	16-23
16.2.3.3	Partner Links View .....	16-24
16.2.3.4	Partner Link Type Role View.....	16-24
16.2.3.5	Partner Link Bindings View.....	16-24
16.2.3.6	Modeled Entities View.....	16-25
16.2.3.7	Topology View.....	16-25
16.2.3.8	Node Hierarchy .....	16-25
16.2.4	Oracle ESB.....	16-25
16.2.4.1	Service Details View.....	16-26
16.2.4.2	Service Parent Details View .....	16-27

16.2.4.3	Service Definition View .....	16-27
16.2.4.4	Service Operations View .....	16-27
16.2.4.5	Operation Routing Rules View.....	16-28
16.2.5	Oracle WebCenter.....	16-28
16.2.5.1	ADF Task Flows.....	16-28
16.2.5.1.1	User-Defined Taskflows.....	16-28
16.2.5.1.2	Web 2.0 Service .....	16-29
16.2.5.2	JSF Pages .....	16-29
16.2.5.3	Portlets .....	16-29
16.2.6	Processes .....	16-30
16.2.6.1	Node Hierarchy .....	16-31
16.2.6.1.1	Delay Analysis View.....	16-31
16.2.6.1.2	Events View.....	16-31
16.2.6.2	Persistent Containers .....	16-32
16.2.6.2.1	Entity EJB Activity Table.....	16-32
16.2.6.2.2	Entity EJB Cache Table .....	16-33
16.2.6.2.3	Entity EJB Transactions Table.....	16-33
16.2.6.2.4	Entity EJB Locking Table.....	16-33
16.2.6.3	Instrumentation .....	16-34
16.2.7	Web Services.....	16-35
16.2.8	Pageflows .....	16-35
16.2.9	Services.....	16-35
16.2.9.1	HTTP .....	16-36
16.2.9.2	EJBs .....	16-36
16.2.9.3	JDBCs.....	16-36
16.2.10	WSRP Producers .....	16-37
16.2.10.1	WSRP Summary .....	16-37
16.2.10.2	WSRP Topology .....	16-38
16.2.10.3	Display Portal Desktop.....	16-39
16.2.11	Integration.....	16-39
16.2.11.1	Health .....	16-40
16.2.11.1.1	Execute Queues.....	16-40
16.2.11.1.2	Async Dispatchers.....	16-41
16.2.11.1.3	Sync Dispatchers .....	16-41
16.2.11.1.4	JMS Destinations .....	16-42
16.2.11.1.5	Stateless Containers .....	16-43
16.2.11.1.6	Persistent Containers .....	16-43
16.2.11.2	Performance .....	16-45
16.2.11.3	Channels .....	16-46
16.2.11.4	Subscribers.....	16-47
16.2.12	Applications.....	16-47
16.2.12.1	Services.....	16-49
16.2.12.2	Dependencies .....	16-49
16.2.12.3	Deployments .....	16-50
16.2.12.4	Workshop Projects.....	16-50
16.2.12.5	Web Applications .....	16-51
16.2.12.6	Stateless Beans.....	16-51

16.2.12.7	Stateful Beans .....	16-52
16.2.12.7.1	Stateful EJB Cache .....	16-52
16.2.12.7.2	Stateful EJB Transactions.....	16-52
16.2.12.7.3	Stateful EJB Locking.....	16-53
16.2.12.8	Entity Beans .....	16-53
16.2.12.8.1	Entity EJB Activity.....	16-53
16.2.12.8.2	Entity EJB Cache .....	16-54
16.2.12.8.3	Entity EJB Transactions .....	16-54
16.2.12.8.4	Entity EJB Locking.....	16-55
16.2.12.9	Message Driven Beans .....	16-55
16.2.12.9.1	Message Driven EJB Activity .....	16-56
16.2.12.9.2	Message Driven EJB Transactions.....	16-56
16.2.13	Oracle WebLogic Resources .....	16-57
16.2.14	WebSphere Resources .....	16-57
16.2.15	Oracle Resources .....	16-58
16.2.16	Custom Metrics .....	16-59
16.2.17	CAMM Node .....	16-59
16.2.18	Service Component Architecture (SCA).....	16-60
16.2.18.1	Components .....	16-60
16.3	Exploring the Configuration Tab.....	16-61
16.3.1	Database Configuration .....	16-61
16.3.2	Resource Configuration .....	16-61
16.3.3	Service Level Objective Configuration .....	16-61
16.3.3.1	Service Level Objectives Configuration .....	16-61
16.3.4	SLO Blackout Configuration .....	16-62
16.3.5	Action Configuration .....	16-62
16.3.6	Custom Metric Configuration.....	16-63
16.4	Exploring the Registration Tab .....	16-64
16.4.1	Adding a New Manager (RMI Configuration).....	16-64
16.4.2	Editing a Previously Configured Manager (RMI Configuration) .....	16-64
16.4.3	Removing or Disabling a Previously Configured Manager .....	16-65

## 17 Performance Analytics

17.1	Entity Performance Ranking .....	17-1
17.2	Performance Characterization .....	17-1
17.2.1	Multi-Point Performance/Load Regression .....	17-2
17.2.2	Performance/Load Scattergram.....	17-2
17.2.3	Time-Based Performance Distribution .....	17-2
17.2.4	Performance Histogram.....	17-2
17.2.5	Time-Based Performance Trend .....	17-2
17.3	Memory Leak Detection.....	17-3
17.4	Drill Down - Bottleneck Analysis .....	17-3
17.5	Drill Out - Impact Analysis .....	17-4

## 18 Exporting Data

18.1	Data Export Modes .....	18-1
18.1.1	Export to File .....	18-1

18.1.2	Export to Database.....	18-1
18.1.3	Aggregation Export to File .....	18-1
18.2	ADP Export Configuration.....	18-1
18.2.1	ADP Periodic Export Configuration.....	18-2
18.2.2	Manual Execution of Metric Export.....	18-2
18.2.3	export.xml File.....	18-3
18.3	Example of Exported Data for WebLogic.....	18-4

## 19 ADP Methodology

19.1	ADP Methodology Activities.....	19-2
19.1.1	Map Business SLAs to Performance SLOs.....	19-2
19.1.2	Specify Target Performance Characteristics .....	19-3
19.1.3	Performance Improvement Process .....	19-3
19.1.3.1	Characterize Baseline Performance .....	19-3
19.1.3.2	Identify Performance Bottlenecks .....	19-4
19.1.3.3	Remove Performance Bottlenecks.....	19-4
19.1.3.4	Set SLOs on Key Metrics.....	19-4
19.2	Map Business SLAs to Performance SLOs .....	19-5
19.3	Characterize Baseline Performance.....	19-6
19.4	Identify Performance Bottlenecks.....	19-7
19.4.1	System Level Performance .....	19-7
19.5	Set SLOs on Key Metrics .....	19-8
19.6	Conclusion .....	19-10

## 20 Frequently Asked Questions About ADP

20.1	How Many Resources Can Be Added to the ADP Manager? .....	20-1
20.2	Can I Erase the darchive Directory?.....	20-1
20.3	How Do I Undeploy the Agent?.....	20-1

## A Configuration Directories and Files

A.1	Configuration Directories.....	A-1
A.1.1	Directory Structure .....	A-1
A.1.2	Config Directory .....	A-2
A.1.3	BIN Directory .....	A-2
A.1.4	Deploy Directory .....	A-3
A.1.5	apache-tomcat-5.5.20 Directory .....	A-3
A.2	Acsera.properties File.....	A-3
A.2.1	Log Files Management.....	A-3
A.2.2	Multi-Domain Monitoring Configuration.....	A-4
A.2.3	ADP RMI Port Assignment .....	A-4
A.2.4	ADP Aggregation and Data Life Time Configuration .....	A-4
A.2.5	Aggregating Incoming Metrics On the Fly .....	A-5
A.2.6	Configuring List of Applications To Be Monitored or Excluded From Monitoring. ....	A-5
A.2.7	Firewall Mitigation (for Internal RMI Ports) .....	A-5
A.2.8	SLO Dampening.....	A-5
A.3	UrlMap.properties .....	A-6

## **B Support Matrix for Application Dependency and Performance**

B.1	Standalone ADP Manager System Requirements.....	B-1
B.2	ADP Manager Platform Support .....	B-1

### **Index**

---

---

# Preface

This guide introduces you to managing Oracle Fusion Middleware 11g with Oracle Enterprise Manager Grid Control 11g Release 1.

The preface covers the following:

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

## Audience

This document is intended for users who want to discover and monitor Oracle Fusion Middleware products such as Oracle WebLogic Server and Oracle SOA components, as well as deployed Java EE applications. It is also intended for users of JVM Diagnostics and Application Dependency and Performance.

## Documentation Accessibility

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

### Accessibility of Code Examples in Documentation

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

## Accessibility of Links to External Web Sites in Documentation

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

## Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/support/contact.html> or visit <http://www.oracle.com/accessibility/support.html> if you are hearing impaired.

## Related Documents

For more information, see the following documents in the Enterprise Manager Grid Control documentation library:

- *Oracle Enterprise Manager Concepts*
- *Oracle Enterprise Manager Grid Control Basic Installation Guide*

For the latest releases of these and other Oracle documentation, check the Oracle Technology Network at

<http://otn.oracle.com/documentation/oem.html>

Oracle Enterprise Manager also provides extensive online Help. Click **Help** at the top of any Enterprise Manager page to display the online help window.

## Terminology

The following terminology is used throughout this manual.

Term	Definition
ASM	Application Service Management
APM	Application Performance Management
DIY	Do-It-Yourself
ISV	Independent Software Vendor
Request Trace	A single thread executing a request. Shows as a bar in the hierarchy view (Application Dependency and Performance).
Request Trace Events	The nodes in the left pane (Application Dependency and Performance)
Request Trace Event View	Transaction event view (Application Dependency and Performance)
SLA	Service Level Agreement
SLO	Service Level Objective
SOA	Service-Oriented Architecture
Transaction Analysis	Transaction tracing (Application Dependency and Performance)
Transaction Hierarchy View	The Transaction Hierarchy View displays a complete transaction hierarchy in a Gantt chart representation showing the execution of each transaction on an absolute timeline.
UI	User Interface
WSRP	Web Services Remote Portlets



Term	Definition
WSDL	Web Services Description Language

## Conventions

The following text conventions are used in this document:

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



# Part I

---

## Managing Oracle Fusion Middleware

The chapters in this part describe how you can discover and monitor Oracle Fusion Middleware targets, including Oracle WebLogic Server and deployed Java EE applications.

Part I contains the following chapters:

- [Chapter 1, "Introduction to Middleware Management"](#)
- [Chapter 2, "Discovering Middleware Targets"](#)
- [Chapter 3, "Monitoring Middleware Targets"](#)



---

---

# Introduction to Middleware Management

This chapter describes how you can use Enterprise Manager to monitor and manage middleware software including Oracle Fusion Middleware 11g and Oracle WebLogic Server.

This chapter covers the following:

- [Middleware Management in Enterprise Manager](#)
- [Using Oracle Enterprise Manager Grid Control to Manage the Oracle Fusion Middleware Environment](#)
- [Oracle Fusion Middleware Control versus Oracle Enterprise Manager Grid Control](#)

## 1.1 Middleware Management in Enterprise Manager

Middleware is the software that connects software components or enterprise applications. Middleware is the software layer that lies between the operating system and the applications on each side of a distributed computer network. Typically, it supports complex, distributed business software applications.

Middleware is the infrastructure which facilitates creation of business applications, and provides core services like concurrency, transactions, threading, messaging, and the SCA framework for service-oriented architecture (SOA) applications. It also provides security and enables high availability functionality to your enterprise.

Middleware includes Web servers, application servers, content management systems, and similar tools that support application development and delivery. It is especially integral to information technology based on Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web services, SOA, Web 2.0 infrastructure, and Lightweight Directory Access Protocol (LDAP), etc.

Managing these applications and the underlying middleware technology can be difficult and IT organizations often have to rely on a variety of specialized tools. This can lead to inefficiency and may introduce complexities and risks. Enterprise Manager Grid Control is a definitive tool for middleware management and allows you to manage both Oracle applications and custom Java EE applications that run on a combination of Oracle Middleware and non Oracle Middleware software.

This section introduces you to Enterprise Manager Grid Control which offers a unique management solution for Oracle and non Oracle middleware technologies.

## 1.2 Using Oracle Enterprise Manager Grid Control to Manage the Oracle Fusion Middleware Environment

After you install and configure Oracle Fusion Middleware, you can use Oracle Enterprise Manager 11g Fusion Middleware Control to manage the Oracle Fusion Middleware environment. Fusion Middleware Control is a Web browser-based, graphical user interface that you can use to monitor and administer a farm with a single Oracle WebLogic Server domain.

A farm is a collection of components managed by Fusion Middleware Control. It can contain one 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.

Fusion Middleware Control organizes a wide variety of performance data and administrative functions into distinct, Web-based home pages for the farm, domain, servers, components, and applications. The Fusion Middleware Control home pages make it easy to locate the most important monitoring data and the most commonly used administrative functions—all from your Web browser.

However, you can manage only one Fusion Middleware farm with Fusion Middleware Control. Typically, an enterprise would have multiple farms. Managing several farms from different, distinct Fusion Middleware Control consoles becomes very difficult.

For centralized management and additional management functionality (such as historical data collections for performance trending, email and page notifications for potential availability and/or performance problems, and service level management), you can use Enterprise Manager Grid Control.

## 1.3 Oracle Fusion Middleware Control versus Oracle Enterprise Manager Grid Control

You can manage your Oracle Fusion Middleware environment using Oracle Enterprise Manager Fusion Middleware Control (Fusion Middleware Control) or Oracle Enterprise Manager Grid Control.

### 1.3.1 Managing Oracle Fusion Middleware with Fusion Middleware Control

Fusion Middleware Control organizes a wide variety of performance data and administrative functions into distinct, Web-based home pages for the farm, cluster, domain, servers, components, and applications. The Fusion Middleware Control home pages make it easy to locate the most important monitoring data and the most commonly used administrative functions all from your Web browser.

Fusion Middleware Control is a part of the Oracle Fusion Middleware 11g installation. With Fusion Middleware Control, you can:

- Manage a single Oracle Fusion Middleware Farm and a single WebLogic Domain.
- Monitor the availability and performance of Fusion Middleware software in real time mode.
- Perform routine administration tasks such as deploying applications, configuring parameters etc.

For more details, see the *Oracle Fusion Middleware 11g Administrator's Guide*.

## 1.3.2 Managing Oracle Fusion Middleware with Oracle Enterprise Manager 11g Grid Control

Oracle Enterprise Manager 11g Grid Control is a Web browser-based, graphical user interface that you can use to monitor multiple Oracle Fusion Middleware Farms and Oracle WebLogic Server Domains. In fact, Grid Control provides deep management solutions for Oracle technologies including Oracle packaged applications, Oracle Database and Oracle VM. Grid Control also offers extensive support for non-Oracle technologies through more than two dozen heterogeneous management plug-ins and connectors including Microsoft MOM, IBM WebSphere, JBoss, EMC storage, F5 BIG IP, Check Point Firewall, and Remedy.

Oracle Enterprise Manager 11g Grid Control needs to be installed separately as it is not a part of the Oracle Fusion Middleware 11g installation. With Oracle Enterprise Manager 11g Grid Control, you can:

- Centrally manage multiple Oracle Fusion Middleware Farms and WebLogic Domains.
- Manage third party tools such as WebSphere, JBoss, Tomcat, and Microsoft.NET Framework.
- Manage non-middleware software such as underlying operating systems and hardware on which the middleware software is installed. This allows administrators to correlate middleware performance with its underlying host performance.
- Manage database software and diagnose application performance problems and identify the true root cause of the problem and the tier (middleware, database) on which it occurs.
- Monitor the availability and performance of Oracle Fusion Middleware software in real time and from a historical perspective for trend analysis.
- Diagnose availability and performance problems.
- Monitor and trace important end-user requests from the client to the service endpoint across all the servers and applications associated with each transaction.
- Use Application Dependency and Performance (ADP) to analyze J2EE and SOA applications.
- Monitor Java applications and diagnose performance problems in production using JVM Diagnostics.
- Define Service Level Objectives (SLOs) in terms of out-of-box system-level metrics as well as end user experience metrics to accurately monitor and report on Service Level Agreement (SLA) compliance.
- Perform several critical tasks like:
  - Setting thresholds on performance metrics. When these thresholds are violated, email and page notifications are sent.
  - Tracking configuration changes and compare configurations between example test environment and production environment.





---

---

## Discovering Middleware Targets

In order to manage and monitor Fusion Middleware components such as WebLogic Server instances, services and clusters, as well as the Java EE applications that depend on them, Enterprise Manager Grid Control must first "discover" the Fusion Middleware Domain containing these components.

Once discovered, the domain and the components within it are added as "managed targets". Management Agents are then assigned to each target, enabling Enterprise Manager Grid Control to collect the data needed to monitor the target.

This chapter covers the following:

- [Discovering a WebLogic 9.x or 10.x Domain Via Grid Control](#)
- [Discovering a WebLogic Server 7.x or 8.x Domain Via Grid Control](#)
- [Discovering Multiple WebLogic Domains Via EMCLI](#)
- [Enabling Automatic Discovery of New Domain Members](#)

### 2.1 Discovering a WebLogic 9.x or 10.x Domain Via Grid Control

Oracle WebLogic Server release 9.x and 10.x domains and their respective components can be discovered via Enterprise Manager Grid Control. A wizard guides you through the discovery process.

1. Log into Enterprise Manager Grid Control.
2. Click the **Targets** tab, then click the **Middleware** tab.
3. Select **Oracle Fusion Middleware/WebLogic Domain** from the Add menu, then click **Go**.
4. Enter the hostname for the WebLogic Administration Server for the domain in the Administration Server Host field. For example: `myhost06.us.oracle.com`
5. Enter the WebLogic Administration Server port. The default value is 7001.
6. Enter the WebLogic Administration Server username and password.

If you want to discover the target only for monitoring purposes, then it is sufficient to provide a user name that has monitoring role. If you want to monitor the target and also perform start/stop operations, then ensure that you provide a user name that has either an operator role or an admin role.

7. Select or specify the hostname and port of the Management Agent that will monitor the domain. An 11.1 Management Agent is required.
8. Specify a Unique Domain Identifier. This value is used as a prefix to ensure farm names are unique in environments with the same domain name. By default,

Enterprise Manager will name the farm "Farm", followed by a two-digit number, such as, "Farm01".

9. If the target domain is secured, expand the Advanced node to specify the protocol to use in the Protocol field. The default value is t3.

For additional details on discovering a domain secured using the Secure Sockets Layer (SSL) protocol, see section "C" in My Oracle Support Note 1093655.1. You can access My Oracle Support at the following URL:

<https://support.oracle.com/CSP/ui/flash.html>

10. Optionally supply the Java Management Extensions (JMX) Service URL that will be used to establish a JMX connection to the WebLogic Administration Server. For example:

```
service:jmx:t3://shutter.company.com:5555/jndi/weblogic.management.mbeanservers.domainruntime
```

If you do not supply a value, Enterprise Manager will provide the Service URL based on the host port and protocol.

11. Optionally enter any system properties to be used by the Java process to connect to the WebLogic Administration Server in the Parameters field.

Supply space-separated name/value pairs. Preface each parameter with -D. For example:

```
-Dparam1=xxx -Dparam2=yyy -Dparam3=zzz
```

12. Click **Continue**. Enterprise Manager will discover all Fusion Middleware targets within the domain.
13. Click **Close** in the Finding Targets dialog to automatically assign Management Agents to the discovered targets.

The Assign Agents page lists each Fusion Middleware target discovered and the Management Agent assigned to each. Agents are automatically assigned as follows:

- If a local Agent can be found on the domain host, that Agent will be assigned.
- If a local Agent cannot be found, the Agent specified in the Targets page will be assigned.

Note that you can also manually assign Agents to specific targets, if desired.

14. Click **Add Targets** to assign Management Agents as listed in the Assign Agents page.

The Saving Target to Agent processing window appears, indicating how many total targets have been added and successfully saved. It will also indicate the number of targets were unsuccessfully added.

15. Click **Close** in the processing window when finished. The Results page displays the targets and Agent assignments.
16. Click **OK** when finished. There may be a delay before these targets are visible and monitored. All the agents used for monitoring the targets must be up

## 2.2 Discovering a WebLogic Server 7.x or 8.x Domain Via Grid Control

Oracle WebLogic Server release 7.x and 8.x domains and their respective components can be discovered via Enterprise Manager Grid Control. A wizard guides you through the target discovery process.

1. Log into Enterprise Manager Grid Control.
2. Click the **Targets** tab, then click the **Middleware** tab.
3. Select **Oracle WebLogic Domain 7.x and 8.x** from the Add menu, then click **Go**.
4. Enter the hostname for the WebLogic Administration Server for the domain in the Administration Server Host field. For example: `myhost06.us.oracle.com`
5. Enter the WebLogic Administration Server port.
6. If the WebLogic Administration Server is secured using the Secure Sockets Layer (SSL) protocol, specify the location of the trusted keystore file. The keystore is a protected database that holds keys and certificates for an enterprise.

For additional details on discovering a domain secured with SSL, see section "C" in My Oracle Support Note 1093655.1. You can access My Oracle Support at the following URL:

<https://support.oracle.com/CSP/ui/flash.html>

7. Supply the WebLogic Administration Server username and password.

If you want to discover the target only for monitoring purposes, then it is sufficient to provide a user name that has monitoring role. If you want to monitor the target and also perform start/stop operations, then ensure that you provide a user name that has either an operator role or an admin role.

8. Supply or select the hostname for the Management Agent that will monitor the domain.

If the Management Agent is not running on the same host as the Administration Server, then specify the hostname for the host the Management Agent is running on.

If the Management Agent is running on the same host as the Administration Server, then set the BEA\_HOME environment variable for the Management Agent user, and also ensure that the user has access to the BEA home directory.

9. Optionally specify the absolute path to the directory where the `wljmxclient.jar` and `wlclient.jar` files are located as the value for Administration Server Home Directory. For example: `<WL-HOME>/server/lib`

If the Management Agent is running on a different host than the host where Oracle WebLogic Server is running, copy the JAR files from the Oracle WebLogic Server host (which should be `<WL-HOME>/server/lib`) and place them in a directory on the Agent host. Then provide the path to that directory.

10. Click **Next**.
11. Supply the credentials of the operating system user who installed the Oracle WebLogic Administration Server on the Management host specified in the previous step. If the Administration Server and Management host are on different hosts, then specify the operating system credentials. These credentials will be used by the Management Agent to discover domains.
12. Click **Next**.
13. Select the Oracle WebLogic Server Domain that you want Grid Control to monitor.

14. Click **Next**. Enterprise Manager will discover the WebLogic Server targets within the domain.
15. Select the target(s) you want to add.  
 Note that only active WebLogic Clusters, Admin Servers, Managed Servers, and Oracle Service Bus instances that are up and running are discovered and shown. There may be other Clusters, Admin Servers, Managed Servers, and OSBs in the domain, but if they are not running, then they are not shown in this page.  
 If you want to add non-running targets later, go to the Oracle WebLogic Server Domain Home page and click **Refresh Domain**.
16. Click **Next**.
17. Review the target details, then click **Next** to finish.

## 2.3 Discovering Multiple WebLogic Domains Via EMCLI

If you have multiple WebLogic domains that you want to manage through Enterprise Manager Grid Control, you can use the Enterprise Manager Command Line Interface (EMCLI) `discover_wls` verb to discover them all at once, rather than discovering them one at a time using the discovery wizards.

The `discover_wls` verb can be used to discover WebLogic Server versions 7.x, 8.x, 9.x, and 10.x domains. The verb reads a file named `domain_discovery_file` that contains the information required to discover each domain.

See the *Enterprise Manager Command Line Interface* book for instructions on using the `discover_wls` verb.

## 2.4 Enabling Automatic Discovery of New Domain Members

In the typical enterprise, Oracle WebLogic domains do not remain static. Instead, membership in the domain changes regularly: new Java EE applications are deployed, WebLogic Service instances are created or removed, clusters are added, and so on.

By default, Enterprise Manger Grid Control is not automatically aware of changes made to Oracle WebLogic domains that have been configured as managed targets. However, you can enable a pre-defined Grid Control job named "WebLogic Domain Refresh" to automatically discover new domain members and add them as managed targets.

Note that targets that have been removed from the domain - such as Java EE applications that have been un-deployed - are NOT automatically removed as managed targets. Because historical performance and configuration data is stored in the Oracle Management Repository for all managed targets, targets remain unless they are explicitly removed by selecting **Target Setup>Remove Target** from the **WebLogic Domain** menu.

1. Log into Enterprise Manager Grid Control.
2. Click the **Targets** tab, then click the **Middleware** tab.
3. Click on the WebLogic Domain you want to enable the job for in the Middleware home page.
4. In the General region of the page, click the timestamp link next to the **WebLogic Domain Refreshed** property. The Refresh WebLogic Domain dialog opens.
5. Check the **Enable Automatic Refresh** box in the Refresh WebLogic Domain dialog, then click **OK**.

Once enabled, the job will check for new domain members once every 24 hours by default. To change the job settings, including the frequency at which it is run:

1. Click the **Jobs** tab.
2. Click the job title in the Job Activity page.
3. Click **Edit**.



---

---

## Monitoring Middleware Targets

This chapter describes how you can use Enterprise Manager to monitor middleware software.

This chapter covers the following:

- [Monitoring Middleware Targets in Oracle Enterprise Manager Grid Control](#)
- [Out-of-Box Monitoring of Middleware Targets](#)
- [Diagnosing Performance Problems](#)
- [Administering Middleware Targets](#)
- [Managing Configurations](#)
- [Managing Service Levels](#)
- [Automating Operations](#)

### 3.1 Monitoring Middleware Targets in Oracle Enterprise Manager Grid Control

To add middleware related targets (e.g. Oracle Fusion Middleware 11g, Oracle WebLogic Domain, Oracle Application Server 10g, JBoss Application Server) to Enterprise Manager Grid Control, follow these steps:

After you have added the target, you can view general information about the targets including their status and availability on the Middleware page. You can drill down into each target to get further details like how the target is performing, where it is deployed, the version, location of its home directory, and so on.

You can also view the number of critical, warning, and error alerts generated for the past 24 hours. These alerts indicate that a particular metric condition has been encountered. For example, an alert is triggered when a metric threshold is reached. Using these details, you can drill down to investigate the target and the problem that triggered the alert.

You can monitor the following components using Oracle Enterprise Manager Grid Control:

- Oracle Fusion Middleware Components
- Oracle Application Server Components
- Non-Oracle Middleware Components

### 3.1.1 Oracle Fusion Middleware Components

You can monitor the following Oracle Fusion Middleware components using Enterprise Manager Grid Control:

- **Oracle WebLogic Server Domains, Clusters, and Managed Servers:** A WebLogic Server domain is a logically related group of WebLogic Server resources that you manage as a unit. A domain includes one or more WebLogic Servers and may also include WebLogic Server clusters. Clusters are groups of WebLogic Servers instances that work together to provide scalability and high-availability for applications. With Oracle Enterprise Manager Grid Control, you can monitor and manage the farm, domains, clusters, servers, and deployed applications.
- **Oracle SOA Suite:** The Oracle SOA Suite enables services to be created, managed, and orchestrated into SOA composite applications. Composite applications enable you to easily assemble multiple technology components into one SOA composite application. Oracle SOA Suite plugs into heterogeneous infrastructures and enables enterprises to incrementally adopt SOA. You can:
  - Automatically discover and model SOA components such as BPEL Process Manager, Oracle Service Bus, Service Engines etc.
  - Monitor the health and performance of the SOA components.
  - Trace the flow of an instance across all SOA Infrastructure applications.
  - Create systems, services, and aggregate services.
- **Oracle WebCenter:** The Oracle WebCenter is an integrated set of components with which you can create social applications, enterprise portals, collaborative communities, and composite applications, built on a standards-based, service-oriented architecture. It combines dynamic user interface technologies with which to develop rich internet applications, the flexibility and power of an integrated, multichannel portal framework, and a set of horizontal Enterprise 2.0 capabilities delivered as services that provide content, collaboration, presence, and social networking capabilities. Based on these components, Oracle WebCenter also provides an out-of-the-box, enterprise-ready customizable application, WebCenter Spaces, with a configurable work environment that enables individuals and groups to work and collaborate more effectively.
- **Oracle WebTier:** This consists of:
  - **Oracle HTTP Server:** Oracle HTTP Server (OHS) is the underlying deployment platform for all programming languages and technologies that Oracle Fusion Middleware supports. It provides a Web listener and the framework for hosting static and dynamic pages and applications over the Web. Based on the proven technology of the Apache 2.2.10 infrastructure, OHS includes significant enhancements that facilitate load balancing, administration, and configuration. It also includes a number of enhanced modules, or mods, which are extensions to the HTTP server that extend its functionality for other enterprise applications and services. You can:
    - \* Automatically discover and monitor Oracle HTTP Servers running within the application servers.
    - \* View a list of metrics to gauge the server performance and virtual host performance.
    - \* View the top URLs being accessed.
    - \* Perform the enterprise configuration management tasks like viewing, comparing, and searching configuration information.



- **Oracle Web Cache:** Oracle Web Cache is a content-aware server accelerator, or reverse proxy, for the Web tier that improves the performance, scalability, and availability of Web sites that run on any Web server or application server, such as Oracle HTTP Server and Oracle WebLogic Server. Oracle Web Cache is the primary caching mechanism provided with Oracle Fusion Middleware. Caching improves the performance, scalability, and availability of Web sites that run on Oracle Fusion Middleware by storing frequently accessed URLs in memory. You can:
  - \* Automatically discover and monitor OracleAS Web Cache instances running within application servers.
  - \* View the metrics associated with this target to analyze their performance.
  - \* Perform enterprise configuration tasks like viewing, comparing, and searching configuration information.
- **Oracle Identity Management:** This is an enterprise identity management system that automatically manages users' access privileges within the resources of an enterprise. It provides a shared infrastructure for all Oracle applications. It also provides services and interfaces that facilitate third-party enterprise application development. These interfaces are useful for application developers who need to incorporate identity management into their applications.
- **Oracle Portal:** This is a Web-based tool for building and deploying e-business portals. It provides a secure, manageable environment for accessing and interacting with enterprise software services and information resources. A portal page makes data from multiple sources accessible from a single location.
- **Oracle Forms Services** is a middle-tier application framework for deploying complex, transactional forms applications to a network such as an intranet or the Internet. With Oracle Forms Services, business application developers can quickly build comprehensive Java client applications that are optimized for the Internet without writing any Java code, and that meet (and exceed) the requirements of professional user communities. These Java client applications are Web-deployed applications available on demand for rapid processing of large amounts of data and rapid completion of complex calculations, analysis, and transactions.
- **Oracle Coherence** is a component of Oracle Fusion Middleware that enables organizations to predictably scale mission-critical applications by providing fast and reliable access to frequently used data. By automatically and dynamically partitioning data in memory across multiple servers, Oracle Coherence enables continuous data availability and transactional integrity, even in the event of a server failure. As a shared infrastructure, Oracle Coherence combines data locality with local processing power to perform real-time data analysis, in-memory grid computations, and parallel transaction and event processing. Oracle Coherence comes in three editions. You can:
  - Discover and manage a Coherence cluster and its various entities.
  - Monitor and configure various components such as nodes, caches, services, connections, and connection manager instances of a Coherence cluster.
  - Deploy and install a Coherence node based on the Provisioning Advisory framework.
- **Oracle Business Intelligence** is a complete, integrated solution that addresses business intelligence requirements. Oracle Business Intelligence includes Oracle Business Intelligence Reporting and Publishing, Oracle Business Intelligence Discoverer, and Oracle Business Intelligence Publisher. You can:

- Manually discover Oracle BI Suite EE targets, and monitor their overall health.
- Diagnose, notify, and correct performance and availability problems in Oracle BI Suite EE targets.
- Access current and historical performance information using graphs and reports.
- Perform enterprise configuration management tasks like viewing, comparing, and searching configuration information.

### 3.1.2 Oracle Application Server Components

You can monitor Oracle Application Server 10g components like Oracle Application Server Farms, Oracle Application Server Clusters, Oracle Application Servers, OC4J, Oracle HTTP Servers, Oracle Web Cache, Oracle Portal, Oracle Wireless, Oracle Forms Services, Oracle Reports Services, Oracle Business Intelligence, and Oracle Identity Management.

### 3.1.3 Non-Oracle Middleware Components

In addition to monitoring Oracle middleware components, Enterprise Manager Grid Control can also be used to monitor non-Oracle middleware software. The third-party middleware software that can be monitored includes the following:

- WebSphere Application Server
- WebSphere MQ
- JBoss Application Server
- Apache Tomcat
- Apache HTTP Server
- Microsoft Exchange Server
- Microsoft Internet Information Services
- Microsoft Active Directory
- Microsoft Commerce Server
- Microsoft BizTalk Server
- Microsoft Internet Security and Acceleration
- Microsoft .NET Framework

## 3.2 Out-of-Box Monitoring of Middleware Targets

Enterprise Manager Grid Control organizes a wide variety of performance data and administrative functions into distinct, Web-based home pages for the farm, domain, servers, components, and applications.

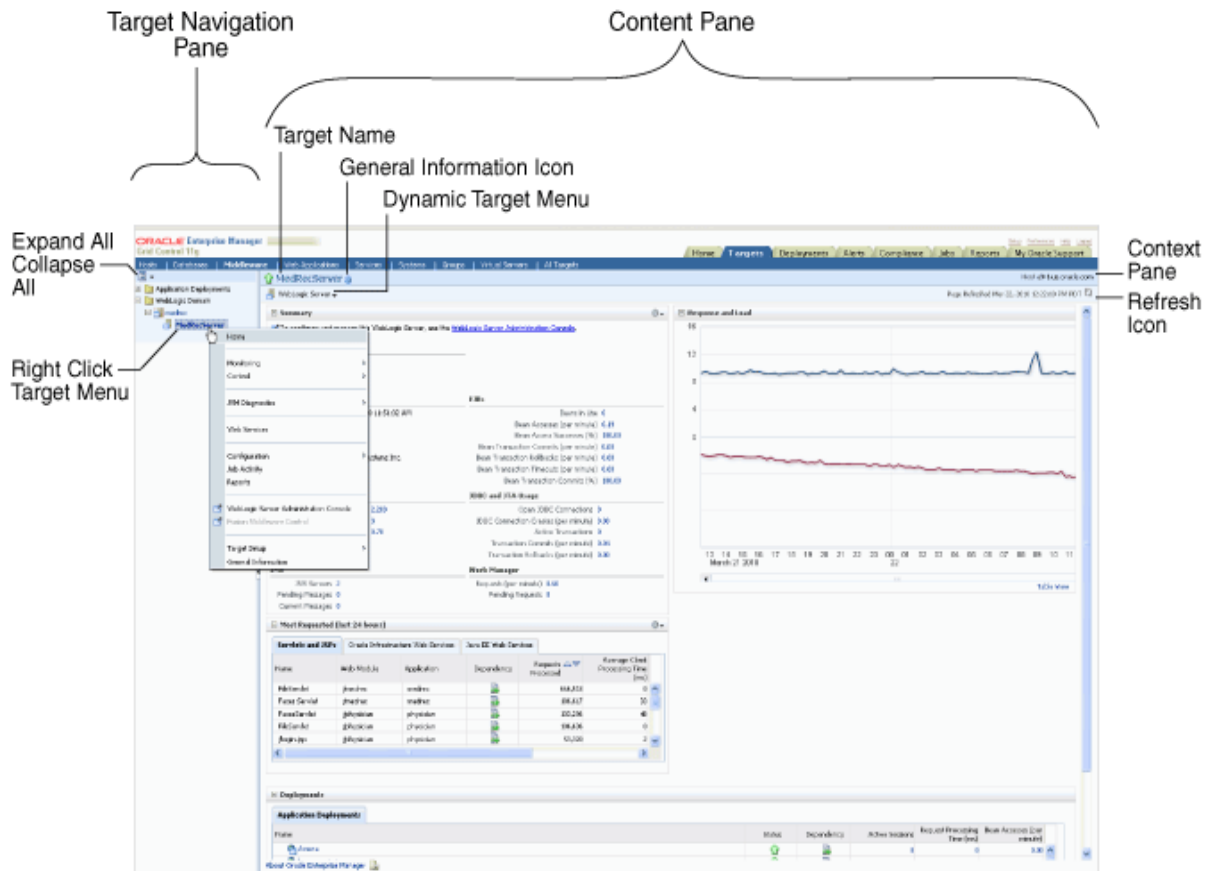
### 3.2.1 Target Home Page

Enterprise Manager Grid Control provides centralized monitoring across domains, configuration management, provisioning, real time and historical performance analysis. You need to drill down to Oracle Fusion Middleware Control to perform administrative tasks and manage components in your farm.

The Home pages make it easy to locate the most important monitoring data and the most commonly used administrative functions—all from your Web browser.

When you login into Enterprise Manager Grid Control and select a Middleware target, the Home page for the target is displayed. For example, when you click on a WebLogic Server target in the Middleware page, the following screen is displayed.

**Figure 3–1 WebLogic Server Home Page**



This figure shows the target navigation pane on the left and the content page on the right. From the target navigation pane, you can expand the tree and select a component or an application.

When you select a target, the target's home page is displayed in the content pane and that target's menu is displayed at the top of the page, in the context pane. You can also view the menu for a target by right-clicking the target in the navigation pane.

In the preceding figure, the following items are called out:

- **Target Navigation Pane** lists all of the targets in a navigation tree
- **Content Pane** shows the current page for the target. When you first select a target, that target's home page is displayed.
- **Dynamic Target Menu** provides a list of operations that you can perform on the currently selected target. The menu that is displayed depends on the target you select. The menu for a specific target contains the same operations as those in the Right-Click Target Menu.

- **Right-Click Target Menu** provides a list of operations that you can perform on the currently selected target. The menu is displayed when you right-click the target name in the target navigation pane. In the figure, even though the WebLogic Server is selected and its home page is displayed, the right-click target menu displays the operations for the MedRecServer because the user has right-clicked the MedRecServer.
- **Target Name** is the name of the currently selected target.
- **Context Pane** provides the host name, the time of the last page refresh, and the Refresh icon.
- **Expand All / Collapse All** lets you expand or collapse the navigation tree.
- **Refresh** icon indicates when the page is being refreshed. Click it to refresh a page with new data. (Refreshing the browser window refreshes the page but does not retrieve new data.)

From the Home page, you can also access the Fusion Middleware Control and WebLogic Server Administration Console by clicking on the appropriate link or selecting the appropriate menu item on the page.

### 3.2.2 Out-of-box Metrics

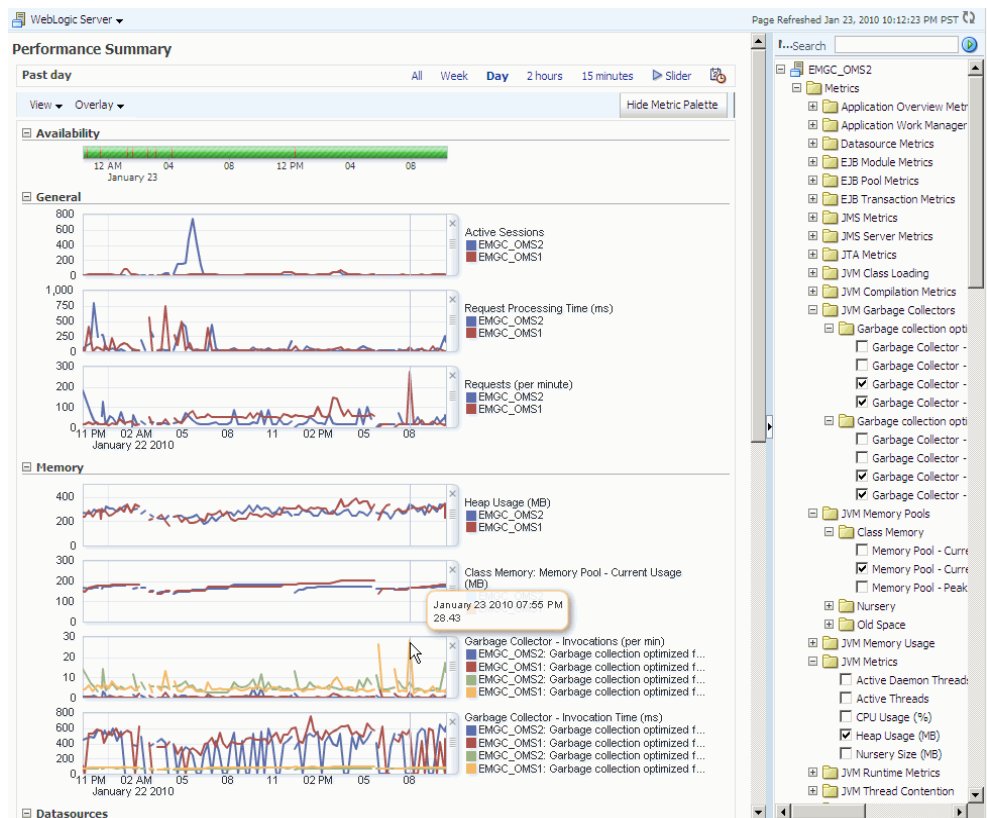
Enterprise Manager Grid Control provides a set of pre-defined performance metrics for each middleware target. These metrics are available for domains, cluster, server, applications, web services, resources etc. The metric data is collected and stored in the Management Repository. For more details on the pre-defined metrics, see the *Oracle Fusion Middleware Metric Reference Guide*.

For example, Enterprise Manager Grid Control can automatically monitor:

- The CPU or memory consumption of the application server, including detailed monitoring of individual Java Virtual Machines (JVMs) being run by Oracle WebLogic servers.
- Java EE application responsiveness from the application down through individual servlets and Enterprise JavaBeans (EJBs)
- Oracle HTTP Server session volumes, connection duration, and error rates
- Oracle Web Cache hit rates and volumes
- Top servlets based on number of requests, maximum processing time, and highest average processing time

The performance metrics provide details about the metric as a current real time value (30 seconds, 1 minute, or 5 minutes) or a previous value (past 24 hours, 7 days, or 31 days). The historical information is displayed as graphs and a table. By using graphs, you can easily watch for trends, and by using tables, you can examine details of past metric severity history. The out-of-metrics can be viewed from the performance summary pages as shown below:

Figure 3–2 Performance Summary Page



You can customize the charts by rearranging the order of the charts, removing a chart from the page, modify the scale of a chart, display the metrics in a hierarchical tree format, etc.

### 3.2.3 Analyzing Historical Performance

Enterprise Manager Grid Control allows you to analyze historic metric data and perform trend analysis. You cannot view or analyze historical performance in Fusion Middleware Control. The metrics are collected and stored in the Management Repository, so you can analyze the data well after the situation has changed. For example, you can use historical data and diagnostic reports to research an application performance problem that occurred days or even weeks ago.

You can even provide a customized time period for which the data should be retrieved from the Management Repository. You can customize the time period for:

- Pre-defined range of the last 24 hours, last 7 days, or last 31 days
- Customized range of any number of days, weeks, months, or years
- Any start date and end date (such that the duration is not greater than 99 years)

---

**Note:** You can analyze historical metric data with Enterprise Manager Grid Control only. You cannot use Fusion Middleware Control to analyze historic performance.

---

### 3.2.4 Setting Metric Thresholds for Alert Notifications

Metric snapshots are named snapshots of a target's past performance. You can use metric snapshots to calculate thresholds based on deviations from this past performance. Thresholds are boundary values against which monitored metric values are compared. You can specify a threshold such that, when a monitored metric value crosses that threshold, an alert is generated. You can get critical alerts when a monitored metric has crossed its critical threshold or warning alerts when a monitored metric has crossed its warning threshold.

Enterprise Manager Grid Control provides a comprehensive set of features that facilitates automated monitoring and generation of alerts. You can gather and evaluate diagnostic information for targets distributed across the enterprise, and an extensive array of middleware performance metrics are automatically monitored against predefined thresholds. By selecting a metric, you can determine whether the thresholds have been defined for a particular metric. These thresholds are used as a mechanism to generate alerts. These alerts in turn are used to notify you whether a target is up or down, the response time is slow, and so on. Thus, you can monitor their overall performance.

You can set up corrective actions to automatically resolve an alert condition. These corrective actions ensure that routine responses to alerts are automatically executed, thereby saving you time and ensuring that problems are dealt with before they noticeably impact the users.

### 3.2.5 Monitoring Templates

You can also use monitoring templates to simplify the task of standardizing monitoring settings across your enterprise. You can specify the monitoring settings once and apply them to all Oracle Fusion Middleware targets. A Monitoring template defines all the parameters you would normally set to monitor Middleware target, such as:

- Target type to which the template applies
- Metrics (including user-defined metrics), thresholds, metric collection schedules, and corrective actions

When a change is made to a template, you can reapply the template across the affected targets in order to propagate the new changes. You can reapply monitoring templates as often as needed.

### 3.2.6 Managing and Creating Blackouts

Enterprise Manager Grid Control comes with a bundle of performance and health metrics that enable automated monitoring of application targets in your environment. When a metric reaches the predefined warning or critical threshold, an alert is generated and the administrator is notified.

However, there are occasions when you want to perform maintenance work on your middleware targets, and not want any alerts to be generated while you are bringing them down. In this case, you can schedule a blackout and suspend monitoring of the middleware targets.

Blackouts allow you to suspend any data collection activity on one or more monitored targets, thus allowing you to perform scheduled maintenance on targets. If you continue monitoring during these periods, the collected data will show trends and other monitoring information that are not the result of normal day-to-day operations. To get a more accurate, long-term picture of a target's performance, you can use

blackouts to exclude these special-case situations from data analysis. Enterprise Manager Grid Control allows you to define new blackouts; view the status of existing blackouts; and edit, stop, and delete blackouts that are not required.

### 3.2.7 Extend Monitoring for Applications Deployed to WebLogic Server

Many administrators often require custom logic to be written to check for conditions specific to their application environments. Enterprise Manager Grid Control allows integration of application instrumentation in Enterprise Manager Grid Control's event monitoring infrastructure. If application developers expose application instrumentation using standards like JMX or Web Services operations, then you can build management plug-ins for the instrumentation using easy-to-use command line tools, and leverage Enterprise Manager Grid Control's event monitoring system to monitor it. You do not have to edit any XML files or write any integration code to integrate such instrumentation. Follow these procedures to integrate application-defined instrumentation:

- Use Command Line Interfaces that analyze MBean interfaces for JMX and WSDL for Web Services and create management plug-ins
- Import Management Plug-in Archive in Enterprise Manager Grid Control
- Deploy Management Plug-in to Management Agents
- Create Target-type instances for the target types defined in Management Plug-in Archive
- Leverage Enterprise Manager Grid Control's event monitoring system including monitoring templates, corrective actions, historical and real time metric views, alerts, customization of notification rules, and methods on events generated from application instrumentation metrics.

Beginning with Enterprise Manager Grid Control release 11.1, administrators are able to add performance metrics beyond those available out-of-box for JMX-instrumented applications deployed on Oracle WebLogic Server. Administrators can additionally monitor JMX-enabled applications by defining new target type that can be monitored via management plug-ins, and then use a command line tool `emjmxcli` to automate the generation of the target metadata and collection files.

All JMX-enabled applications deployed to the WebLogic Server can be consolidated and monitored by a single management tool, Enterprise Manager Grid Control.

### 3.2.8 Request Monitoring

Request Monitoring provides end-to-end visibility into requests and helps localize end-user performance problems based on the deployment model. You can monitor, understand, and visualize how servers interact with each other to deliver business end-user services requests. You can trace important end-user requests from the client to the service endpoint across all the servers and applications associated with each transaction. The trace data is used to model the call-path for these requests where call-paths represent the inter-server relationship and performance metrics for these requests. You can only monitor synchronous transactions running on WebLogic servers. With Request Monitoring, you can:

- Trace end-user transactions and capture the complete call-path of important requests.
- Reduce problem localization by diagnosing poorly performing requests and identifying the servers whose behavior (service time) deviates most from the norm.

- Reduce fault discovery time and set performance level objectives on request response times.
- Reduce fault-reason identification time by allowing the user to launch into Oracle Enterprise Manager JVM Diagnostics feature. This allows performance diagnostics to be done within the context of the problem experienced as well as historical statistics of the system when it was behaving normally.

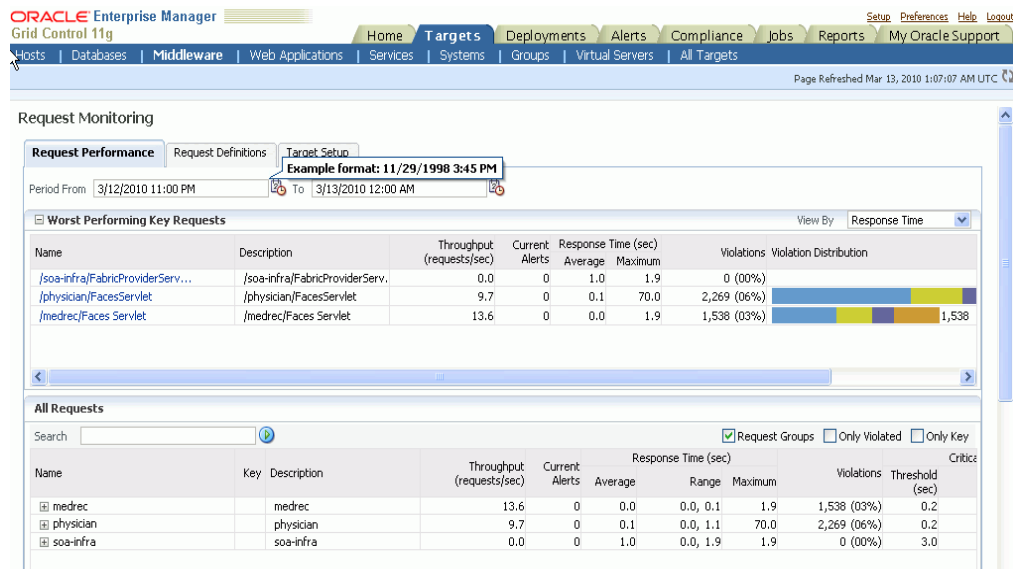
### 3.2.8.1 Defining and Managing Transactions Being Monitored

A request is a server entry-point that can be invoked by a Web Browser or an application. A group of request instances related to each other by some common attributes is known as a request. A collection of requests is known as a transaction group. A transaction can belong to one or more transaction groups. Violations occur when the critical and warning thresholds are exceeded. The administrator can mark one or more important requests as key requests. Alerts are generated for key requests only.

You can define requests that need to be monitored and tracked and group them for reporting purposes. A set of pre-defined requests and request groups are provided by default. You can create a new request, edit or delete an existing transaction. To define a request:

1. Log into Oracle Enterprise Manager and click the **Middleware** tab.
2. All the Middleware targets are displayed. Under the Related Links, click **Request Monitoring**.
3. Click **Request Performance** to view the performance of the request during a specific time period.

**Figure 3–3 Request Performance Page**

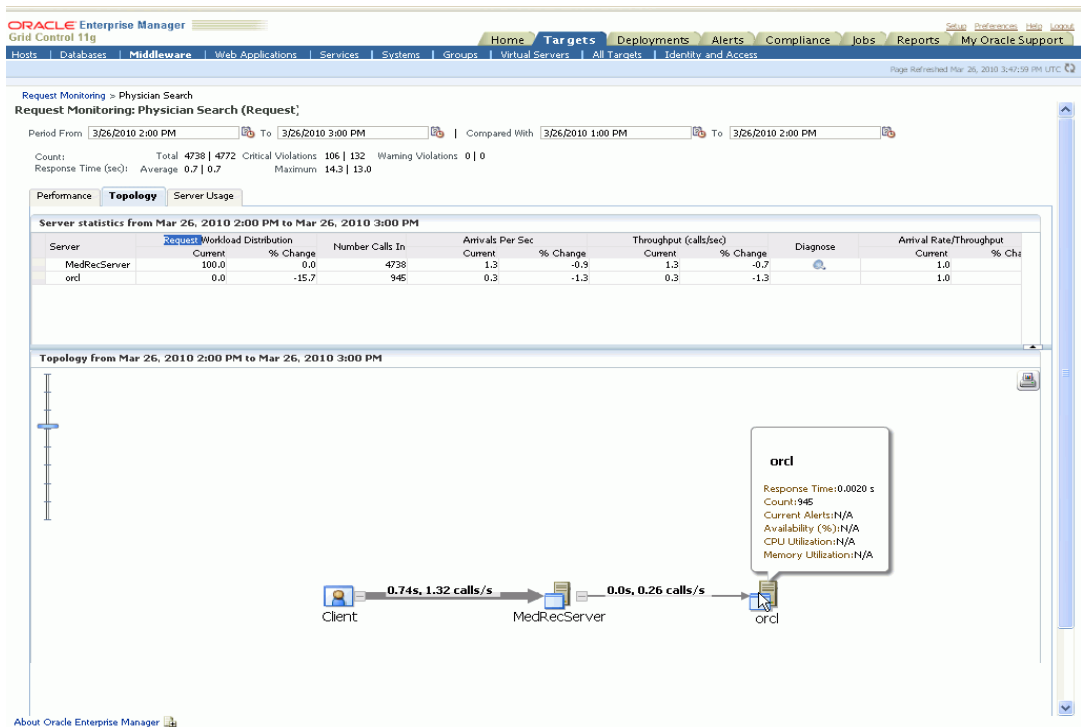


4. Click on a request to drill down to the Performance tab. The performance charts show the volume of requests over a time period and the average response time of all requests during this period. Two performance charts are displayed for the current and the comparison period. The charts provide a historic response time analysis of the request over the selected period of time.



5. Click on the Topology tab. The following screen is displayed.

**Figure 3–4 Topology Page**



The topology chart shows the aggregate server level call paths of the request. The chart shows the calls made between the clients and the clusters or servers. Click on the icon on a cluster to view all the servers that are part of the cluster. Place the cursor on a cluster or server to view the following details:

- **Response Time:** The average response time taken by the server to process all the calls for the selected period.
- **Count:** The total number of calls that have been made to the server.
- **Current Alerts:** These alerts show the status of the server or cluster and indicate whether it is up or down, blacked out, or unreachable.

6. Click on the Server Usage tab. The following screen is displayed.

Figure 3–5 Server Usage Page



These charts show the workload distribution for the selected request in each server. The charts display the average of the percentage contribution of each server or cluster in the selected time period. You can see the workload distribution for good requests, requests that have violated the warning thresholds, and requests that have violated the critical thresholds.

### 3.3 Diagnosing Performance Problems

This section describes the methods and tools used to diagnose performance problems. You can:

- View the list of most active Servlets and JSPs and identify the ones that are causing the bottleneck.
- Analyze Java EE and SOA applications using Application Dependency and Performance.
- Use Java Diagnostics to diagnose performance problems in production.

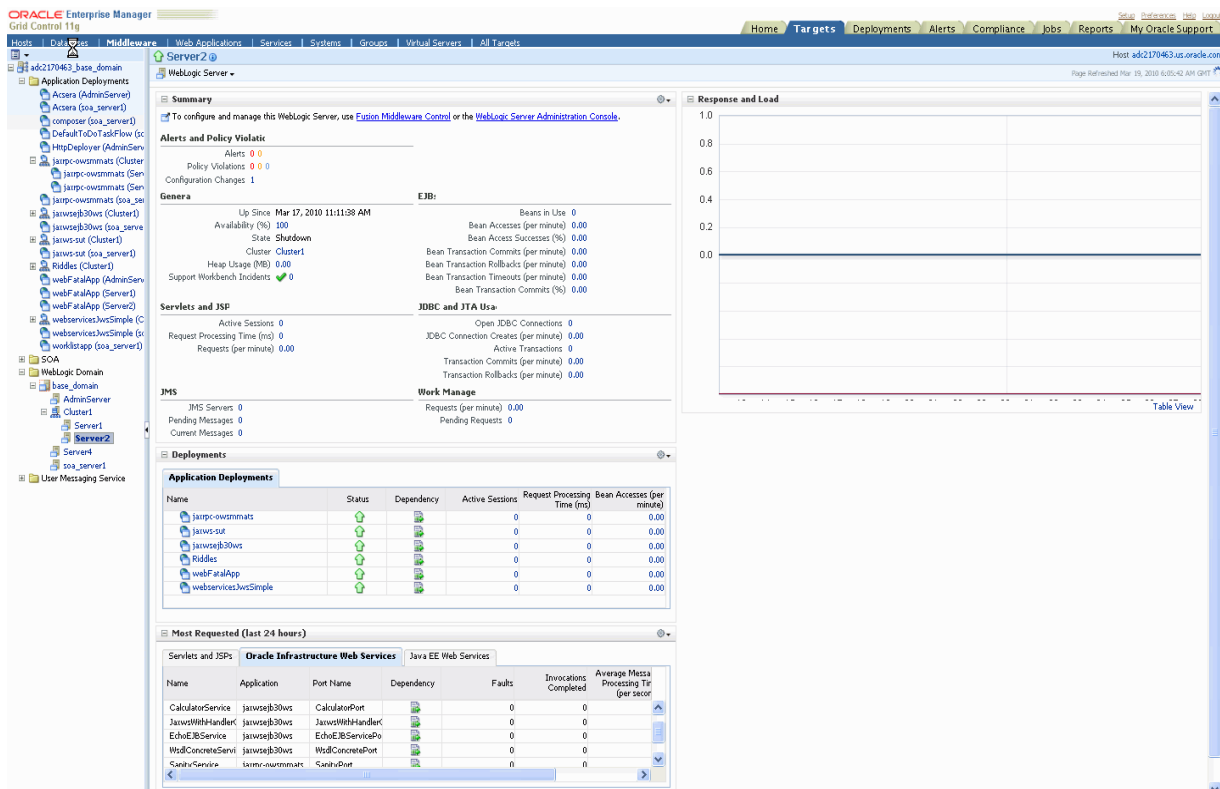
#### 3.3.1 Using Home Pages to Diagnose Performance Issues

When you are troubleshooting performance problems, it can be helpful to know which servlets or JSPs are the most active. By viewing the Most Requested section on the WebLogic Server Home page, you can identify the most active Java servlets, JSPs, Web Services, or Java EE Services running on the WebLogic Server instance.

When you receive an alert notification, Enterprise Manager Grid Control makes it easy for you to investigate the problem and take corrective actions wherever required. For example, notification of excessive CPU consumption by WebLogic Server may lead to investigation of the applications running on that instance. By using the Servlets and

JSPs tab in the Most Requested section of the WebLogic Server Home page, you can quickly identify the highest volume or least responsive application. You can then drill down and diagnose application's servlets, Java Server Pages (JSPs), or EJBs to identify the bottleneck.

**Figure 3–6 WebLogic Server Home Page (Most Requested)**



### 3.4 Administering Middleware Targets

Enterprise Manager Grid Control allows you to monitor multiple domains and provides configuration management, diagnostics, automation, and historical performance analysis. From Enterprise Manager Grid Control, you can drill down to Oracle Fusion Middleware Control or Oracle WebLogic Administration Console to monitor and administer your Oracle Fusion Middleware environment

- **Oracle Fusion Middleware Control:** Fusion Middleware Control is a Web browser-based, graphical user interface that you can use to monitor and administer a farm. Fusion Middleware Control organizes a wide variety of performance data and administrative functions into distinct, Web-based home pages for the farm, domain, servers, components, and applications. With Fusion Middleware Control, you can perform various tasks like managing the SOA environment, deploying ADF applications, managing Fusion Middleware components, etc.
- **Oracle WebLogic Server Administration Console:** Oracle WebLogic Server Administration Console is a Web browser-based, graphical user interface that you use to manage a WebLogic Server domain. It is accessible from any supported Web browser with network access to the Administration Server. With the WebLogic

Server Administration Console, you can perform various tasks like managing the WebLogic Server, managing data sources, JMS resources, etc.

### 3.4.1 Process Control

Enterprise Manager Grid Control allows you to perform process control tasks such as starting, stopping, or restarting middleware targets. You can perform these tasks by selecting the Control option in the menu from the Home page of a target. You can also schedule a job to perform these operations. For example, for a WebLogic Server Domain, you can create a job to automatically start, stop, or restart the servers in the domain. You can also view details about the jobs that are scheduled, running, suspended, or the ones that have a problem.

### 3.4.2 Creating WebLogic Domain Groups

The WebLogic Domain Group is a custom group that can have WebLogic Domains as members. You can navigate to the page where you can create a WebLogic Domain Group from the All Targets page or the Groups page. A WebLogic Server Domain includes a special WebLogic Server instance called the Administration Server, which is the central point from which you configure and manage all resources in the domain. It also typically includes one or more WebLogic Server instances, called managed servers, on which Java EE applications and components run.

You can perform the following tasks on a WebLogic Domain Group:

- View response and load information of the top 5 servers by average response times.
- View details about the domain such as status, alerts and policy violations, and configuration changes.
- View the resource usage of the top 5 servers by CPU usage percentage.
- See useful performance data related to JMS/JDBC/EJBs/JSPS and servlets of all servers across all domains

## 3.5 Managing Configurations

Enterprise Manager Grid Control provides a suite of configuration management functions that can be performed on middleware targets.

Oracle Management Agent collects configuration information about Oracle Fusion Middleware targets from their respective configuration files, and communicates this information over HTTP/HTTPS to Oracle Management Service, which stores it in the Management Repository. This information is periodically collected and updated while maintaining the audit of changes. Enterprise Manager's configuration management capabilities efficiently guide the users to desired configuration data in a particular component.

You can compare these configuration details and view the differences and similarities between the two instances of a middleware target. You have the flexibility to compare two last collected configurations or two saved configuration files. You can also compare one configuration with multiple configurations or one configuration in the Management Repository with a saved configuration file.

Using Enterprise Manager Grid Control, you can search configurations across middleware targets and find configuration anomalies - whether they are a mismatch of an install/patch version of Oracle Fusion Middleware software, or they are a mismatch of the software configuration data. You can perform more intelligent

searches to identify all the components hosting a particular application or other resources.

In addition, for BPEL Process Manager targets, you can view the BPEL Processes, its different versions, and the suitcase files associated with each version. You can also compare the BPEL Process suitcase files of different versions and track the changes that were made to a version. This allows you to identify the cause for improved or deteriorated performance due to a change in the BPEL Process suitcase file.

### 3.5.1 Policy Violations

Enterprise Manager Grid Control also shows the number of informational, warning, and critical policy rules violated for the application server target. You can see the roll up of individual policy compliance scores for that target. Compliance scores provide a quick way to determine the health of the your middleware target. You can also see the policy trend overview for the last 24 hours, for the last week, for the last month, or for a user-define time period, and determine your course of action in solving the policy violations.

Enterprise Manager Grid Control also shows the time when security policy rules were last evaluated, and shows the roll up of individual policy compliance scores for that target.

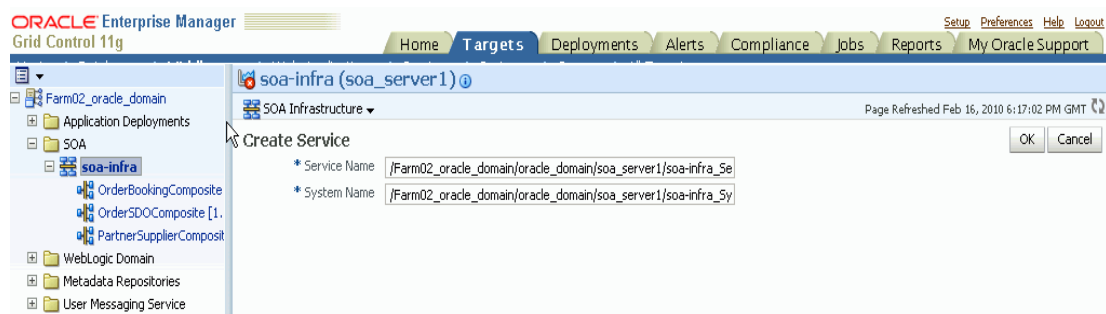
## 3.6 Managing Service Levels

Enterprise Manager Grid Control allows you to create infrastructure services for middleware targets such as Oracle BPEL Process Manager targets, Oracle Service Bus targets and Oracle SOA Composite and SOA Infrastructure instances.

An infrastructure service is a dependency service that is created to identify the infrastructure components on which the middleware target depends. Here, the infrastructure components refer to hosts, databases, application servers, and so on that work together to host the middleware target.

You can either create an infrastructure service with a new system or an existing system, or simply refresh an existing infrastructure service, if there is already one existing. By creating infrastructure services and systems, you can better manage your middleware targets and also the components on which the middleware targets depend.

**Figure 3–7 Creating Infrastructure Service for SOA Infrastructure**



For example, once you create an infrastructure service for an Oracle SOA Infrastructure target, Grid Control allows you to create an aggregate service for every

process within that SOA Infrastructure target. An aggregate service is a logical grouping of services, in this case, infrastructure services and availability services. Aggregate Services give you a bird's-eye view of the services that have been created for the SOA Infrastructure target and helps you monitor their availability, performance, and usage. Service availability can be composed of both metrics on the underlying target and service test results from period synthetic transaction execution.

You can define service level (measure of service quality) for a service. A service level is defined as the percentage of time during business hours a service meets specified availability, performance and business criteria.

A Service Level specifies the percentage of time a service meets the performance and availability criteria as defined in the Service Level Rule. By default, a service is expected to meet the specified criteria 85% of the time during defined business hours. You may raise or lower this percentage level according to service expectations. A service level measures service quality using two parameters: Expected and Actual Service Levels.

- **Expected Service Level:** A Service Level specifies the percentage of time a service meets the performance and availability criteria as defined in the Service Level Rule. By default, a service is expected to meet the specified criteria 85% of the time during defined business hours. You may raise or lower this percentage level according to service expectations.
- **Actual Service Level:** The Actual Service Level defines the baseline criteria used to define service quality.

## 3.7 Automating Operations

The following topics are described in this section:

- Job System
- Provisioning
- Cloning and Patching the Middleware Environment

### 3.7.1 Job System

You can use Enterprise Manager Grid Control job system to schedule tasks you want to automate. You can schedule a job for a target by selecting the Control menu option on the Home page. For example, for an Oracle WebLogic Server Domain, you can create a job to automatically start, stop, or restart the servers in the domain.

You can view details about the jobs that are scheduled, running, suspended, or the ones that have a problem. You can also use jobs to automate the execution of the WLST (WebLogic Scripting Tool) scripts.

#### 3.7.1.1 Log File Rotation

Oracle Application Server components generate log files containing messages that record all types of events, including startup and shutdown information, errors, warning messages, access information on HTTP requests, and additional information.

However, the information recorded in log files is voluminous, thus making it difficult to track what update was made at what time. Also because of the huge quantity of information updated periodically, the log files grow in size and occupy more space on the system over a period of time. The only way to manage these log files is to manually archive the contents to another file and store them in a different location.

Considering these impediments, Enterprise Manager Grid Control has been enhanced with a log rotation feature that helps you manage the logs of Oracle Application Server components more effectively. In particular, you can use Enterprise Manager Grid Control to:

- Schedule a job that automatically rotates a log at the scheduled date and time
- Manage space on your system by storing the rotated log files in a different directory

Enterprise Manager Grid Control allows you to view the logs of a particular Oracle Application Server component type and select the ones that need to be rotated. Note that a log rotation job can also be part of a multi-task job.

When a log rotation job is executed, Enterprise Manager Grid Control automatically stops the component whose logs have to be rotated. After it is stopped, the content from its existing log file is moved to another file that is distinguished with the timestamp when it was actually rotated. The original log file is kept empty for new log details to be populated. Once this is done, Enterprise Manager Grid Control restarts the component.

---

---

**Note:** The log rotation feature cannot be used with the WebLogic Server.

---

---

## 3.7.2 Operations Automation

Rather than spend resources on manually installing and configuring Oracle Fusion Middleware software, administrators would rather spend time and money on more strategic initiatives. To help achieve this, Enterprise Manager Grid Control has automated common provisioning operations such as cloning Oracle SOA Suite 11g and scaling out an Oracle WebLogic Domain. Making such critical datacenter operations easy, efficient and scalable results in lower operational risk and lower cost of ownership. To access these provisioning operations, navigate to the Deployments page in Enterprise Manager Grid Control and click **Deployment Procedure** and select either of the following:

- Fusion Middleware Provisioning
- Fusion Middleware Domain Scale Up

For more details on using these procedures, see Enterprise Manager Online Help.

### 3.7.2.1 Cloning from Test to Production Environments

Typically, creating a new environment to support SOA applications entails several manual, error prone installation and configuration steps. With Oracle Enterprise Manager this can be accomplished with very little effort and time via a predefined, customizable deployment procedure. This deployment procedure clones an existing SOA Suite environment to a new set of hardware per a hierarchical series of steps. These predefined steps can be edited or disabled and new steps or custom scripts can be added to the deployment procedure to satisfy unique business needs. The deployment procedure also supports secure host authentication using super user do (sudo) or pluggable authentication modules (PAM). While running the deployment procedure, administrators can specify configuration settings such as the domain name, credentials for the administration console, port values, and JDBC data resources. After the procedure completes, the newly created SOA environment is discovered and automatically added to the console for centralized management and monitoring.

### **3.7.2.2 Scaling Out Domains**

To address growing business demands, modern data centers must augment and relocate resources quickly. Using Oracle Enterprise Manager, administrators can rapidly scale out a WebLogic Domain and Cluster with additional managed servers to accommodate an increase in application load.



# Part II

---

## Managing Oracle SOA

The chapters in this part describe how you can discover and monitor Oracle BPEL Process Manager, Oracle Service Bus, and Oracle SOA Suite 11g.

Part I contains the following chapters:

- [Chapter 4, "Overview of Oracle SOA Management Pack Enterprise Edition"](#)
- [Chapter 5, "Discovering and Monitoring Oracle BPEL Process Manager"](#)
- [Chapter 6, "Discovering and Monitoring Oracle Service Bus"](#)
- [Chapter 7, "Discovering and Monitoring SOA Suite 11g"](#)



## Overview of Oracle SOA Management Pack Enterprise Edition

This chapter provides an overview of Oracle SOA Management Pack Enterprise Edition and describes how you can enable this pack and monitor SOA products such as Oracle BPEL Process Manager (BPEL Process Manager), Oracle Service Bus, Oracle SOA Suite 11g, and so on using Enterprise Manager Grid Control.

The Oracle SOA Management Pack Enterprise Edition delivers comprehensive management capabilities for a Service-Oriented Architecture-based (SOA) environment. By combining SOA runtime governance, business-IT alignment, and SOA infrastructure management with Oracle's rich and comprehensive system management solution, Enterprise Manager Grid Control significantly reduces the cost and complexity of managing SOA-based environments.

**Table 4–1 Highlights of Oracle SOA Management Pack Enterprise Edition**

Feature	Benefit
Centralized management console	Provides administrators managing SOA environments with a consolidated browser-based view of the entire enterprise, thereby enabling them to monitor and manage all of their components from a central location.
Discovery and service modeling	Provides discovery of the following: <ul style="list-style-type: none"> <li>■ Oracle SOA Infrastructure deployed to the WebLogic Server.</li> <li>■ Oracle SOA Composite applications deployed to the SOA Infrastructure.</li> <li>■ Oracle BPEL processes deployed to the Oracle BPEL Process Manager (BPEL Process Manager) server and the dependent partner links.</li> <li>■ Oracle Service Bus-based business and proxy services.</li> <li>■ Service modeling offers out-of-the-box automated system modeling capabilities for the SOA infrastructure.</li> </ul>
Runtime governance	Defines SOAP tests to measure and record availability and performance of partner links (or any Web service) and business/proxy services for historical trending, troubleshooting, and root cause analysis purposes. Also provides an error hospital of process instances with drilldowns into instance details.
Infrastructure management	Monitors the availability and performance of the SOA infrastructure components. Both current and historic availability of targets (such as BPEL Process Manager or Oracle Service Bus) are recorded for troubleshooting and root cause analysis.
Configuration management	Collects configuration information for the BPEL Process Manager server/domains/processes and Oracle Service Bus. The parameters can be refreshed, saved, or compared with another target. Different versions of the same target can also be compared.

**Table 4–1 (Cont.) Highlights of Oracle SOA Management Pack Enterprise Edition**

Feature	Benefit
Deployment automation	<p>Automates the deployment of the following:</p> <ul style="list-style-type: none"> <li>■ SOA Artifacts Provisioning: This includes provisioning of SOA Composites, Oracle WebLogic Server Policies, Assertion Templates, and JPS Policy and Credential Stores.</li> <li>■ BPEL processes on BPEL Process Managers</li> <li>■ Oracle Service Bus resources from a source OSB domain to a target OSB domain.</li> </ul> <p>For detailed information on the provisioning procedures, see <i>Enterprise Manager Administrator's Guide for Software and Server Provisioning and Patching</i>.</p>
Adapter metrics	Provides throughput and error metrics for different adapters in graphical format.
Business-IT alignment	Enables you to consolidate their IT and business management tools into a unified system. BAM-EM integration unites business KPIs and system metrics in one system for correlation and trending.
Service level management	Enables you to monitor services from the end-user's perspective using service tests or synthetic transactions, model relationships between services and underlying IT components, and report on achieved service levels.
Application Dependency and Performance	<p>Enables you to manage your SOA solutions by leveraging a model-driven top-down approach within your development, quality assurance (QA), staging, and production environments. Business application owners and operational staff can automatically discover your BPEL workflows and correlate them with the underlying Web services; Enterprise Service Buses (ESBs); and back-end Java 2 Platform, Enterprise Edition (Java EE) resources through detailed modeling and drill-down directly into the performance metrics at the component level.</p> <p>For more information, see <a href="#">Chapter 15, "Introduction to Application Dependency and Performance"</a></p>
Historical analysis and reporting	Store the collected metric and configuration data in a central repository, thereby enabling administrators to analyze metrics through various historical views and facilitate strategic trend analysis and reporting.
Instance Tracing	Allows you to trace the message flow across SOA Composites and SOA Infrastructure instances monitored by Enterprise Manager Grid Control.

# Discovering and Monitoring Oracle BPEL Process Manager

This chapter describes how you can discover and monitor Oracle BPEL Process Manager (BPEL Process Manager) using Enterprise Manager Grid Control.

In particular, this document covers the following:

- [Section 5.1, "Supported Versions"](#)
- [Section 5.2, "Understanding the Discovery Mechanism"](#)
- [Section 5.3, "Understanding the Discovery Process"](#)
- [Section 5.4, "Setting Up Oracle Software Library"](#)
- [Section 5.5, "Downloading One-Off Patches"](#)
- [Section 5.6, "Discovering BPEL Process Manager"](#)
- [Section 5.7, "Configuring BPEL Process Manager"](#)
- [Section 5.8, "Troubleshooting"](#)

## 5.1 Supported Versions

The following are the versions of BPEL Process Manager that are supported for monitoring in Enterprise Manager Grid Control.

**Table 5–1 Supported Versions**

Supported BPEL Process Manager Version	Application Server Deployed To	Supported Enterprise Manager Grid Control
Oracle BPEL Process Manager 10.1.2	Oracle Application Server 10g Release 1 (10.1.2)	Enterprise Manager 10g Grid Control Release 4 (10.2.0.4) Enterprise Manager Enterprise Manager 11g Grid Control Release 1 (11.1.0.1) or higher
Oracle BPEL Process Manager 10.1.3.1 and 10.1.3.3 <i>(Part of Oracle SOA Suite 10.1.3.1 and 10.1.3.3)</i>	Oracle Application Server 10g Release 1 (10.1.3.1) and (10.1.3.3)	Enterprise Manager 10g Grid Control Release 3 (10.2.0.3) or higher Enterprise Manager Enterprise Manager 11g Grid Control Release 1 (11.1.0.1) or higher
Oracle BPEL Process Manager 10.1.3.1 and 10.1.3.3 <i>(Part of Oracle SOA Suite 10.1.3.1 and 10.1.3.3)</i>	Oracle WebLogic Managed Server 9.2	Enterprise Manager 10g Grid Control Release 5 (10.2.0.5) Enterprise Manager 10g Grid Control Release 4 (10.2.0.4) with one-off patches applied. For details, see <a href="#">Section 5.3, "Understanding the Discovery Process"</a> . Enterprise Manager Enterprise Manager 11g Grid Control Release 1 (11.1.0.1) or higher

**Table 5–1 (Cont.) Supported Versions**

Supported BPEL Process Manager Version	Application Server Deployed To	Supported Enterprise Manager Grid Control
Oracle BPEL Process Manager 10.1.3.1 and 10.1.3.3 <i>(Part of Oracle SOA Suite 10.1.3.1 and 10.1.3.3)</i>	IBM WebSphere Application Server 6.1	Enterprise Manager 10g Grid Control Release 5 (10.2.0.5) Enterprise Manager 10g Grid Control Release 4 (10.2.0.4) with one-off patches applied. For details, see <a href="#">Section 5.3, "Understanding the Discovery Process"</a> . Enterprise Manager Enterprise Manager 11g Grid Control Release 1 (11.1.0.1) or higher
Oracle BPEL Process Manager 10.1.3.4 <i>(Part of Oracle SOA Suite 10.1.3.4)</i>	Oracle Application Server 10g Release 1 (10.1.3.1) and (10.1.3.3)	Enterprise Manager 10g Grid Control Release 5 (10.2.0.5) or higher Enterprise Manager Enterprise Manager 11g Grid Control Release 1 (11.1.0.1) or higher
Oracle BPEL Process Manager 10.1.3.4 <i>(Part of Oracle SOA Suite 10.1.3.4)</i>	Oracle WebLogic Managed Server 9.2	Enterprise Manager 10g Grid Control Release 5 (10.2.0.5) or higher Enterprise Manager Enterprise Manager 11g Grid Control Release 1 (11.1.0.1) or higher
Oracle BPEL Process Manager 10.1.3.4 <i>(Part of Oracle SOA Suite 10.1.3.4)</i>	IBM WebSphere Application Server 6.1	Enterprise Manager 10g Grid Control Release 5 (10.2.0.5) or higher Enterprise Manager Enterprise Manager 11g Grid Control Release 1 (11.1.0.1) or higher

## 5.2 Understanding the Discovery Mechanism

The following describes the mechanism followed for discovering BPEL Process Managers in Enterprise Manager Grid Control.

**Table 5–2 Mechanism for Discovering BPEL Process Managers**

BPEL Process Manager Version	Application Server Deployed To	Discovery Mechanism	Process
Oracle BPEL Process Manager 10.1.2	Oracle Application Server 10g Release 1 (10.1.2)	Manual/ Automatic Discovery	<ul style="list-style-type: none"> <li>If the Management Agent is installed before Oracle Application Server and BPEL Process Manager are installed, then you must <b>manually discover</b> that Oracle Application Server and BPEL Process Manager in Enterprise Manager Grid Control.</li> <li>If the Management Agent is installed after Oracle Application Server and BPEL Process Manager are installed, then Enterprise Manager Grid Control <b>automatically discovers</b> that Oracle Application Server and BPEL Process Manager</li> </ul> <p>The Management Agent can be installed along with Enterprise Manager Grid Control or separately as a standalone product.</p> <p>For discovery procedures, see <a href="#">Section 5.6.1, "Discovering BPEL Process Manager Deployed to Oracle Application Server"</a>.</p>
Oracle BPEL Process Manager 10.1.3.1, 10.1.3.3, 10.1.3.4 <i>(Part of Oracle SOA Suite 10.1.3.1, 10.1.3.3, 10.1.3.4)</i>	Oracle Application Server 10g Release 1 (10.1.3.1) and (10.1.3.3)	Manual/ Automatic Discovery	<ul style="list-style-type: none"> <li>If the Management Agent is installed before Oracle Application Server and BPEL Process Manager are installed, then you must <b>manually discover</b> that Oracle Application Server and BPEL Process Manager in Enterprise Manager Grid Control.</li> <li>If the Management Agent is installed after Oracle Application Server and BPEL Process Manager are installed, then Enterprise Manager Grid Control <b>automatically discovers</b> that Oracle Application Server and BPEL Process Manager</li> </ul> <p>The Management Agent can be installed along with Enterprise Manager Grid Control or separately as a standalone product.</p> <p>For discovery procedures, see <a href="#">Section 5.6.1, "Discovering BPEL Process Manager Deployed to Oracle Application Server"</a>.</p>

**Table 5–2 (Cont.) Mechanism for Discovering BPEL Process Managers**

BPEL Process Manager Version	Application Server Deployed To	Discovery Mechanism	Process
Oracle BPEL Process Manager 10.1.3.1, 10.1.3.3, 10.1.3.4 <i>(Part of Oracle SOA Suite 10.1.3.1, 10.1.3.3, 10.1.3.4)</i>	Oracle WebLogic Managed Server 9.2	Manual Discovery	First, manually discover Oracle WebLogic Managed Server. For procedures, see <a href="#">Section 5.6.2.1, "Discovering Oracle WebLogic Managed Server"</a> . Then, manually discover BPEL Process Manager. For procedures, see <a href="#">Section 5.6.2.2, "Discovering BPEL Process Manager Deployed to Oracle WebLogic Managed Server"</a> .
Oracle BPEL Process Manager 10.1.3.1, 10.1.3.3, 10.1.3.4 <i>(Part of Oracle SOA Suite 10.1.3.1, 10.1.3.3, 10.1.3.4)</i>	IBM WebSphere Application Server 6.1	Manual Discovery	First, manually discover IBM WebSphere Application Server. For procedures, see <a href="#">Section 5.6.3.1, "Discovering IBM WebSphere Application Server"</a> . Then, manually discover BPEL Process Manager. For procedures, see <a href="#">Section 5.6.3.2, "Discovering BPEL Process Manager Deployed to IBM WebSphere Application Server"</a> .

## 5.3 Understanding the Discovery Process

The following describes the overall process involved in discovering and monitoring BPEL Process Manager in Enterprise Manager Grid Control. Follow the instructions outlined against each step in this process to successfully discover and monitor your BPEL Process Manager.

**Table 5–3 Discovery Process**

Step	Requirement	Description
1	BPEL Process Manager	Install the BPEL Process Manager software in one of the following ways: <ul style="list-style-type: none"> <li>For Oracle middleware, download and install the BPEL Process Manager using Oracle BPEL Process Manager 10.1.2, Oracle SOA Suite 10.1.3.1, 10.1.3.3, or 10.1.3.4 from the following URL: <a href="http://www.oracle.com/technology/software/tech/soa/index.html">http://www.oracle.com/technology/software/tech/soa/index.html</a></li> <li>For non-Oracle middleware, download and install the BPEL Process Manager from the following URL: <a href="http://www.oracle.com/technology/software/products/ias/bpel/index.html">http://www.oracle.com/technology/software/products/ias/bpel/index.html</a></li> </ul>
2	Enterprise Manager Grid Control	<ul style="list-style-type: none"> <li>If you want to monitor Oracle BPEL Process Manager 10.1.3.3 that is deployed to Oracle Application Server 10g Release 1 (10.1.3.3), then install Enterprise Manager 10g Grid Control Release 3 (10.2.0.3) or higher.</li> <li>If you want to monitor Oracle BPEL Process Manager 10.1.2 that is deployed to Oracle Application Server 10g Release 1 (10.1.2), then install Enterprise Manager 10g Grid Control Release 4 (10.2.0.4).</li> <li>If you want to monitor Oracle BPEL Process Manager 10.1.3.3 that is deployed to Oracle WebLogic Managed Server 9.2 or IBM WebSphere Application Server 6.1, then install Enterprise Manager 10g Grid Control Release 4 (10.2.0.4) and apply the one-off patches.</li> <li>If you want to monitor Oracle BPEL Process Manager 10.1.3.4 that is deployed to Oracle Application Server 10g Release 1 (10.1.3.1 or 10.1.3.3), Oracle WebLogic Managed Server 9.2, or IBM WebSphere Application Server 6.1, then install Enterprise Manager 10g Grid Control Release 5 (10.2.0.5).</li> </ul> <p>Note that 10.2.0.3, 10.2.0.4, 10.2.0.5 are patch sets. Therefore, if you are installing Enterprise Manager Grid Control for the first time, then install the base release first and then apply the 10.2.0.3, 10.2.0.4, 10.2.0.5 patch set on it. For information about installing the base release of Enterprise Manager Grid Control, see the <i>Enterprise Manager Grid Control Installation and Basic Configuration Guide</i> available at: <a href="http://www.oracle.com/technology/documentation/oem.html">http://www.oracle.com/technology/documentation/oem.html</a></p> <p>Oracle recommends that you install the Enterprise Manager Grid Control components on a host that is different from the host where the BPEL Process Manager is installed. For example, if the BPEL Process Manager is installed on host1.xyz.com, then install and configure Oracle Management Service (OMS) and the Management Repository on host2.xyz.com.</p>

**Table 5–3 (Cont.) Discovery Process**

Step	Requirement	Description
3	Oracle Management Agent (Management Agent)	<p>Install Oracle Management Agent 10g Release 3 (10.2.0.3) or higher on every host where BPEL Process Manager is installed.</p> <p>If Oracle Application Server/BPEL Process Manager and Enterprise Manager Grid Control are all on the same host, then you do not have to install a separate Management Agent. The Management Agent that comes with Enterprise Manager Grid Control is sufficient. However, if they are different hosts, then you must install a separate Management Agent on every host where BPEL Process Manager is installed.</p> <p>You can install the Management Agent in one of the following ways:</p> <ul style="list-style-type: none"> <li>▪ Invoke the installer provided with Enterprise Manager 10 Grid Control Release 3 (10.2.0.3) or higher, and select the installation type <b>Additional Management Agent</b>.</li> <li>▪ Use the Agent Deploy application within the Grid Control console.</li> <li>▪ Use the full agent kit that is available at: <a href="http://www.oracle.com/technology/software/products/oem/htdocs/agentsoft.html">http://www.oracle.com/technology/software/products/oem/htdocs/agentsoft.html</a></li> </ul> <p>For information about installing the Management Agent, see the <i>Enterprise Manager Grid Control Installation and Basic Configuration Guide</i> available at: <a href="http://www.oracle.com/technology/documentation/oem.html">http://www.oracle.com/technology/documentation/oem.html</a></p>
4	One-Off Patches	<p>If you are using Enterprise Manager 10g Grid Control Release 4 (10.2.0.4), then you will be able to discover and monitor the BPEL Process Manager deployed to Oracle WebLogic Managed Server 9.2 and IBM WebSphere Application Server 6.1 only when the following one-off patches are applied.</p> <p><b>OMS One-Off Patches:</b> Apply these one-off patches to Oracle Management Service (OMS) in the order given below.</p> <ul style="list-style-type: none"> <li>▪ Patch 6678593 (Or ARU Patch 10024972)</li> <li>▪ Patch 6877335 (Or ARU Patch 10024974)</li> <li>▪ Patch 6850462 (Or ARU Patch 10024971) - Before applying this one-off patch, you must set up Oracle Software Library in Enterprise Manager Grid Control. For procedures to set up the library, see <a href="#">Section 5.4, "Setting Up Oracle Software Library"</a>.</li> </ul> <p><b>Agent One-Off Patches:</b> Apply these one-off patches to the Management Agent in the order given below.</p> <ul style="list-style-type: none"> <li>▪ Patch 6692284 (Or ARU Patch 10024969)</li> <li>▪ Patch 6877323 (Or ARU Patch 10024970)</li> </ul> <p>For information about these patches, refer to the ReadMe.txt file that comes with the one-off patch. For procedures to download these one-off patches, see <a href="#">Section 5.5, "Downloading One-Off Patches"</a>.</p>
5	Discovery in Enterprise Manager Grid Control	<p>BPEL Process Managers deployed to Oracle Application Servers are automatically discovered in Enterprise Manager Grid Control.</p> <p>BPEL Process Managers deployed to Oracle WebLogic Managed Servers and IBM WebSphere Application Servers must be manually discovered in Enterprise Manager Grid Control. For procedures to discover them, see <a href="#">Section 5.6, "Discovering BPEL Process Manager"</a>.</p>

## 5.4 Setting Up Oracle Software Library

If you are using Enterprise Manager 10g Grid Control Release 4 (10.2.0.4) to discover and monitor the BPEL Process Manager deployed to Oracle WebLogic Managed Server 9.2 and IBM WebSphere Application Server 6.1, then you must apply the one-off patches as described in [Section 5.3, "Understanding the Discovery Process"](#). However, before applying the one-off patch 6850462, you must set up Oracle Software Library (Software Library) in Enterprise Manager Grid Control.

If you do not set up the Software Library and if you try to apply the one-off patch, you might see the following error:

```
ERROR !
Software Library has not been setup !
Please configure Software Library and try again
```



**Note:** This is a requirement only for patch 6850462. The other patches do not require the Software Library to be set up.

To set up the Software Library:

1. Log in to Enterprise Manager Grid Control.
2. Click **Deployments** and then **Provisioning**.  
Enterprise Manager Grid Control displays the Provisioning page.
3. In the Provisioning page, Click **Administration**.
4. In the Administration page, from the Software Library Configuration section, click **Add**. Enterprise Manager Grid Control displays the Add Software Library Location page.
5. In the Add Software Library Location page, specify a valid directory path where you want to store the raw data for the components, and click **OK**.

**Note:** For more information about setting up the Software Library, see the *Enterprise Manager Advanced Configuration Guide* available at the following URL:

<http://www.oracle.com/technology/documentation/oem.html>

## 5.5 Downloading One-Off Patches

If you are using Enterprise Manager 10g Grid Control Release 4 (10.2.0.4) for discovering and monitoring of BPEL Process Manager deployed to Oracle WebLogic Managed Server 9.2 or IBM WebSphere Application Server 6.1, then apply the following one-off patches.

**Note:** You do not required these patches for Enterprise Manager 10g Grid Control Release 5 (10.2.0.5) or higher.

**Table 5–4 One-Off Patches to Download**

One-Off Patch Type	One-Off Patches to Download
OMS One-Off Patches	Apply these one-off patches to Oracle Management Service (OMS) in the order given below: <ul style="list-style-type: none"> <li>■ Patch 6678593 (Or ARU Patch 10024972)</li> <li>■ Patch 6877335 (Or ARU Patch 10024974)</li> <li>■ Patch 6850462 (Or ARU Patch 10024971)</li> </ul>
Agent One-Off Patches	Apply these one-off patches to Oracle Management Service (OMS) in the order given below: <ul style="list-style-type: none"> <li>■ Patch 6692284 (Or ARU Patch 10024969)</li> <li>■ Patch 6877323 (Or ARU Patch 10024970)</li> </ul>

To download these one-off patches:

1. Log in to My Oracle Support at:

<https://support.oracle.com/>

2. Click **Patches & Updates**.
3. In the Patches & Updates page, from the **Quick Find** list, select **Patch Number**, quote the one-off patch number you want to download, and click **Go**.  
My Oracle Support displays the Patch *<patch\_number>* page that provides details about the one-off patch.
4. In the Patch *<patch\_number>* page, click **Download** to download the patch. Click **View Readme** to view the ReadMe file and learn more about the patch.

## 5.6 Discovering BPEL Process Manager

This section describes the procedures for discovering BPEL Process Managers. In particular, this section covers the following:

- [Section 5.6.1, "Discovering BPEL Process Manager Deployed to Oracle Application Server"](#)
- [Section 5.6.2, "Discovering BPEL Process Manager Deployed to Oracle WebLogic Managed Server"](#)
- [Section 5.6.3, "Discovering BPEL Process Manager Deployed to IBM WebSphere Application Server"](#)

### 5.6.1 Discovering BPEL Process Manager Deployed to Oracle Application Server

A BPEL Process Manager deployed to Oracle Application Server is manually or automatically discovered in Enterprise Manager Grid Control depending on when the Management Agent is installed.

- If the Management Agent is installed before Oracle Application Server and BPEL Process Manager are installed, then you must manually discover that Oracle Application Server and BPEL Process Manager in Enterprise Manager Grid Control.
- If the Management Agent is installed after Oracle Application Server and BPEL Process Manager are installed, then Enterprise Manager Grid Control automatically discovers that Oracle Application Server and BPEL Process Manager.

---

**Note:** You must install a Management Agent on every host where BPEL Process Manager is installed. If Oracle Application Server/BPEL Process Manager and Enterprise Manager Grid Control are all on the same host, then you need not install a separate Management Agent. The Management Agent that comes with Enterprise Manager Grid Control is sufficient. However, if they are different hosts, then you must install a separate Management Agent on every host where BPEL Process Manager is installed. The Management Agent can be installed along with Enterprise Manager Grid Control or separately as a standalone product.

---

Also note that if you have added a new BPEL Process Manager to an Oracle Application Server that is already discovered and monitored in Enterprise Manager Grid Control, then you must manually *rediscover* that Oracle Application Server.

To manually discover or *rediscover* Oracle Application Server:

1. Log in to Enterprise Manager Grid Control.
2. Click **Targets** and then **Middleware**.  
Enterprise Manager Grid Control 11g Release 1 displays the Middleware page that lists all the middleware targets being monitored. In Enterprise Manager 10g Grid Control Release 4 (10.2.0.4) or lower, the Middleware tab is Application Servers.
3. (Only for *Rediscovering*) In the Middleware page, select the Oracle Application Server that you want to rediscover and click **Remove**.
4. In the Middleware page, from the **Add** list, select **Oracle Application Server** and click **Go**.  
Enterprise Manager Grid Control displays the Add Oracle Application Server Target: Specify Host page.
5. In the Add Oracle Application Server Target: Specify Host page, specify the name of the host where that Oracle Application Server is running, and click **Continue**.  
Enterprise Manager Grid Control rediscovers that Oracle Application Server along with its core components and the newly added BPEL Process Manager.

## 5.6.2 Discovering BPEL Process Manager Deployed to Oracle WebLogic Managed Server

To discover the BPEL Process Manager deployed to Oracle WebLogic Managed Server, you have to first discover and add Oracle WebLogic Managed Server to Enterprise Manager Grid Control.

This section describes the procedures for the following:

- [Section 5.6.2.1, "Discovering Oracle WebLogic Managed Server"](#)
- [Section 5.6.2.2, "Discovering BPEL Process Manager Deployed to Oracle WebLogic Managed Server"](#)

### 5.6.2.1 Discovering Oracle WebLogic Managed Server

To discover and add Oracle WebLogic Managed Server to Enterprise Manager Grid Control:

1. Log in to Enterprise Manager Grid Control (as a SYSMAN user).
2. Click **Targets** and then **Middleware**.  
Enterprise Manager Grid Control displays the Middleware page that lists all the middleware targets being monitored. In Enterprise Manager 10g Grid Control Release 4 (10.2.0.4) or lower, the Middleware tab is Application Servers
3. In the Middleware page, from the **Add** list, select **Oracle Fusion Middleware / WebLogic Server Domain**, and click **Go**.  
Enterprise Manager Grid Control displays the Add Oracle Fusion Middleware / WebLogic Server Domain wizard that captures the details of the Oracle WebLogic Server Domain to be discovered and monitored.
4. In the wizard, specify the required details and click **Next** on each page to reach the end of the wizard.  
For information about the details to be provided for each page of the wizard, click **Help** on each page.

5. In the last page of the wizard, click **Finish** to complete the discovery process and add the target to Grid Control for monitoring purposes.

Enterprise Manager Grid Control displays the Middleware page with a confirmation message that confirms that the Oracle WebLogic Manager Server has been successfully added to Grid Control. In the Middleware page that shows all the middleware targets being monitored, you can see the Oracle WebLogic Managed Server you just added.

### 5.6.2.2 Discovering BPEL Process Manager Deployed to Oracle WebLogic Managed Server

To discover and add the BPEL Process Manager deployed to Oracle WebLogic Managed Server:

1. Log in to Enterprise Manager Grid Control (as a SYSMAN user).
2. Click **Targets** and then **All Targets**.  
Enterprise Manager Grid Control displays the All Targets page that lists all the targets being monitored.
3. In the All Targets page, from the **Search** menu, select **Agent** and click **Go**.  
Enterprise Manager Grid Control displays all the Agents that are being monitored.
4. From the list of monitored agents, click the agent that monitors the Oracle WebLogic Managed Server.  
Enterprise Manager Grid Control displays the Agent home page that provides details about the selected agent.
5. In the Agent home page, from the **Add** list, select **Oracle BPEL Process Manager** and click **Go**.

---

**Note:** At this point, if you encounter an exception error (404 Not Found), then follow the workaround steps given in [Table 5-9](#) and resolve the issue.

---

Enterprise Manager Grid Control displays the Select Application Server page of the Add BPEL Process Manager wizard.

- a. In the Select Application Server page, provide the following details and click **Next**.

**Table 5-5 Select Application Server Page - Element Description**

UI Page Element	Description
Application Server Type	Select the type of application server where the BPEL Process Manager to be discovered is running.
Application Server Name	Specify the name of the application server where the BPEL Process Manager to be discovered is running. If you are not sure about the name, click the each icon (torch icon) to view a list of application servers and select the appropriate one. The application server name must be suffixed with <i>oracleBPELServer</i> .

- b. In the Target Details page, provide the following details and click **Next**.

**Table 5–6 Target Details Page - Element Description**

UI Page Element	Description
Oracle Home	Specify the full path to the Oracle Application Server home directory where the BPEL Process Manager is installed. For example, /opt/app/orabpel/product/10.1.3.1/OracleAS.
Application Server Home	Specify the full path to the directory where Oracle Weblogic Managed Server (to which the BPEL target is deployed) is running. For example, /opt/wls9.2/weblogic9.2.

**Note:**

- Enterprise Manager Grid Control checks the configuration settings of the associated application server and prefills the values for fields such as BPEL Process Manager Name, Display Name, Context Provider URL, and Oracle BPEL PM Console URL.
  - At this point, if you encounter a discovery failure error, then follow the workaround steps given in [Table 5–11](#) to resolve the issue.
- c. In the Host Credentials page, specify the operating system credentials of the host where BPEL Process Manager is running. By default, the fields are prefilled with preferred credentials that are stored in the Management Repository for the selected host. You can either use these prefilled values or edit them to override the preferred credentials with your new credentials.
  - d. In the Review page, review the details and click **Finish** to complete the discovery process and add the target to Enterprise Manager Grid Control. Enterprise Manager Grid Control displays the Agent home page with a confirmation message that confirms that the BPEL Process Manager has been successfully added for monitoring.

**Note:** At this point, if you encounter a discovery failure error, then follow the workaround steps given in [Table 5–11](#) and resolve the issue.

6. To verify whether the BPEL Process Manager has been added, click **Targets** and then **Middleware**.

Enterprise Manager Grid Control displays the Middleware page that shows all the middleware targets being monitored, including the Oracle WebLogic Managed Server and the BPEL Process Manager you just added. In Enterprise Manager 10g Grid Control Release 4 (10.2.0.4) or lower, the Middleware tab is Application Servers

### 5.6.3 Discovering BPEL Process Manager Deployed to IBM WebSphere Application Server

To discover the BPEL Process Manager deployed to IBM WebSphere Application Server, you have to first discover and add IBM WebSphere Application Server to Enterprise Manager Grid Control.

This section describes the procedures for the following:

- [Section 5.6.3.1, "Discovering IBM WebSphere Application Server"](#)
- [Section 5.6.3.2, "Discovering BPEL Process Manager Deployed to IBM WebSphere Application Server"](#)

### 5.6.3.1 Discovering IBM WebSphere Application Server

To discover and add IBM WebSphere Application Server to Enterprise Manager Grid Control:

1. Log in to Enterprise Manager Grid Control (as a SYSMAN user).
2. Click **Targets** and then **Middleware**.

Enterprise Manager Grid Control displays the Middleware page that lists all the middleware targets being monitored.

3. In the Middleware page, from the **Add** list, select **IBM WebSphere Application Server**, and click **Go**.

Enterprise Manager Grid Control displays the Add IBM WebSphere Application Server wizard that captures the details of the IBM WebSphere Application Server to be discovered and monitored.

4. In the Add IBM WebSphere Application Server wizard, specify the required details and click **Next** on each page to reach the end of the wizard.

For information about the details to be provided for each page of the wizard, click **Help** on each page.

5. In the last page of the Add IBM WebSphere Application Server wizard, click **Finish** to complete the discovery process and add the target to Enterprise Manager Grid Control for monitoring purposes.

Enterprise Manager Grid Control displays the Middleware page with a confirmation message that confirms that the IBM WebSphere Application Server has been successfully added for monitoring. In the Middleware page that shows all the application server being monitored, you can see the IBM WebSphere Application Server you just added.

### 5.6.3.2 Discovering BPEL Process Manager Deployed to IBM WebSphere Application Server

To discover and add the BPEL Process Manager deployed to IBM WebSphere Application Server:

1. Log in to Enterprise Manager Grid Control (as a SYSMAN user).
2. Click **Targets** and then **All Targets**.

Enterprise Manager Grid Control displays the All Targets page that lists all the targets being monitored.

3. In the All Targets page, from the **Search** menu, select **Agent** and click **Go**.

Enterprise Manager Grid Control displays all the Agents that are being monitored.

4. From the list of monitored agents, click the agent that monitors the IBM WebSphere Application Server.

Enterprise Manager Grid Control displays the Agent home page that provides details about the selected agent.

5. In the Agent home page, from the **Add** list, select **Oracle BPEL Process Manager** and click **Go**.

Enterprise Manager Grid Control displays the Select Application Server page of the Add BPEL Process Manager wizard.

- a. In the Select Application Server page, provide the following details and click **Next**.

**Table 5–7 Select Application Server Page - Element Description**

UI Page Element	Description
Application Server Type	Select <b>IBM WebSphere Application Server</b> from the list.
Application Server Name	Specify the name of IBM WebSphere Application Server where the BPEL Process Manager to be discovered is running. If you are not sure about the name, click the search icon (torch icon) to view a list of application servers and select the appropriate one. The application server name must be suffixed with <i>oracleBPELServer</i> .

- b. In the Target Details page, provide the following details and click **Next**.

**Table 5–8 Target Details Page - Element Description**

UI Page Element	Description
Oracle Home	Specify the full path to the Oracle Application Server home directory where the BPEL Process Manager is installed. For example, <code>/opt/app/orabpel/product/10.1.3.1/OracleAS</code> .
Application Server Home	Specify the full path to the directory where IBM WebSphere Application Server (to which the BPEL target is deployed) is running.
BPEL Application Installation Location	Specify the full path to the installation directory where the BPEL application is installed.  For example, if the BPEL application is installed in <code>&lt;\$WEBSPPHERE_HOME&gt;/profiles/AppSrv01/installedApps/sta00114Cell01/CollaxaWebApplications-sta00114Node01.ear</code> , then specify the path as <code>&lt;\$WEBSPPHERE_HOME&gt;/profiles/AppSrv01/installedApps</code> . Here, replace <code>\$WEBSPPHERE_HOME</code> with the full path of the application home location.

---



---

**Note:**

- Enterprise Manager Grid Control checks the configuration settings of the associated application server and prefills the values for fields such as BPEL Process Manager Name, Display Name, Context Provider URL, and Oracle BPEL PM Console URL.
- 
- 

- c. In the Host Credentials page, specify the operating system credentials of the host where BPEL Process Manager is running. By default, the fields are pre-filled with preferred credentials that are stored in the Management Repository for the selected host. You can either use these pre-filled values or edit them to override the preferred credentials with your new credentials.

- d. In the Review page, review the details and click **Finish** to complete the discovery process and add the target to Enterprise Manager Grid Control.  
Enterprise Manager Grid Control displays the Agent home page with a confirmation message that confirms that the BPEL Process Manager has been successfully added for monitoring.

---

**Note:** At this point, if you encounter a discovery failure error, then follow the workaround steps given in [Table 5-12](#) and resolve the issue.

---

6. To verify whether the BPEL Process Manager has been added, click **Targets** and then **Middleware**.

Enterprise Manager Grid Control displays the Middleware page that shows all the middleware targets being monitored, including the IBM WebSphere Application Server and the BPEL Process Manager you just added.

## 5.7 Configuring BPEL Process Manager

After discovering BPEL Process Manager, you must perform the following configuration steps:

- [Section 5.7.1, "Specifying Details for Monitoring BPEL Process Manager"](#)
- [Section 5.7.2, "Adding BPEL JAR Files To Agent CLASSPATH"](#)

### 5.7.1 Specifying Details for Monitoring BPEL Process Manager

Follow these steps to specify the details required for monitoring BPEL Process Managers. If the values are prefilled, then validate them.

1. In the BPEL Process Manager Home page, from the Related Links section, click **Monitoring Configuration**.
2. In the Monitoring Configuration page, specify the following details. If these values are prefilled, then validate them.
  - **BPEL Admin Username** - Specify the BPEL administrator user ID.
  - **BPEL Password** - Specify the BPEL admin password.

When adding the credentials, validate the following two criteria:

- BPEL Admin User ID and password should have BPEL Admin role
- The same credentials should succeed for the BPEL console login operation

- **Initial Context Factory** - Specify the initial context factor. You can copy the following string value:

```
com.evermind.server.rmi.RMIInitialContextFactory
```

- **Context Provider URL** - Specify the context provider URL. You can copy the following string value:

```
opmn:ormi://<host>:<opmn_port>:home/orabpel
```



---

**Note:** Replace the <host>,<opmn port> with the correct host address and opmn port number details for the Oracle Application Server where the BPEL Process Manager is deployed.

To retrieve SOA Applications Server OPMN PORT details, follow these steps:

1. Open the configuration file `$SOA_ORACLE_HOME/opmn/conf/opmn.xml`. `$SOA_ORACLE_HOME` corresponds to SOA Application server home location.
2. Identify the value of the request port attribute in the configuration file.
  - **BPEL Repository Host Name** - Specify the BPEL Dehydration store (database) host name.
  - **BPEL Repository Port** - Specify the BPEL Dehydration store (database) port.
  - **BPEL Repository SID** - Specify the BPEL Dehydration store (database) SID.
  - **BPEL Repository User Name** - Specify the BPEL Dehydration store (database) user name. By default, the user name is `orabpel`.
  - **BPEL Repository Password** - Specify the BPEL Dehydration store (database) password. By default, the password is `welcome1`.
  - **Recoverable Instances Time Threshold (Days)** - Specify the number of days for which the retryable instances must be shown.
  - **Process Aggregate State** - Specify 5, a numeric value that signifies the "constant" state of the BPEL target.
3. Click **OK** to save the settings.

## 5.7.2 Adding BPEL JAR Files To Agent CLASSPATH

Follow these steps to add the required BPEL JAR files (storage containers) to the Agent CLASSPATH:

1. Log in to the host machine where SOA is installed.
2. Go to the `ORACLE_HOME` of the EM Agent installed on the same host.
3. Open file '`$AGENT_ORACLE_HOME/sysman/config/emd.properties`' as shown in figure.
4. Check whether the following jar files have been added to the CLASSPATH property in the file. The property must manually update with the BPEL-specific jar file names to get the process listing.
  - `$BPEL_SERVER_ORACLE_HOME/opmn/lib/optic.jar`
  - `$BPEL_SERVER_ORACLE_HOME/bpel/lib/orabpel.jar`
  - `$BPEL_SERVER_ORACLE_HOME/bpel/lib/orabpel-common.jar`
  - `$BPEL_SERVER_ORACLE_HOME/bpel/lib/orabpel-thirdparty.jar`
  - `$BPEL_SERVER_ORACLE_HOME/j2ee/home/oc4jclient.jar`
  - `$BPEL_SERVER_ORACLE_HOME/j2ee/home/j2ee_1.3.01.jar`

---

**Note:** The \$BPEL\_SERVER\_ORACLE\_HOME should be replaced with the absolute path of the ORACLE\_HOME path of the application server where the SOA is installed.

---

5. Add the JAR filenames to CLASSPATH property. When adding the jar files to CLASSPATH, ensure that the BPEL home `optic.jar` property is the first value in the classpath.
6. Restart the Agent.

### 5.7.3 Registering BPEL Process Manager Credentials and Host Credentials

Follow these steps to register the credentials of the BPEL Process Manager, and the credentials of the host where BPEL Process Manager is running.

1. In Grid Control, from the top-right corner of the page, click **Preferences**.
2. On the General page, from the vertical menu bar, click **Preferred Credentials**.
3. On the Preferences page, for the target type **Oracle BPEL Process Manager**, from the **Set Credentials** column, click the icon.
4. On the Oracle BPEL Process Manager Preferred Credentials page, in the Target Credentials section, for the BPEL Process Manager target you discovered, specify the administrator credentials and host credentials. Then click **Apply**.
5. Similarly, on the Preferences page, for the target type **Host**, from the **Set Credentials** column, click the icon.
6. On the Host Preferred Credentials page, in the Target Credentials section, for the host on which the BPEL Process Manager is running, specify the normal credentials and privileged credentials. Then click **Apply**.

## 5.8 Troubleshooting

This section describes the errors you might encounter while discovering BPEL Process Managers, and the workaround steps you can follow to resolve each of them.

This section covers the following:

- [Section 5.8.1, "404 Not Found Errors"](#)
- [Section 5.8.2, "Discovery Errors on Target Details Page \(Adding BPEL Deployed to Oracle WebLogic Managed Server\)"](#)
- [Section 5.8.3, "Discovery Errors on Review Page \(Adding BPEL Deployed to Oracle WebLogic Managed Server\)"](#)
- [Section 5.8.4, "Discovery Errors on Review Page \(Adding BPEL Deployed to IBM WebSphere Application Server\)"](#)

### 5.8.1 404 Not Found Errors

The following exception error occurs when you try to access the Add BPEL Process Manager wizard to add a BPEL Process Manager installed on Oracle WebLogic Managed Server.

**Table 5–9 Error Message: 404 Not Found - Workaround Steps**

Error Message	Workaround Steps
<p>404 Not Found                      OracleJSP:java.io.FileNotFoundException                      Set the init-param debug_mode to "true"                      to see the complete exception message</p>	<ol style="list-style-type: none"> <li>1. Copy the following file:                              /opt/app/consola10g/oms10g/j2ee/OC4J_EM/config/global-web-application.xml</li> <li>2. Paste the file in the following location:                              /opt/app/consola10g/oms10g/sysman/j2ee/config/                              /</li> <li>3. Ensure that the following parameter is uncommented in the file:                              &lt;init-param&gt;                              &lt;param-name&gt;main_mode&lt;/param-name&gt;                              &lt;param-value&gt;justrun&lt;/param-value&gt;                              &lt;/init-param&gt;</li> </ol>

### 5.8.2 Discovery Errors on Target Details Page (Adding BPEL Deployed to Oracle WebLogic Managed Server)

The following error occurs in the Target Details page of the Add BPEL Process Manager wizard where you provide details about the BPEL Process Manager installed on Oracle WebLogic Managed Server.

**Table 5–10 Errors on Target Details Page While Adding BPEL Process Manager Deployed to Oracle WebLogic Managed Server**

Error Message	Workaround Steps
<p>Oracle BPEL Process Manager not found in the selected Application Server. Select another Application Server.</p>	<p>This error may occur if BPEL is not deployed on the selected Application Server or if the configuration data has not been collected.</p> <p>To resolve this issue:</p> <ol style="list-style-type: none"> <li>1. Select another Application Server.</li> <li>2. Navigate to the Application Server Home page and click <b>Administration</b> and <b>Last Collected</b>.</li> </ol>

### 5.8.3 Discovery Errors on Review Page (Adding BPEL Deployed to Oracle WebLogic Managed Server)

The following errors occur in the Review page of the Add BPEL Process Manager wizard when you are about to add a BPEL Process Manager installed on Oracle WebLogic Managed Server, to Enterprise Manager Grid Control for monitoring purposes.

**Table 5–11 Errors on Review Page While Adding BPEL Process Manager Deployed to Oracle WebLogic Managed Server**

Error Message	Workaround Steps
Discovery Failure - Oracle BPEL Process Manager target discovery failed due to incorrect host credentials.	<ol style="list-style-type: none"> <li>1. In the last page of the Add BPEL Process Manager wizard where you see this error message, click <b>Previous</b> to reach the Host Credentials page.</li> <li>2. In the Host Credentials page, specify the correct host credentials or set the preferred credentials for the specific host. Ensure that these are Agent user credentials.</li> </ol>
Oracle BPEL Proess Manager Discovery Failed - Unable to connect to Oracle BPEL Process Manager. The possible reasons can be incorrect path or insufficient permission to access Oracle BPEL Process Manager home location or inaccessible Oracle BPEL Process Manager home location. Review the specified value.	<ol style="list-style-type: none"> <li>1. In the last page of the Add BPEL Process Manager wizard, click <b>Previous</b> repeatedly to reach the Target Details page.</li> <li>2. In the Target Details page, verify the Oracle home location of the BPEL Process Manager.</li> <li>3. In the Target Details page, verify the installation location of the associated application server.</li> </ol>
Oracle BPEL Proess Manager Discovery Failed - Unable to connect to Oracle BPEL Process Manager. The possible reasons can be incorrect path or insufficient permission to access Oracle BPEL Process Manager home location or inaccessible Oracle BPEL Process Manager home location. Review the specified value.	Ensure that the BPEL directories have read permission for the Agent user.

### 5.8.4 Discovery Errors on Review Page (Adding BPEL Deployed to IBM WebSphere Application Server)

The following errors occur in the Review page of the Add BPEL Process Manager wizard when you are about to add a BPEL Process Manager installed on IBM WebSphere Application Server, to Enterprise Manager Grid Control for monitoring purposes.

**Table 5–12 Error on Review Page While Adding BPEL Process Manager Deployed to IBM WebSphere Application Server**

Error Message	Workaround Steps
<p>Discovery Failure - Oracle BPEL Process Manager target discovery failed due to incorrect host credentials.</p>	<ol style="list-style-type: none"> <li>1. In the last page of the Add BPEL Process Manager wizard where you see this error message, click <b>Previous</b> to reach the Host Credentials page.</li> <li>2. In the Host Credentials page, specify the correct host credentials or set the preferred credentials for the specific host. Ensure that these are Agent user credentials.</li> </ol>
<p>Oracle BPEL Process Manager Discovery Failed - Unable to connect to Oracle BPEL Process Manager. The possible reasons can be incorrect path or insufficient permission to access Oracle BPEL Process Manager home location or inaccessible Oracle BPEL Process Manager home location. Review the specified value.</p>	<ol style="list-style-type: none"> <li>1. In the last page of the Add BPEL Process Manager wizard where you see this error message, click <b>Previous</b> repeatedly to reach the Target Details page.</li> <li>2. In the Target Details page, verify the BPEL application installation location.  For example, the BPEL application may be installed at the following location:   <pre>&lt;\$WEBSPPHERE_ HOME&gt;/profiles/AppSrv01/installedApps/sta00114 Cell01/CollaxaWebApplications-sta00114Node01.e ar</pre>                     In this case, the path you specify must look like this:   <pre>&lt;\$WEBSPPHERE_ HOME&gt;/profiles/AppSrv01/installedApps</pre> <b>Note:</b> Replace \$WEBSPPHERE_HOME with the absolute application home location.                 </li> <li>3. In the Target Details page, verify the application server home location of the associated application server.</li> <li>4. In the Target Details page, verify the Oracle home location of the BPEL Process Manager.</li> </ol>

### 5.8.5 Display Errors on Processes Page

Sometimes, after the discovery of a BPEL Process Manager, the BPEL process may occasionally not be listed in the BPEL Process Manager Processes page in Enterprise Manager Grid Control.

There are two causes for this and two ways to ensure they display on the Processes page. The sections below discuss these causes and workaround steps to fix them.

#### 5.8.5.1 No Credentials Specified for Monitoring BPEL Process Manager

You may not have specified the credentials required for monitoring BPEL Process Managers. To address this, do the following:

1. In the BPEL Process Manager Home page, from the Related Links section, click **Monitoring Configuration**.
2. In the Monitoring Configuration page, check the following fields:
  - **BPEL Admin Username** - Provide the BPEL administrator user ID.
  - **BPEL Password** - Provide the BPEL admin password.

When adding the credentials validate the following two criteria:

- BPEL Admin User ID and password should have BPEL Admin role
- The same credentials should succeed for the BPEL console login operation

- **Initial Context Factory** - In case this field is empty, copy the following string value:  
`com.evermind.server.rmi.RMIInitialContextFactory`
- **Context Provider URL** - In case this field is empty, copy the following highlighted string value:  
`opmn:ormi://<host>:<opmn_port>:home/orabpel`

---

**Note:** Replace the <host>,<opmn port> with the correct host address and opmn port number details for the Oracle Application Server where the BPEL Process Manager is deployed.

---

3. Click **OK** to save the settings.

### 5.8.5.2 Required BPEL JAR Files Not Added To Agent CLASSPATH

You may not have added the required BPEL JAR files (storage containers) to the Agent CLASSPATH. To add the JAR files, do the following:

1. Log in to the host machine where SOA is installed.
2. Go to the ORACLE\_HOME of the EM Agent installed on the same host.
3. Open file '\$AGENT\_ORACLE\_HOME/sysman/config/emd.properties' as shown in figure.
4. Check whether the following jar files have been added to the CLASSPATH property in the file. The property must manually update with the BPEL-specific jar file names to get the process listing.
  - \$BPEL\_SERVER\_ORACLE\_HOME/opmn/lib/optic.jar
  - \$BPEL\_SERVER\_ORACLE\_HOME/bpel/lib/orabpel.jar
  - \$BPEL\_SERVER\_ORACLE\_HOME/bpel/lib/orabpel-common.jar
  - \$BPEL\_SERVER\_ORACLE\_HOME/bpel/lib/orabpel-thirdparty.jar
  - \$BPEL\_SERVER\_ORACLE\_HOME/j2ee/home/oc4jclient.jar
  - \$BPEL\_SERVER\_ORACLE\_HOME/j2ee/home/j2ee\_1.3.01.jar

---

**Note:** The \$BPEL\_SERVER\_ORACLE\_HOME should be replaced with the absolute path of the ORACLE\_HOME path of the application server where the SOA is installed.

---

5. Add the JAR filenames to CLASSPATH property. When adding the jar files to CLASSPATH, ensure that the BPEL home `optic.jar` property is the first value in the classpath.
6. Restart the Agent.

## 5.8.6 Retrieving the OPMN Port

To retrieve SOA Applications Server OPMN PORT details, follow these steps.

1. Open the configuration file `$SOA_ORACLE_HOME/opmn/conf/opmn.xml`.  
\$SOA\_ORACLE\_HOME corresponds to SOA Application server home location.

2. Identify the value of the request port attribute in the configuration file.

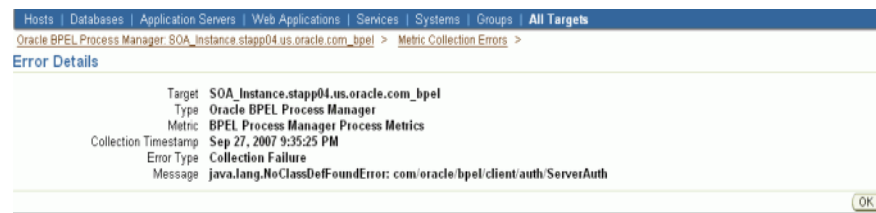
### 5.8.7 java.lang.NoClassDefFoundError Error

The following error occurs in the error details page when incorrect CLASSPATH setting is specified.

**Table 5–13 java.lang.NoClassDefFoundError Error - Workaround Steps**

Error Message	Workaround Steps
Error Type: Collection Failure Message: java.lang.NoClassDefFoundError: com/oracle/bpel/client/auth/ServerAuth  (See Figure 5–1)	<ol style="list-style-type: none"> <li>1. Identify the BPEL JARs that must be added to the CLASSPATH:  <pre> \$BPEL_SERVER_ORACLE_ HOME/opmn/lib/optic.jar \$BPEL_SERVER_ORACLE_ HOME/bpel/lib/orabpel.jar \$BPEL_SERVER_ORACLE_ HOME/bpel/lib/orabpel-common.jar \$BPEL_SERVER_ORACLE_ HOME/bpel/lib/orabpel-thirdparty.jar \$BPEL_SERVER_ORACLE_ HOME/j2ee/home/oc4jclient.jar \$BPEL_SERVER_ORACLE_ HOME/bpel/lib/j2ee_1.3.01.jar </pre> </li> <li>2. Verify the PATH for the JAR files to ensure that there are no errors in the path settings. If there are errors, then the Java classes in the JAR files will not be loaded.</li> <li>3. Restart the Agent to ensure that the new JAR files are loaded.</li> </ol>

**Figure 5–1 java.lang.NoClassDefFoundError Error**



### 5.8.8 javax.naming.NameNotFoundException Error

The following error occurs in the error details page when incorrect provider URL is specified.

**Table 5–14 *javax.naming.NameNotFoundException Error - Workaround Steps***

Error Message	Workaround Steps
<pre>oracle.sysman.emSDK.emd.fetchlet.FetchletException: java.lang.Exception: Failed to create "ejb/collaxa/system/ServerBean" bean; exception reported is: "javax.naming.NameNotFoundException:..."  (See Figure 5–2)</pre>	<ol style="list-style-type: none"> <li>1. Validate the format of the string.</li> <li>2. Verify if the OPMN port is correct.</li> <li>3. Verify if the &lt;oc4j_instance&gt; name is properly substituted with the correct value, that is, the OC4J name value. The format must be like this:  opmn:ormi://&lt;host&gt;:&lt;opmn_port&gt;:home/orapbel</li> </ol>

**Figure 5–2 *javax.naming.NameNotFoundException Error***



### 5.8.9 javax.naming.NamingException Error

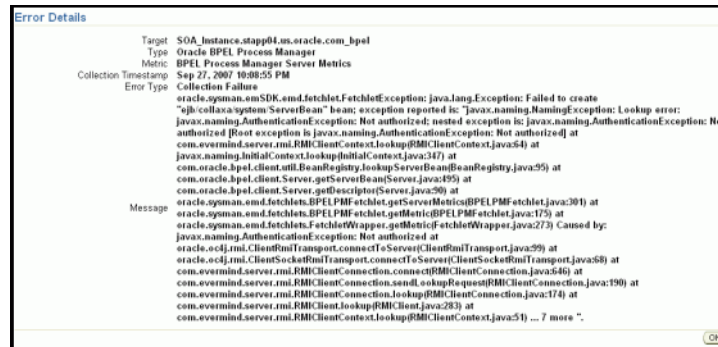
The following error occurs in the error details page when incorrect password is specified.

**Table 5–15 *javax.naming.NamingExceptionError - Workaround Steps***

Error Message	Workaround Steps
<pre>oracle.sysman.emSDK.emd.fetchlet.FetchletException: java.lang.Exception: Failed to create "ejb/collaxa/system/ServerBean" bean; exception reported is: "javax.naming.NamingException: Lookup error:..."  (See Figure 5–3)</pre>	<ol style="list-style-type: none"> <li>1. Validate the values specified for <b>BPEL Admin username</b> and <b>BPEL Password</b> fields in the Monitoring Configuration page. (Confirm the validity of credentials by using the same credentials to log in to the BPELConsole).</li> </ol>



Figure 5–3 *javax.naming.NamingException* Error



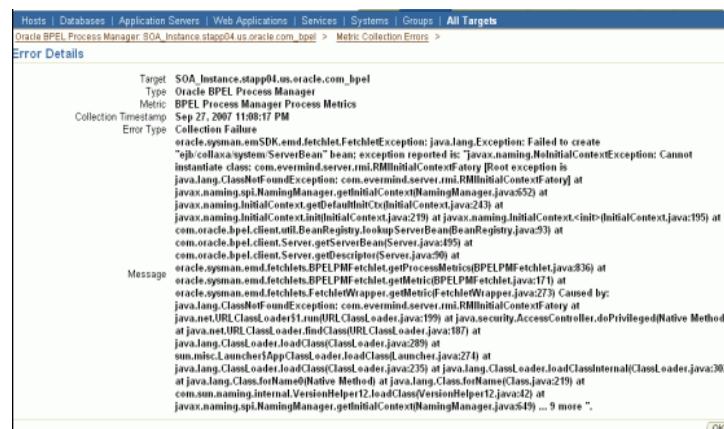
### 5.8.10 *javax.naming.NoInitialContextException* Error

The following error occurs in the error details page when incorrect *Initial Context Factory* value is specified.

Table 5–16 *javax.naming.NoInitialContextException* Error - Workaround Steps

Error Message	Workaround Steps
<pre>oracle.sysman.emSDK.emd.fetchlet.FetchletException: java.lang.Exception: Failed to create "ejb/collaxa/system/ServerBean" bean; exception reported is: "javax.naming.NoInitialContextException : Cannot instantiate class:...</pre> <p>(See Figure 5–4)</p>	<ol style="list-style-type: none"> <li>Provide the following value for the <b>Initial Context Factory</b> field in the Monitoring Configuration page: <code>com.evermind.server.rmi.RMIInitialContextFactory</code></li> </ol>

Figure 5–4 *javax.naming.NoInitialContextException* Error



### 5.8.11 Error While Creating BPEL Infrastructure Services

The following error occurs when you are creating a new BPEL infrastructure service.

**Table 5–17 *javax.naming.NoInitialContextException* Error - Workaround Steps**

Error Message	Workaround Steps
An error encountered while discovering the dependencies. Please try again.	<ol style="list-style-type: none"> <li data-bbox="878 270 1373 411">1. Apply patch 10849036 on the OMS and try creating the BPEL infrastructure service again:  com.evermind.server.rmi.RMIInitialContextFactory</li> </ol>

### 5.8.12 Metric Collection Errors for BPEL Process Manager Partner Link Metrics

The following metric collection error appears on the home page when you monitor BPEL 10.1.3.3 or 10.1.3.4 using Oracle Management Agent 10g Release 5 (10.2.0.5):

**Table 5–18 *Metric Collection Errors for BPEL Process Manager Partner Link Metrics* - Workaround Steps**

Error Message	Workaround Steps
<pre>java.rmi.UnmarshalException: Error deserializing return-value: java.io.InvalidClassException: javax.xml.namespace.QName; local class incompatible: stream classdesc serialVersionUID = -916876369326528164, local class serialVersionUID = -9120448754896609940 at com.oracle.bpel.client.util.ExceptionUtils.handleServer Exception(ExceptionUtils.java:82) at com.oracle.bpel.client.BPELProcessHandle.getDescriptor (BPELProcessHandle.java:207) at oracle.sysman.emd.fetchlets.BPELPMFetchlet.getPartner LinkMetrics(BPELPMFetchlet.java:873) at oracle.sysman.emd.fetchlets.BPELPMFetchlet.getMetric (BELPMFetchlet.java:235) at oracle.sysman.emd.fetchlets.FetchletWrapper.getMetric (FetchletWrapper.java:382)</pre>	Follow the workaround described in <i>My Oracle Support</i> note 735128.1.

---



---

# Discovering and Monitoring Oracle Service Bus

This chapter describes how you can discover and monitor Oracle Service Bus (OSB) using Enterprise Manager Grid Control.

In particular, this document covers the following:

- [Section 6.1, "Supported Versions"](#)
- [Section 6.2, "Understanding the Discovery Mechanism"](#)
- [Section 6.3, "Understanding the Discovery Process"](#)
- [Section 6.4, "Downloading One-Off Patches"](#)
- [Section 6.5, "Discovering Oracle Service Bus"](#)
- [Section 6.6, "Enabling Management Packs"](#)
- [Section 6.7, "Troubleshooting"](#)

## 6.1 Supported Versions

The following are the versions of OSB that are supported for monitoring in Enterprise Manager Grid Control.

**Table 6–1 Supported Versions**

Supported OSB Version	Oracle WebLogic Managed Server Deployed To	Supported Enterprise Manager Grid Control
Aqualogic Service Bus 2.6, 3.0, and Oracle Service Bus 10gR3	Oracle WebLogic Server 9.2, 10, 10.3	Enterprise Manager 10g Grid Control Release 5 (10.2.0.5) Enterprise Manager Enterprise Manager 11g Grid Control Release 1 (11.1.0.1) or higher

## 6.2 Understanding the Discovery Mechanism

The OSB deployed to Oracle WebLogic Managed Server is automatically discovered in Enterprise Manager Grid Control when that Oracle WebLogic Managed Server is discovered and added to Enterprise Manager Grid Control.

So, the discovery of OSB depends on the whether the Oracle WebLogic Managed Server is already being monitored in Enterprise Manager Grid Control.

- If Oracle WebLogic Managed Server is not being monitored in Grid Control, then first discover and add it to Grid Control; this will automatically discover the OSB that is deployed to it.
- If Oracle WebLogic Managed Server is already being monitored in Grid Control, then refresh the membership of the Oracle WebLogic Server Domain to which the Oracle WebLogic Managed Server belongs. This will automatically discover the OSB that is deployed to it.

For instructions to discover OSB, see [Section 6.5, "Discovering Oracle Service Bus"](#).

### 6.3 Understanding the Discovery Process

The following describes the overall process involved in discovering and monitoring OSB in Enterprise Manager Grid Control. Follow the instructions outlined against each step in this process to successfully discover and monitor your OSB.

**Table 6–2** *Discovery Process*

Step	Requirement	Description
1	Oracle Service Bus	Install the OSB software.
2	Enterprise Manager Grid Control	<p>Install Enterprise Manager 10g Grid Control Release 5 (10.2.0.5) or higher.</p> <p>Note that 10.2.0.5 is a patch set. Therefore, if you are installing Enterprise Manager Grid Control for the first time, then install the base release first and then apply the 10.2.0.5 patch set on it. For information about installing the base release of Enterprise Manager Grid Control, see the <i>Enterprise Manager Grid Control Installation and Basic Configuration Guide</i> available at: <a href="http://www.oracle.com/technology/documentation/oem.html">http://www.oracle.com/technology/documentation/oem.html</a></p> <p>Oracle recommends that you install the Enterprise Manager Grid Control components on a host that is different from the host where OSB is installed. For example, if OSB is installed on host1.xyz.com, then install and configure Oracle Management Service (OMS) and the Management Repository on host2.xyz.com.</p>
3	Oracle Management Agent (Management Agent)	<p>Install Oracle Management Agent 10g Release 5 (10.2.0.5) or higher on the host where OSB is installed.</p> <p>If OSB and Enterprise Manager Grid Control are on the same host, then you do not have to install a separate Management Agent. The Management Agent that comes with Enterprise Manager Grid Control is sufficient. However, if they are different hosts, then you must install a separate Management Agent on the host where OSB is installed. Alternatively, the Management Agent can also be installed on a different host and made to remotely monitor the OSB target on another host.</p> <p>You can install the Management Agent in one of the following ways:</p> <ul style="list-style-type: none"> <li>■ Invoke the installer provided with Enterprise Manager 10 Grid Control Release 2 (10.2.0.1), and select the installation type <b>Additional Management Agent</b>. Then apply the 10.2.0.5 Agent patch on it.</li> <li>■ Use the Agent Deploy application within the Enterprise Manager 10g Grid Control Release 5 (10.2.0.5) console.</li> <li>■ Use the full agent kit that is available at: <a href="http://www.oracle.com/technology/software/products/oem/htdocs/agentsoft.html">http://www.oracle.com/technology/software/products/oem/htdocs/agentsoft.html</a></li> </ul> <p>For information about installing the Management Agent, see the <i>Enterprise Manager Grid Control Installation and Basic Configuration Guide</i> available at: <a href="http://www.oracle.com/technology/documentation/oem.html">http://www.oracle.com/technology/documentation/oem.html</a></p>
4	One-Off Patches	The support for discovering and monitoring of OSB is enabled only when the one-off patches as described in <a href="#">Section 6.4, "Downloading One-Off Patches"</a> are applied to the WebLogic Server Home where OSB is running.
5	Discovery in Enterprise Manager Grid Control	OSB is automatically discovered when the Oracle WebLogic Server Domain to which it is deployed is discovered and added to Enterprise Manager Grid Control.

## 6.4 Downloading One-Off Patches

To view OSB services in Enterprise Manager Grid Control, you must apply the following patches to your OSB servers.

**Table 6–3 One-Off Patches**

Oracle Service Bus Version	ID	Password
Oracle Service Bus 2.6	EMMU	83XNT2D4
Oracle Service Bus 2.6.1	9NAF	TLZE4IPI
Oracle Service Bus 3.0	RPCD	JJEC2EY2
Oracle Service Bus 10.3.0	9HPA	FFLQHDHP
Oracle Service Bus 10.3.1	No Patch Required	

You can apply the patches in one of the following ways:

- **Online mode** - Using the SmartUpdate tool available with Oracle WebLogic Managed Server
- **Offline mode** - Manually copying the JAR files and classes to the OSB directories

For information about downloading these patches and applying them in either offline or online mode, see My Oracle Support note 804148.1. You can access My Oracle Support at:

<https://support.oracle.com/>

---

**Note:** After applying the patches, restart the WebLogic domain and all of the management agents monitoring the domain.

---

## 6.5 Discovering Oracle Service Bus

The OSB deployed to Oracle WebLogic Managed Server is automatically discovered in Enterprise Manager Grid Control when that Oracle WebLogic Managed Server is discovered and added to Enterprise Manager Grid Control.

So, before discovering OSB, identify whether the Oracle WebLogic Managed Server is already being monitored in Enterprise Manager Grid Control.

- If Oracle WebLogic Managed Server is not being monitored in Grid Control, then first discover and add it to Enterprise Manager Grid Control; this will automatically discover the OSB that is deployed to it.
- If Oracle WebLogic Managed Server is already being monitored in Grid Control, then refresh the membership of the Oracle WebLogic Server Domain to which the Oracle WebLogic Managed Server belongs. This will automatically discover the OSB that is deployed to it.

This section outlines the instructions for discovering OSB for the cases described above. In particular, this section covers the following:

- [Section 6.5.1, "Discovering OSB Deployed to Oracle WebLogic Managed Server That Is Not Monitored in Grid Control"](#)
- [Section 6.5.2, "Discovering OSB Deployed to Oracle WebLogic Managed Server That Is Already Monitored in Grid Control"](#)

## 6.5.1 Discovering OSB Deployed to Oracle WebLogic Managed Server That Is Not Monitored in Grid Control

To discover OSB deployed to Oracle WebLogic Manager Server that is not monitored in Grid Control, first discover that Oracle WebLogic Manager Server in Enterprise Manager Grid Control; this will automatically discover the OSB that is deployed to it. To discover Oracle WebLogic Manager Server, follow these steps:

1. Log in to Enterprise Manager Grid Control (as a SYSMAN user).
2. Click **Targets** and then **Middleware**.

Enterprise Manager Grid Control displays the Middleware page that lists all the middleware targets being monitored.

3. In the Middleware page, from the **Add** list, select **Oracle Fusion Middleware/WebLogic Server Domain** and click **Go**.

Enterprise Manager Grid Control displays the Add Oracle Fusion Middleware / WebLogic Server Domain wizard that captures the details of the Oracle WebLogic Server Domain to be discovered and monitored.

4. In the Add Oracle Fusion Middleware / WebLogic Server Domain wizard, specify the required details and click **Next** on each page to reach the end of the wizard.

For information about the details to be provided for each page of the wizard, click **Help** on each page.

5. In the last page of the Add Oracle Fusion Middleware / WebLogic Server Domain wizard, click **Finish** to complete the discovery process and add the target to Grid Control for monitoring purposes.

Enterprise Manager Grid Control displays the Middleware page with a confirmation message that confirms that the Oracle WebLogic Manager Server has been successfully added to Grid Control.

In the Middleware page that shows all the middleware targets being monitored, you can see the Oracle WebLogic Managed Server and the OSB you just added. Note that, at this point, OSB will be the last target listed in the table. To see it nested under its Oracle WebLogic Managed Server, click **Refresh** on this page. Alternatively, navigate to another tab or page, and then return to the Middleware page.

---

**Note:** After discovering and adding OSB to Enterprise Manager Grid Control, you can monitor its status from the OSB Home page. You can use the Services page to view a list of services.

For the first collection that happens, you will see the value "0" for all the metrics. This is an expected behavior. From the second collection onwards, you should see the actual metric values. However, if you still see the value "0", then perhaps the service monitoring is turned off. To resolve this issue, on the Services page, click **Launch Console** to access the OSB Console, and turn on the service monitoring and set the level to "pipeline" or "action".

---

## 6.5.2 Discovering OSB Deployed to Oracle WebLogic Managed Server That Is Already Monitored in Grid Control

To discover OSB deployed to Oracle WebLogic Managed Server that is already being monitored in Grid Control, refresh the membership of the Oracle WebLogic Server

Domain to which the Oracle WebLogic Managed Server belongs. This will automatically discover the OSB that is deployed to it.

To refresh the membership of the Oracle WebLogic Server Domain to which the Oracle WebLogic Managed Server belongs, follow these steps:

1. In Grid Control, click **Targets** and then **Middleware**.
2. On the Middleware page, from the table that lists all the middleware targets being monitored in Grid Control, click the name of the Oracle WebLogic Server Domain.
3. On the Oracle WebLogic Server Domain Home page, in the General section, click **Refresh Domain**. Enterprise Manager Grid Control displays the membership page that lists the OSB that is currently not being monitored. Click **OK**.

Enterprise Manager Grid Control refreshes the membership and returns to the Oracle WebLogic Server Domain Home page.

---

**Note:** On the Oracle WebLogic Server Domain Home page, in the Status section, the legend of the status pie chart may not show an increased count to indicate the newly added OSB target. This is an expected behavior because Enterprise Manager Grid Control takes a few seconds to reflect the membership details in this section.

---

4. Click the **Members** tab and verify whether the OSB has been added.

## 6.6 Enabling Management Packs

Besides monitoring the status of OSB, if you want to gain access to additional value-added features, then you must enable the Management Pack for SOA.

To enable the Management Pack for SOA:

1. Log in to Enterprise Manager Grid Control.  
Enterprise Manager Grid Control displays the home page.
2. From the top-right corner of the Home page, click **Setup**.  
Enterprise Manager Grid Control displays the Overview of Setup page.
3. In the Overview of Setup page, from the left-vertical menu pane, click **Management Pack Access**.  
Enterprise Manager Grid Control displays the Management Pack Access page.
4. In the Management Pack Access page, from the Search list, select **Oracle Service Bus**.  
Enterprise Manager Grid Control lists all the Oracle Service Bus targets being monitored.
5. From the table, for the Oracle Service Bus target you are interested in, enable the SOA Management Pack Enterprise Edition and click **Apply**.

## 6.7 Troubleshooting

This section describes the errors you might encounter while discovering OSB, and the workaround steps you can follow to resolve each of them.

This section covers the following:

- [Section 6.7.1, "Required Patches Missing"](#)

## 6.7.1 Required Patches Missing

The following error occurs when you try to discover OSB from an Oracle WebLogic Admin Server that has not been patched with the required one-off patches.

**Table 6–4** *oracle.sysman.emSDK.emd.fetchlet.FetchletException Error - Workaround Steps*

Error Message	Workaround Steps
oracle.sysman.emSDK.emd.fetchlet.FetchletException: The MBean is not available on the OSB instance. The required EM plug in patch should be missing on OSB instance.	Apply the one-off patches as described in <a href="#">Section 6.4, "Downloading One-Off Patches"</a> .

## 6.7.2 System and Service

The following error occurs if configuration information has not been collected for the selected Application Server.

**Table 6–5** *Create System and Service Error - Workaround Steps*

Error Message	Workaround Steps
An error encountered while discovering the dependencies. This may occur if some configuration information is missing. Check whether the configuration information was collected for the dependent targets and then try again.	Collect the latest configuration data by navigating to the Application Server Home page and clicking <b>Administrations</b> and then select <b>Last Collected</b> .



---

---

## Discovering and Monitoring SOA Suite 11g

This chapter describes how you can discover and configure the components of the SOA Suite 11g using Enterprise Manager Grid Control.

In particular, this document covers the following:

- [Section 7.1, "New Features in This Release"](#)
- [Section 7.2, "Supported Versions"](#)
- [Section 7.3, "Understanding the Discovery Process"](#)
- [Section 7.4, "Discovering SOA Suite 11g"](#)
- [Section 7.5, "Post Discovery Steps"](#)
- [Section 7.6, "Creating Reports"](#)
- [Section 7.7, "Provisioning SOA Artifacts and Composites"](#)
- [Section 7.8, "Troubleshooting"](#)

### 7.1 New Features in This Release

The new features that have been introduced in the 11.1 version of the SOA Suite are:

- **SOA Home Page:** The SOA Home page provides a single view of all the SOA related artifacts being monitored by Enterprise Manager Grid Control. You can access this page by clicking on the SOA Home related link in the Middleware page.
- **Instance Tracing:** This feature allows to trace the message flow across SOA Composites and SOA Infrastructure instances monitored by Enterprise Manager Grid Control. The message flow can be traced across servers, clusters, and WebLogic domains.
- **Historic Chart (Home Pages):** The Home pages show real time and historic metric data. You can click on a specific metric value to view its historic trend.

### 7.2 Supported Versions

The following are the versions of the SOA Suite 11g that are supported in Enterprise Manager Grid Control 11g Release1:

- Oracle SOA Suite 11g PS1
- Oracle SOA Suite 11g PS2

## 7.3 Understanding the Discovery Process

The following describes the overall process involved in discovering and monitoring SOA Suite 11g in Enterprise Manager Grid Control. Follow the instructions outlined against each step in this process to successfully discover and monitor the SOA Suite.

**Table 7–1 Understanding the Discovery Process**

Oracle SOA Suite Version	Application Server Deployed To	Discovery Mechanism	Process
Oracle SOA Suite 11.1 PS2	Oracle WebLogic Managed Server	Manual Discovery	<ol style="list-style-type: none"> <li>1. First, manually discover Oracle WebLogic Managed Server. For procedures, see <a href="#">Section 5.6.2.1, "Discovering Oracle WebLogic Managed Server"</a>.</li> <li>2. To monitor the SOA Suite, you can use an agent running locally on the Administration Server of the WebLogic domain or a remote management agent running on another host that is not part of the WebLogic domain. <b>Note:</b> If you use a remote agent to monitor the SOA Suite, the Instance Tracing and View Recoverable Instances features are not supported.</li> <li>3. To ensure the all the metric data is collected, add the <code>soa-infra-mgmt.jar</code> and the <code>oracle-soa-client-api.jar</code> files to the <code>\$AGENT_HOME/sysman/jlib/extjlib</code> directory (the Agent Home directory). If the <code>extjlib</code> directory does not exist, it can be created. This step is required only if you are using a remote agent to monitor the SOA Suite.</li> </ol>

## 7.4 Discovering SOA Suite 11g

This section describes the procedure for discovering the SOA Suite 11g. You can use a local or a remote Management Agent to perform the discovery process. In the case of discovery using a local agent, you need to use a Management Agent that is running on the same host as the Administration Server.

1. Log in to Oracle Enterprise Manager Grid Control.
2. Click **Targets** and then **Middleware**.  
Oracle Enterprise Manager Grid Control displays the Middleware page that lists all the middleware targets being monitored.
3. In the Middleware page, from the **Add** list, select Oracle Fusion Middleware / WebLogic Domain and click **Go**. Specify the Administration Server Host, Port, User Name, Password, Agent (local or remote) and the JMX Protocol and click **Continue**.
4. You will return to the Middleware page. You will see the SOA instances under the WebLogic Domain.

---

**Note:** New SOA Composites are not displayed by default. To display all the SOA Composites, navigate to the Home page of the WebLogic Server target and select the **Refresh Domain** option from the menu.

---

### 7.4.1 Discovering the SOA Suite 11g Using a Remote Agent

You can discover the SOA Suite 11g using a remote agent which may be running on a host that is different from the host on which the Administration Server is running. In this case, you may not be able to perform certain operations like instance tracing and viewing recoverable instances. To ensure that all metrics are collected, you must copy

the following jar files to the agent home path at \$AGENT\_HOME/sysman/jlib/extjlib (if the extjlib directory does not exist, it can be created)

- soa-infra-mgmt.jar
- oracle-soa-client-api.jar

## 7.5 Post Discovery Steps

After discovering the SOA Suite 11g, you must perform the following additional configuration steps:

1. Set the instance state in the Common Properties page:
  - a. Click **Fusion Middleware Control** on the SOA Infrastructure Home Page.
  - b. Navigate to the Home page of the SOA Infrastructure target.
  - c. Select **Common Properties** from the SOA-Infra drop-down menu.
  - d. On the Common Properties page, select the **Capture Composite Instance State** check box.
2. Set the SOA database details like the host name, port, and credentials.
  - a. Navigate to the Middleware page in Enterprise Manager Grid Control.
  - b. Select a SOA Infrastructure home from the list and click **Configure**.
  - c. The SOA Infrastructure Home page is displayed. Click **Monitoring Configuration** from the SOA-Infra drop-down menu.
  - d. Set the SOA database details in the Monitoring Configuration page.
3. Set preferred credentials for the WebLogic Domain.
  - a. Click the **Preferences** link in the top-right corner of the main page in Enterprise Manager Grid Control.
  - b. In the Preferences page, click **Preferred Credentials** on the right pane.
  - c. Select the Oracle WebLogic Domain target and click the **Set Credentials** icon.
  - d. On the Oracle WebLogic Domain Preferred Credentials page, in the Target Credentials section, specify the administrator credentials and host credentials and click **Apply**.
  - e. Similarly, on the Preferences page, select the **Host** target and click the **Set Credentials** icon.
  - f. On the Host Preferred Credentials page, in the Target Credentials section, specify the normal credentials and privileged credentials. Then click **Apply**.

### 7.5.1 Configuring Instance Tracing

To enable Instance Tracing for any SOA Infrastructure instances involved in executing composite instances:

1. Follow the sequence listed under Step 3 of the [Post Discovery Steps](#) section.
2. To view the state of the listed instances, enable the Capture Composite State flag by following the sequence listed Step 1 of the [Post Discovery Steps](#) section.

## 7.5.2 Viewing Application Dependency and Performance (ADP) Metrics

If the SOA instance is being monitored by the ADP Manager, additional metrics such as Arrival Rate, Minimum, Maximum, and Average Response Time will be collected.

**Tip:** The ADP Manager must be registered before it can collect the metric data. For details on registering the ADP Manager, see [Section 10, "Installing and Configuring Application Dependency and Performance"](#)

## 7.6 Creating Reports

This section describe the procedure to create SOA Reports.

1. To access the reports wizard, click the **Reports** tab on the Enterprise Manager Console Home page.
2. On the Report Definitions tab, select the **SOA Composite** target type and click **Go**. The out-of-box SOA reports are displayed under the SOA Performance Reports section.
3. Select a report from the section (for example, you can select **Pending Instance Statistics**) and click **Create Like**. The Create Report Definition page is displayed.
4. In the General page, enter the following details:
  - a. Enter the BPEL Process Name as the title.
  - b. Click the Set Time Period to set the time interval for the report.
  - c. Click the **Run report using target privileges of the Report Owner (SYSMAN)** checkbox in the Privileges section.
5. Click the **Elements** tab and click the **Set Parameters** icon for the Pending Instance Statistics Element in the table.
6. In the Set Parameters page, click the torch icon to select a Composite Name. The Result Set Size with default values for the Pending Instance Statistics report is displayed.
7. Select a Component Name from the list, enter the Result Set Size and click **Continue** to return to the Elements page.
8. The selected target name is displayed in the Elements table.
9. To schedule periodic report generation, click the **Schedule** tab.
10. Specify the schedule type and other details and click **OK**.
11. You will return to the Report Home page where the newly scheduled report is displayed in the table. Click the report name to view the details.

## 7.7 Provisioning SOA Artifacts and Composites

The SOA Artifacts Deployment Procedure allows you to:

- Provision SOA Artifacts from a reference installation or from a gold image
- Create a gold image of the SOA Artifacts
- Provision SOA Composites either from the Software Library or from another accessible location.

For more details on the SOA Artifacts Deployment Procedure, see the *Enterprise Manager Administrator's Guide for Software and Server Provisioning and Patching*.

## 7.8 Troubleshooting

This section describes the errors you might encounter while discovering the SOA Suite 11g and the workaround steps you can follow to resolve each of them.

This section covers the following:

- [Section 7.8.1, "Discovery"](#)
- [Section 7.8.2, "Monitoring"](#)
- [Section 7.8.3, "Instance Tracing"](#)
- [Section 7.8.4, "Faults"](#)
- [Section 7.8.5, "Application Dependency and Performance Integration"](#)
- [Section 7.8.6, "Reports"](#)
- [Section 7.8.7, "Systems and Services"](#)
- [Section 7.8.8, "BPEL Recovery"](#)

### 7.8.1 Discovery

The following error occurs when the SOA instances are being discovered.

**Table 7-2 Error Message:**

Error Message	Workaround Steps
New SOA Composite deployed on the SOA Server from JDeveloper are not displayed automatically in Enterprise Manager Grid Control.	To discover the newly deployed SOA Composites in Enterprise Manager Grid Control, you must run the <b>Refresh Farm</b> menu option for the associated Weblogic Domain.

### 7.8.2 Monitoring

The following error occurs when the collection frequency causes a delay in the collection of configuration data.

**Table 7-3 Error Description**

Error Description	Workaround Steps
All metrics are not displayed.	Enterprise Manager Grid Control uses the Management Agent to collect metric data. For the first collection, the agent may need 15 minutes to upload the metric data.

### 7.8.3 Instance Tracing

The following error occurs when the instance is traced.

Instance Search Fails - Same reason as BPEL first column. If Management Agent is down or unreachable.

**Table 7-4 Error Message:**

Error Message	Workaround Steps
Instance Tracing Job Fails	<ol style="list-style-type: none"> <li>1. Navigate to the Jobs page and locate the Instance Tracing job (Composite Name + Instance ID + Timestamp in ms) and view the output to identify the step that has failed.</li> <li>2. Resolve the issue and run the job again by clicking <b>Retry</b> on the Jobs page.</li> <li>3. Navigate to the Instance Tracing page to view the trace results. You can also submit a new job by running the Trace Instance option on the Instance Tracing page.</li> </ol>

## 7.8.4 Faults

The following errors occur when:

- All instances with faults are not displayed as only the last 10 values are collected.
- The most recently collected fault instances do not appear in the Faults and Messages page.

**Table 7-5 Error:**

Error Description	Workaround Steps
All instances with faults are not populated in Enterprise Manager Grid Control.	By default, you can only view the latest 10 faults collected during the last 15 minutes. To view additional faults, navigate to Fusion Middleware by clicking the link in the General section on the target Homepage.

## 7.8.5 Application Dependency and Performance Integration

When you click on the Application Dependency and Performance link in the SOA Instance Home page, you may see a blank page. This error may occur if:

- Application Dependency and Performance is not being used to monitor the SOA instance.
- Application Dependency and Performance has not been registered in Enterprise Manager Grid Control.

**Table 7-6 Error Message:**

Error Message	Workaround Steps
Missing ADP Data - Add the metrics - and add one for blank page.	To monitor data collected using ADP, the ADP Manager must be registered and configured.

## 7.8.6 Reports

This section lists report related errors.

**Table 7-7 Error:**

Error Description	Workaround Steps
Report generation fails due to invalid database details.	<ol style="list-style-type: none"> <li>1. Navigate to the All Targets page.</li> <li>2. Select the SOA Infrastructure target on which the specific SOA Composite has been deployed and click <b>Configure</b>.</li> <li>3. In the Monitoring Configuration page, specify the database connection details and the credentials and click <b>OK</b>.</li> </ol>

**Table 7-7 (Cont.) Error:**

Error Description	Workaround Steps
No targets found message for Oracle SOA Composite Reports.	You cannot use the out-of-box reports directly. You must use the Create Like option to generate custom reports based on the SOA Composite Target type.
Report generation fails due to invalid host details.	Set valid credentials for the host target on which the SOA Infrastructure instance is running.

## 7.8.7 Systems and Services

The following error occur when you try to refresh a service that has not been created.

**Table 7-8 Error Message:**

Error Message	Workaround Steps
Create Service option does not work.	System and service creation depends on the configuration collection of the SOA Infrastructure and related targets. Check the log file for details.
Refresh Service option does not work.	The Refresh Service function works for an existing Infrastructure service. In case the service does not exist, it should be created using the Create Service menu option.

## 7.8.8 BPEL Recovery

The following error occurs when invalid credentials are provided.

**Table 7-9 Error Message:**

Error Message	Workaround Steps
Invalid Host and WebLogic Domain Credentials	For the BPEL Recovery functionality to work, the host credentials and WebLogic Domain credentials must to be available in the preferred credential store. Set the valid credentials and try again.
BPEL Processes disappear on the BPEL Process tab.	<p>This error may occur due to the missing parameter required for the DMS metric. To resolve this issue:</p> <ol style="list-style-type: none"> <li>1. Update the AGENT_HOME/sysman/config/emd.properties file.</li> <li>2. Grep for property agentJavaDefines in the file</li> <li>3. Append the following string:  <pre>-Doracle.dms.gatherers=15000,2:300000,3:360000,3</pre> </li> <li>4. Bounce the agent.</li> </ol> <p>Example:</p> <pre>agentJavaDefines=-Doracle.dms.refresh.wait.time 1000 -DUrlTiming.UseJSSE=true -Dnetworkaddress.cache.ttl=1800-Djava.awt.headle ss=true-Doracle.dms.gatherers=15000,2:300000,3:3 600000,3</pre>





# Part III

---

## Installing and Configuring Performance Management Components

The chapters in this part explain how to install the performance management components.

Part II contains the following chapters:

- [Chapter 8, "Installing and Configuring Performance Management Components"](#)
- [Chapter 9, "Configuring JVM Diagnostics"](#)
- [Chapter 10, "Installing and Configuring Application Dependency and Performance"](#)
- [Chapter 11, "Configuring Request Monitoring"](#)



---

---

# Installing and Configuring Performance Management Components

This chapter provides an overview of the process to follow to install and configure the application performance management components provided with Enterprise Manager Grid Control. These components include:

- **JVM Diagnostics Manager**  
This component allows you to monitor Java applications and diagnose performance problems directly in your production environment.
- **Application Dependency and Performance**  
This feature analyzes Java EE and SOA applications to capture relationships among various application building blocks, enabling more efficient management of distributed applications.
- **Request Monitoring**  
This feature provides end-to-end visibility into requests and helps localize and diagnose end-user performance problems.

These components are installed and configured on an existing Enterprise Manager Grid Control installation. This means that you must have the core components of Enterprise Manager Grid Control (Oracle Management Service, Oracle Management Agents, and Oracle Management Repository) fully installed and configured before installing or configuring the performance management components.

The recommended installation and configuration processes are as follows:

1. **Install Enterprise Manager Grid Control.**  
The installation process is fully documented in the *Enterprise Manager Grid Control Basic Installation Guide*.  
Ensure that all prerequisites defined in the document have been met before beginning the installation.
2. **Configure JVM Diagnostics.**  
This process is documented in [Chapter 9, "Configuring JVM Diagnostics"](#).
3. **Install and Configure Application Dependency and Performance.**  
This is documented in [Chapter 10, "Installing and Configuring Application Dependency and Performance"](#).
4. **Configure Request Monitoring.**  
This is documented in [Chapter 11, "Configuring Request Monitoring"](#).



---

## Configuring JVM Diagnostics

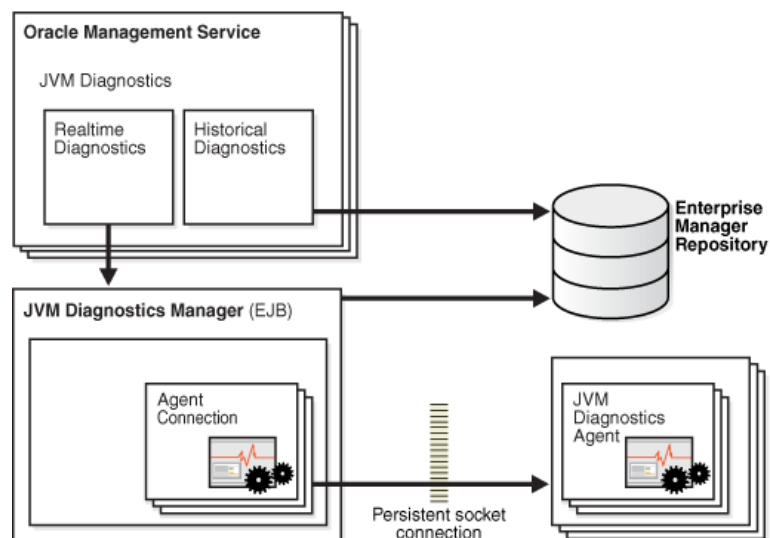
This chapter describes the process of configuring and enabling the JVM Diagnostics feature. It contains the following sections:

- Section 9.1, "JVM Diagnostics Architecture"
- Section 9.2, "Prerequisites"
- Section 9.3, "Deploying the JVM Diagnostics Manager"
- Section 9.4, "Scaling Out by Deploying Multiple JVM Diagnostics Managers"
- Section 9.5, "Deploying the JVM Diagnostics Agent"
- Section 9.6, "Undeploying the JVM Diagnostics Agent"
- Section 9.7, "Deploying the Database Agent"
- Section 9.8, "Creating a Special Repository User for Loading JVM Diagnostics Heaps"

### 9.1 JVM Diagnostics Architecture

The following diagram shows the JVM Diagnostics Architecture.

*Figure 9–1 JVM Diagnostics Architecture*



The JVM Diagnostics Manager runs as an EJB on a WebLogic Server. The JVM Diagnostics Agent is deployed on the targeted JVM (E.g.: the one running a production WebLogic Server). It collects real-time data and transmits it to the JVM Diagnostics Manager. The communication between the JVM Diagnostics Manager and Agent can be a secure (SSL) or non-secure connection.

---

---

**Note:** This diagram shows the deployment of a single JVM Diagnostics Manager. This is not applicable if you are deploying multiple managers as described in [Section 9.4, "Scaling Out by Deploying Multiple JVM Diagnostics Managers"](#).

---

---

## 9.2 Prerequisites

- The `DeployAD4Jmanager.sh` script must be started from the `$ORACLE_HOME/oms11g/ad4j` directory.
- The port number specified by user to create a new managed server should not be in use.
- You must have appropriate login credentials to access and execute the `DeployAD4Jmanager.sh` script.
- One-off patch prerequisites - TBD

## 9.3 Deploying the JVM Diagnostics Manager

The `DeployAD4Jmanager.sh` script is used to deploy the JVM Diagnostics Manager. This script does the following:

- Create (clones) the managed server from the existing OMS server.
- Deploys the `jamanager.ear` file on the new managed server.

This section describes the following:

- [Section 9.3.1, "Running the DeployAD4Jmanager.sh Script"](#)
- [Section 9.3.2, "Deploying the JVM Diagnostics Agent on 64-bit JVMs"](#)

### 9.3.1 Running the DeployAD4Jmanager.sh Script

To install the JVM Diagnostics Manager, follow the steps given below:

1. Run the `DeployAD4Jmanager.sh` script from the command prompt:

```
$ ./DeployAD4JManager.sh
```

You will see the following:

```
Welcome to AD4J Manager Installation script
Please follow the guided instructions to install AD4J manager
Setting up environment variables
Environment variable used in the script:
ORACLE_HOME=/u01/atejaswy/Oracle/Middleware/oms11g
MW_HOME=/u01/atejaswy/Oracle/Middleware
WL_HOME=/u01/atejaswy/Oracle/Middleware/wlserver_10.3
MODULES_DIR=/u01/atejaswy/Oracle/Middleware/modules
Delete previous existing ad4j directory..
Done
Unzipping ad4j.zip.....
Archive:  ad4j.zip
```

```
creating: ad4j/  
inflating: ad4j/ad4jjobtype.xml  
inflating: ad4j/jamagent.war  
inflating: ad4j/jamagent.ear
```

2. You are now prompted for the WebLogic Server host name. Press **Enter** to select the default values. The following prompts are displayed:

```
Enter weblogic host name:
```

Enter the host name or the IP address of the machine on which the WebLogic server is running.

```
Is secured connection (uses https)? [Default:[y]es]
```

Press **Enter** to indicate that it is a secure connection.

```
Enter weblogic ssl port number
```

Enter the ssl(https) port number if user selects https, otherwise its http port number.

```
Enter weblogic admin username
```

Enter the name of the admin user.

```
Enter weblogic admin password
```

Enter the password for the admin user.

3. The WebLogic server details that you have selected are displayed. You are then prompted for the details of the new managed server. Press **Enter** to accept the default values. The following prompts are displayed:

```
Enter managed server name:
```

Enter the name of the new managed server. The default name is EMAD4JMANAGER.

```
Enter managed server machine name:
```

Enter the name of machine on which managed server is to be created. The default is EMGC\_MACHINE1.

```
Enter managed server listen address
```

Enter the listen address of the managed server. The default is the name of the current machine.

```
Enter managed server listen port
```

Enter the user configured port number for the managed server. The default value is 3800.

```
Enter managed server SSL listen port
```

Enter the user configured ssl port number for the managed server. The default value is 3801.

```
Enter existing OMS server name
```

Enter the name of the OMS server.

```
Enter existing EMGC_DOMAIN path
```

Enter the location of the EM domain.

Enter existing INSTANCE\_HOME path

Enter the directory in which the OMS server has been installed.

Do you want secure communication between AD4J Agent and AD4J Manager (y/n)  
[Default:n]

Enter **Y** if secure communication should be supported.

4. If you enter Y, you are prompted for the following:

Enter the absolute location of the wallet

Enter the complete path of the wallet including the wallet name.

Enter wallet password

Enter the password for the wallet. This will ensure that there is secure communication between the JVM Diagnostics Agent and the JVM Diagnostics Manager.

5. You are prompted to continue with the installation as follows:

Do you wish to continue with the installation of AD4J Manager (y/n) [Default:y]

Press **Enter** to continue. The following details are displayed:

Initializing WebLogic Scripting Tool (WLST) ...

Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

Deployment Summary:

```
WEBLOGICHOSTNAME dadvmi0105
WEBLOGICPORTNUMBER: 7101
USERNAME: weblogic
MACHINENAME: EMGC_MACHINE1
ORACLEHOME: /u01/atejaswy/Oracle/Middleware/oms11g
WLHOME: /u01/atejaswy/Oracle/Middleware/wlserver_10.3
MODULEHOME: /u01/atejaswy/Oracle/Middleware/modules
SERVERNAME ad4jmanager
LISTENADDRESS dadvmi0105
LISTENPORT 3800
SSL LISTENPORT 3801
Connecting to t3s://dadvmi0105:7101 with userid weblogic ...
Successfully connected to Admin Server 'EMGC_ADMINSERVER' that belongs to
domain 'EMGC_DOMAIN'.
Location changed to edit tree. This is a writable tree with
DomainMBean as the root. To make changes you will need to start
an edit session via startEdit ().
For more help, use help(edit)
You already have an edit session in progress and hence WLST will
continue with your edit session.

Started edit session, please be sure to save and activate your
changes once you are done.
Saving all your changes ...
Saved all your changes successfully.
Activating all your changes, this may take a while ...
The edit lock associated with this edit session is released
```



```

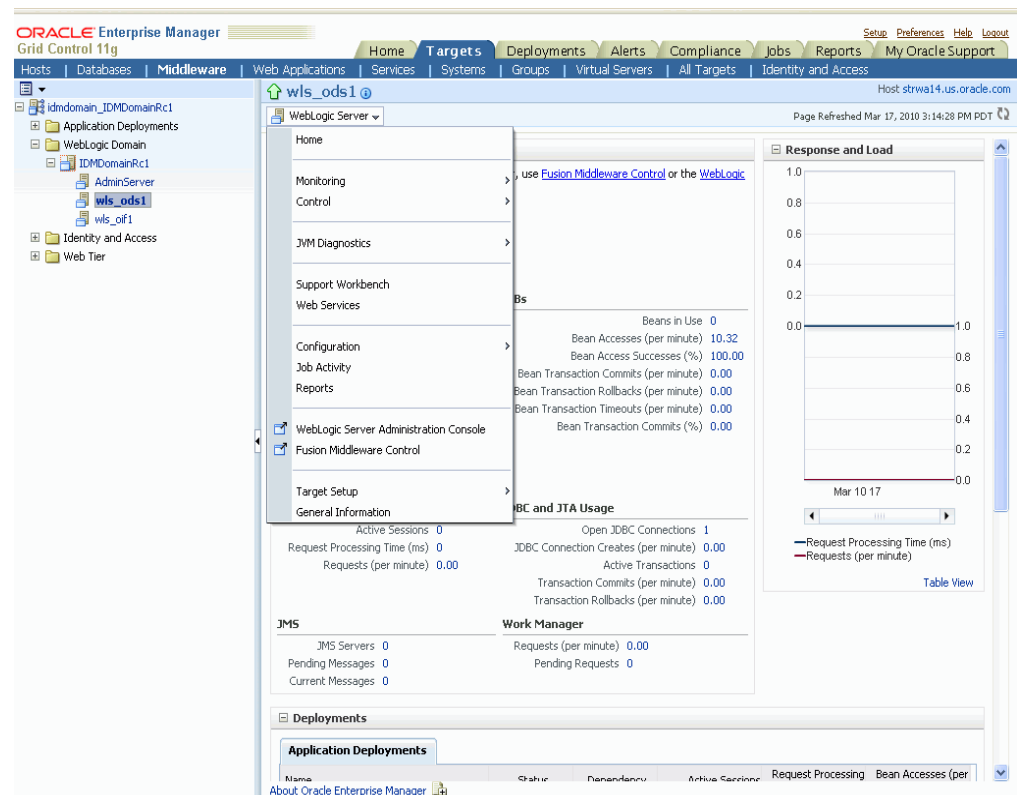
once the activation is completed
Activation completed
Starting server ad4jmanager
.....
.....
Server with name ad4jmanager started successfully
Deploying application from
/u01/atejaswy/Oracle/Middleware/oms11g/ad4j/./ad4j/jamanager.ear to targets
ad4jmanager (upload=false) ...
<Jan 8, 2010 6:33:16 AM PST> <Info> <J2EE Deployment SPI> <BEA-260121>
<Initiating deploy operation for application, ad4jmanagerapp [archive:
/u01/atejaswy/Oracle/Middleware/oms11g/ad4j/./ad4j/jamanager.ear], to
ad4jmanager .>
.....Completed the deployment of Application with status
completed
Current Status of your Deployment:
Deployment command type: deploy
Deployment State      : completed
Deployment Message    : no message
Exiting WebLogic Scripting Tool.

```

The JVM Diagnostics Manager has now been installed. To start using this feature, follow these steps:

1. Login to Enterprise Manager Grid Control.
2. Click **Targets** and the **Middleware** tab to navigate to the Middleware page.
3. Click on a **WebLogic Domain** target in the list. The following screen is displayed.

**Figure 9–2 WebLogic Domain - JVM Diagnostics Page**



4. To start using JVM Diagnostics, select the JVM Diagnostics option from the WebLogic Domain drop-down menu and choose the appropriate option.

### 9.3.2 Deploying the JVM Diagnostics Agent on 64-bit JVMs

The JVM Diagnostics Manager can be deployed on 32-bit JVMs. If you try to deploy the JVM Diagnostics Manager on a 64-bit JVM, the `jammanager.ear` will fail and you will see this error:

```
JAM Console: Only 32 bit JVM is supported. The shared lib
might not be loaded on this platform
>>java.library.path: /scratch/skbalakr/Oracle/Middleware/oms11g/lib
JAM Console: loadNative Exception loading [/tmp/libJamConsole.so.1]
/tmp/libJamConsole.so.1: ld.so.1: java: fatal:
/tmp/libJamConsole.so.1: wrong ELF class: ELFCLASS32 (Possible cause: architecture
word width mismatch)
java.lang.UnsatisfiedLinkError: /tmp/libJamConsole.so.1: ld.so.1: java: fatal:
/tmp/libJamConsole.so.1: wrong ELF class:
ELFCLASS32 (Possible cause: architecture word width mismatch)
at java.lang.ClassLoader$NativeLibrary.load(Native Method)
at java.lang.ClassLoader.loadLibrary0(ClassLoader? .java:1778)
at java.lang.ClassLoader.loadLibrary(ClassLoader? .java:1674)
at java.lang.Runtime.load0(Runtime.java:770)
at java.lang.System.load(System.java:1003)
at oracle.sysman.e2e.model.ad4j.remote.Jam.init(Jam.java:597)
at
oracle.sysman.e2e.model.ad4j.remote.servlet.AD4JManagerServlet.init(AD4JManagerSe
rvlet? .java:38)
```

To resolve this issue, you must change the value of the `JAVA_VM` variable as follows:

1. Open the `<DOMAIN_HOME>/bin/startWeblogic.sh` file.
2. Change value of `JAVA_VM` environment variable to `-d32`.
3. Run the `jammanager.ear` file again to deploy JVM Diagnostics Manager.

## 9.4 Scaling Out by Deploying Multiple JVM Diagnostics Managers

You can now deploy multiple JVM Diagnostics Managers and connect one or more agents to these managers. You can add a new JVM Diagnostics Manager by running the `DeployAD4JManager.sh` script on Unix and `DeployAD4Manager.cmd` on Windows. When you run the `DeployAD4JManager.sh` script, steps 1 to 3 as documented in the [Section 9.3.1, "Running the DeployAD4Jmanager.sh Script"](#) section are executed. After you enter the SSL listen port for the managed server, you will see the following prompt:

```
Enter an available port at which jammanager should listen [JamConsPort]:[3600]
```

Specify the listen port number for the JVM Diagnostics Manager. This is 3600 by default but can be changed. After entering the port number, continue with the rest of the steps as documented in [Section 9.3.1, "Running the DeployAD4Jmanager.sh Script"](#) to deploy the JVM Diagnostics Manager. After it has been deployed, you can check the status by clicking **Setup > JVMs & Managers**. [Figure 9-3, "Registered Managers Page"](#) is displayed.

Figure 9–3 Registered Managers Page

Manager ID	Machine	Port	Secure Communication	Build Version	Status
83	10.232.129.234	4502		4072	Inactive
84	10.232.129.234	3600		4097	Active
85	10.232.129.234	4402		4093	Inactive
86	10.232.129.234	17094		4075	Inactive
87	10.232.129.234	3601		4097	Active
88	10.232.129.234	4504		4095	Inactive
89	10.232.129.234	4604		4097	Active
90	10.232.129.234	4704		4097	Active

Manager ID	JVM ID	Pool	Machine	JVM Name	CPU%	ID	Heap Size	Heap Dump ID
84	244	Default	oc2195661.us.oracle.com	oc2195661.us.oracle.com/nagaw	0		nagaw:11055981	
90	253	Default	sdemv5457.us.oracle.com	sdemv	1		4604	510048
89	252	Default	devusn4.us.oracle.com	devusn4.us.oracle.com:1000	2		1080	516824
84	243	Default	usornad09.us.oracle.com	usornad09.us.oracle.com/oc2195661/home/default/istatd1	1	istatd	home/default	istatd1

You can see a list of all the JVM Diagnostics Managers and their current status on this page. You can also see the list of JVMs connected to each manager. To view the monitoring status of each manager, click **Setup > Monitoring**. Figure 9–4, "Monitoring Page" is displayed.

Figure 9–4 Monitoring Page

Manager ID	Machine	Port	Monitoring Status
84	10.232.129.234	3600	On
87	10.232.129.234	3601	On
89	10.232.129.234	4604	On
90	10.232.129.234	4704	On

Manager ID	Machine	Port	Monitoring Status
84	10.232.129.234	3600	On
87	10.232.129.234	3601	On
89	10.232.129.234	4604	On
90	10.232.129.234	4704	On

Number	Rule Type	Rule Value

A list of active managers with their monitoring status is displayed. The status of all the managers must be the same (on or off).

## 9.5 Deploying the JVM Diagnostics Agent

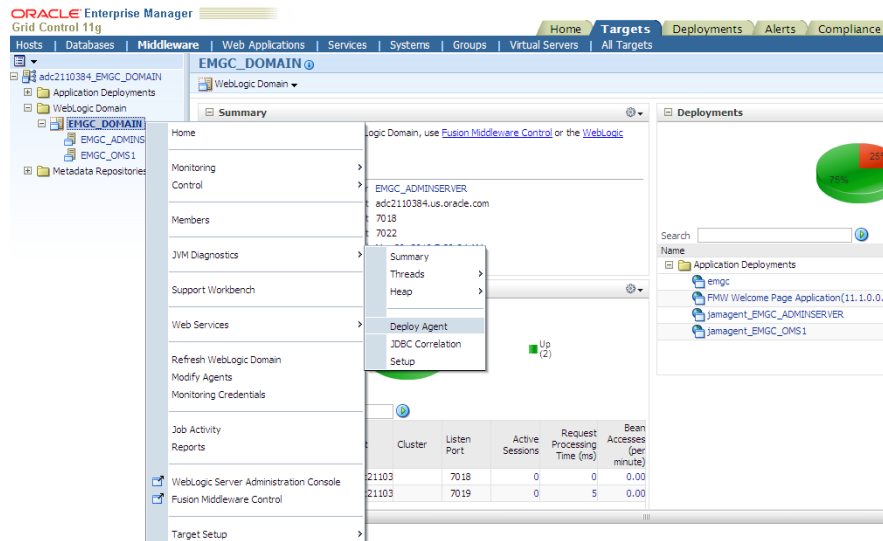
You can deploy the JVM Diagnostics agent manually or by using the automated deployment procedure.

### 9.5.1 Automated Deployment Procedure

To deploy the JVM Diagnostics Agent, follow these steps:

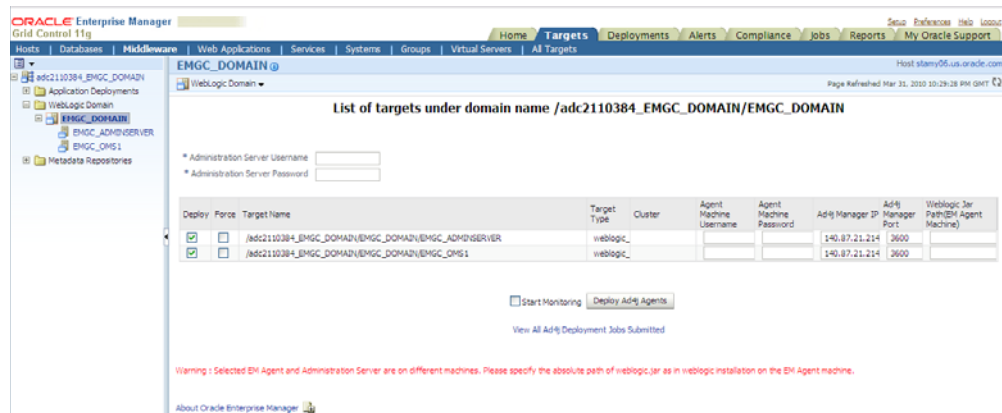
1. Select a WebLogic Domain target on the Middleware page.
2. Select the JVM Diagnostics option in the drop-down list and click **Deploy Agent** as shown below.

**Figure 9–5 Deploy Agent Selection Page**



3. The following screen is displayed.

**Figure 9–6 Deploy Agent Page**



4. Enter the user name and password for the Administration Server.
5. Check the **Deploy** checkbox to select the agents that are to be deployed and click **Deploy AD4J Agents** to deploy them.

## 9.5.2 Manual Deployment Procedure

To deploy the JVM Diagnostics Agent manually, follow these steps:

1. Navigate to the **Setup > Download > Java Agent** page.
2. Download the JVM Diagnostics Agent and deploy it manually onto the target JVM or Application Server.

### 9.5.2.1 Installing the JVM Diagnostics Agent on Standalone Java Applications

This section describes the procedure to install the JVM Diagnostics Agent on Standalone Java Applications. Before you install the agent, you must have installed

and configured the JVM Diagnostics Console. To install the agent, you must do the following:

- Include the `jamagent.war` in the `CLASSPATH`. See [Adding the JVM Diagnostics Agent Libraries, Classes to Classpath](#).
- Change the `java` call to call the `jamagent.jamrun` wrapper class. The wrapper class is required to monitor and diagnose Java programs that do not have a way of loading a servlet or another class. The following sections describe the original and modified `java` call:
  - [Original Java Call](#)
  - [Modified Java Call with JVM Diagnostics Agent](#)
- Customize the JVM Diagnostics Agent and specify the parameters as described in [Customizing the JVM Diagnostics Agent](#).

**9.5.2.1.1 Original Java Call** `java $JVM_OPTIONS $TARGET_CLASS $TARGET_CLASS_PARAMS`

Here, `JVM_OPTIONS` are the JVM properties and options.

For example:

```
-Xmx512M -Dweblogic.name
```

`TARGET_CLASS` is the program which being examined

`TARGET_CLASS_PARAMS` are parameters passed to the class

The idea is to run the target program through `jamagent.jamrun`. This is done by calling the `jamagent.jamrun`.

**9.5.2.1.2 Modified Java Call with JVM Diagnostics Agent** `java $JVM_OPTIONS jamagent.jamrun [$JAMAGENT_PARAMS_LIST] $TARGET_CLASS $TARGET_CLASS_PARAMS`

JVM Diagnostics uses default parameters. If you want to change any parameters, you can specify them as `name=value` pairs. To specify a different console and port, enter the following command:

```
jamconshost=<console hostname>, jamconsport=<console port>
```

To run the JVM Diagnostics Agent, you need to edit the script which starts your program. You are not required to do anything with the target program parameters as they are picked up and passed along to the program by `jamrun`.

**9.5.2.1.3 Sample Configuration** In this example, the `MainClass` is being called and the steps to edit the Java Call are shown below:

```
"$JAVA_HOME/bin/java" ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}
com.example.MainClass
```

The class being called is `com.example.MainClass`. There are no parameters. Insert the class and parameters before the class as follows:

```
"$JAVA_HOME/bin/java" ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}
jamagent.jamrun com.example.MainClass
```

If you want to change some JVM Diagnostics parameters, you can specify them as:

```
"$JAVA_HOME/bin/java" ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}
jamagent.jamrun jamconshost=myconsole01 jamisdaemon=true
jamjvmid=3001 com.example.MainClass
```

Refer to the [Frequently Asked Questions](#) section for more details on setting the `jamisdemon` parameter.

**9.5.2.1.4 Adding the JVM Diagnostics Agent Libraries, Classes to Classpath** You also need to add the `jamagent.war` directory to the `CLASSPATH`. Preferably, you must add this before the call to `java`.

Here we find a couple of lines above the call to `Java`.

```
CLASSPATH="example1.jar:example2.jar
```

You can add the `jamagent.war` file to the `CLASSPATH` by adding the following line before the original `Java` call.

```
CLASSPATH=${CLASSPATH} : /opt/jamagent/jamagent.war
```

**9.5.2.1.5 Target Parameters** About the `-Dweblogic` and other properties, you must insert `jamagent.jamrun` just before the calling program and its parameters. This should happen after all the JVM options and properties have been specified.

### 9.5.2.2 Customizing the JVM Diagnostics Agent

The JVM Diagnostics Agent WAR file contains an embedded `web.xml` deployment descriptor. This file contains the default values of the JVM Diagnostics Agent input parameters. You can change the parameters if you want. To do so, extract the `web.xml` file from the WAR archive and then update it.

To explode or update the WAR file, do the following:

- To extract the `web.xml` file from the WAR archive, run the following command:
 

```
jar xvf jamagent.war WEB-INF/web.xml
```
- Edit the `web.xml` file with your custom values for input parameters in any text editor. You may only change the values within the `<param-name>` fields of this file. No other changes are supported.
- Run the following command to place the new `web.xml` file back into the archive:
 

```
jar uvf jamagent.war WEB-INF/web.xml
```
- To remove the newly created directory, run the following command:
 

```
rm -rf WEB-INF
```

The following are the JVM Diagnostics Agent parameters.

**Table 9–1 Oracle JVM Diagnostics Agent Parameters**

Parameter	Default	Description
<code>jamconshost</code>	<code>localhost</code>	The server where console is running.
<code>jamconSPORT</code>	<code>3600</code>	The port where the console is listening for the agents.
<code>jamjvmid</code>	Application Server Port or <code>5555</code>	Identifies the specific JVM on the console when you have multiple JVMs on the same machine. For most app servers, this identifier is the port which the server is listening on (Web Server Port for Weblogic, Jserv port for Apache). If the application server port cannot be discovered then this value is used.
<code>jamconSretr</code>	<code>90</code>	If the console goes down, the agent will keep trying to reconnect. This parameter specifies the duration in seconds between each attempt. The default value is <code>90</code> seconds (15 minutes) between each try. If this parameter is set to <code>0</code> , then the agent will not try to reconnect.
<code>jamtimeOut</code>	<code>300</code>	Maximum time duration for a request. If a request takes longer time than this timeout, it is terminated.

**Table 9–1 (Cont.) Oracle JVM Diagnostics Agent Parameters**

Parameter	Default	Description
jamloglevel	3	Level of logging. Valid values range from 1 to 5.
jammaxbackoff	10	Some times we wait for other operations to finish (like GC or main thread initialization). The amount of sleep time between the retries increases exponentially till this number. When this number is reached, we give up and return with failure.
jamdelaystartup	0	This parameter is only for standalone programs. If you want to analyze the startup behavior of a monitored program, you can use this parameter. This specifies the number of seconds to wait before starting the target program.
jamisdaemon	false	This parameter is only for standalone programs. When using jamrun with standalone programs, the agent will normally exit when the main method in the target program completes. In some programs the threads might still be active after main completes. In such cases specify jamisdaemon=true to prevent the agent from exiting.

**9.5.2.2.1 Web.XML Contents**

```

<?xml version="1.0" ?>
<!DOCTYPE web-app PUBLIC "-//ABC, Inc.//DTD Web Application
2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
<servlet>
<servlet-name>jamagent</servlet-name>
<servlet-class>jamagent.jaminit</servlet-class>
<init-param>
<param-name>jamconshost</param-name>
<param-value>localhost</param-value>
<description>Default Jam Console host</description>
</init-param>
<init-param>
<param-name>jamconspport</param-name>
<param-value>3600</param-value>
<description>Jam console port</description>
</init-param>
<init-param>
<param-name>jamconsretr</param-name>
<param-value>90</param-value>
<description>Jam console number of retries</description>
</init-param>
<init-param>
<param-name>jamtimeout</param-name>
<param-value>900</param-value>
<description>Jam console timeout</description>
</init-param>
<init-param>
<param-name>jamloglevel</param-name>
<param-value>3</param-value>
<description>Jam log level</description>
</init-param>
<init-param>
<param-name>jammaxbackoff</param-name>
<param-value>10</param-value>
<description>Max time to wait for long operations</description>
</init-param>
<init-param>
<param-name>jamjvmid</param-name>
<param-value>Application Server Port</param-value>
<description> Unique Identifier for JVM. It will detect and use the WLS port by
default.</description>

```

```

</init-param>
<load-on-startup>1</load-on-startup>
</servlet>
</web-app>

```

## 9.6 Undeploying the JVM Diagnostics Agent

To undeploy the JVM Diagnostics Agent, follow these steps:

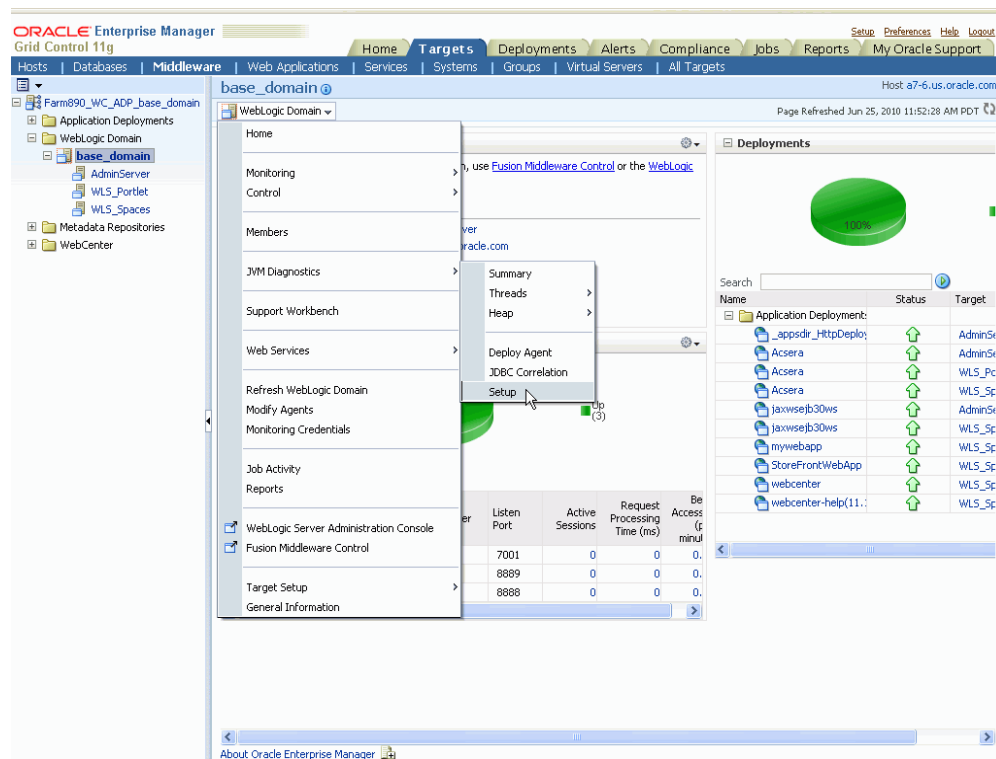
1. Log into the WebLogic Administration Console for the domain on which the JVM Diagnostics Agent is to be undeployed.
2. Navigate to the Deployments page and search for WAR deployments starting with the string jamagent\_.
3. Stop the deployments that you wish to deactivate. The JVM Diagnostics Agent for these deployments will be permanently removed.

## 9.7 Deploying the Database Agent

To deploy the database agent, follow these steps:

1. Select a WebLogic Domain target from the Middleware page.
2. Select the **JVM Diagnostics** option from the drop-down list and click **Setup** as shown in [Figure 9-7, "Setup Selection Page"](#).

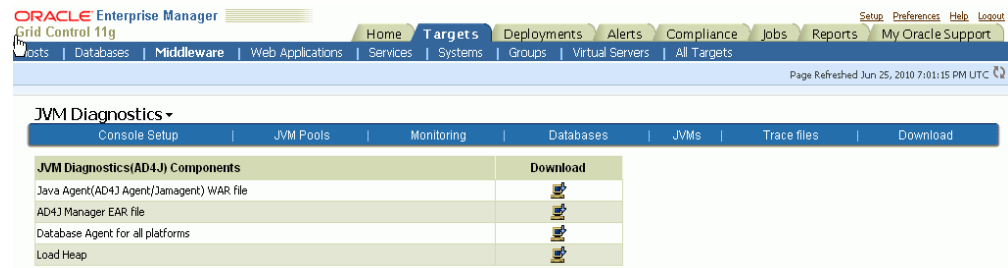
**Figure 9-7 Setup Selection Page**



3. Click the **Download** tab in the Setup page. [Figure 9-8, "Download Page"](#) is displayed.



Figure 9–8 Download Page



4. Download the database agent by clicking the download link for **Database Agent for All Platforms**.
5. After you have downloaded the database agent, return to the Setup page and click **Databases**.
6. As the OS user running the Oracle database, run the database agent on the machine on which the database is running. The database agent needs two parameters to be passed to it, which are:
  - the host where the console is running.
  - the port which is being used to accept connections from the agents (default value 3600).

To pass these parameters, enter the following command:

```
nohup dbagent jamconshost=console01 jamconspport=3600 >
dbagent.log 2>&1 &
```

You should see the following message in the `stdout.log` for the JVM D Manager's managed server (`gc_inst/user_projects/domains/GCDomain/servers/EMAD4JMANAGER/log` by default):

```
JAM Console: Agent connection from 192.168.1.31:59269,
[Hostname] jamdb.us.oracle.com
```

```
JAM Console: Received AJDBOracle|Oracle-Sun9i|oracle|SunOS
5.8|oracle|3|0|0 Oracle AD4J Console: New DB 0
```

---

**Note:** To run the database agent on an AIX machine, you must

- Login as `root` user and run the database agent.
  - Ensure that the `setuid` bit for the database agent is set to run as the `root` user.
- 

7. In the Databases page, click **Register New DB**. [Figure 9–9, "Add Database Information Page"](#) is displayed.

**Figure 9–9 Add Database Information Page**

8. Specify the information required to register the database.
  - The OS user you specify here must be the same as the OS user running the database agent.
  - The OS user must have full database privileges and appropriate environment settings including the SID and path binaries.
  - The DB User must be the same as the user running the application being monitored.

---

**Note:** Multiple registrations may be necessary for a single database agent if different database users are running multiple applications.

---

9. After the database agent has been registered, the JVM Diagnostics Manager will start monitoring the cross-tier JVM calls between applications being monitored for a particular JVM and the underlying database.

## 9.8 Creating a Special Repository User for Loading JVM Diagnostics Heaps

To create a special Enterprise Manager repository user who can load JVM Diagnostics Heaps to the repository, follow these steps:

---

**Note:** This is an optional step and can be used to create less privileged users who can load heaps using the loadheap script.

---

1. Run the `create_jvm_diagnostic_db_user.sh` script from the command prompt
 

```
$ ./create_jvm_diagnostic_db_user.sh
```
2. You will see a message prompting you to continue with the user creation. Enter Y to continue. The following prompts are displayed:

```
Enter Database Host Machine Name or IP address:[localhost]:
```

Enter the host name or the IP address of the machine on which the database has been installed.

Enter Database Port on "localhost":[1521]

Enter the database port number as 25062.

Enter ORACLE\_SID for database on "localhost":[s0222]

Enter the Oracle system identifier for the database.

Enter database admin user name:[SYSTEM]

Enter the name of the database administration user.

Enter "SYSTEM" user password:

Enter the password for the database administrator.

Enter JVM Diagnostics database username:[JVMDIAG]

Enter the user name for the JVM Diagnostics Administrator being created.

Enter "JVMDIAG" user password:

Enter the password for the JVM Diagnostics Administrator. You are then prompted to re-enter the password. After you have re-entered the password, you will see the following:

```

Creating Database user "JVMDIAG" ....
Database user "JVMDIAG" created
Public synonyms would be created for following database objects:
Tables
1) JAM_HEAPREL
2) JAM_HEAPOBJ
3) JAM_HEAPUSAGE
4) JAM_HEAPROOTS
5) JAM_HEAPROOTREL
6) JAM_HEAPOBJSUM
7) JAM_HEAPSNAP
    
```

3. You are prompted to continue with the setup. Press **Enter** to continue. You will see the following:

```

Required privileges granted
JVM Diagnostics Database User "JVMDIAG" created
Now onwards, to load heap, use "JVMDIAG" user
    
```



---

# Installing and Configuring Application Dependency and Performance

This chapter explains the steps to enable Application Dependency and Performance in the Enterprise Manager Grid Control environment. Information includes:

- [Section 10.1, "Quick Steps for Installing and Configuring the ADP Manager"](#)
- [Section 10.2, "Pre-Installation Requirements"](#)
- [Section 10.3, "Installation Process"](#)
- [Section 10.4, "Post-Installation Requirements"](#)

## 10.1 Quick Steps for Installing and Configuring the ADP Manager

The following list summarizes the steps used to install the ADP Agent (Agent). For details of the installation, refer to the chapters in this manual.

Before you install the ADP Agent, ensure you have installed the Enterprise Manager Grid Control software. See the *Oracle Enterprise Manager Grid Control Basic Installation Guide* for details.

1. Create a user for the ADP Manager in a database. Each ADP Manager requires its own user which needs the CONNECT and RESOURCE roles. Use the following SQL:

```
CREATE USER adp1 IDENTIFIED BY adp1;  
GRANT CONNECT, RESOURCE to adp1;
```

2. Collect the following application server information:
  - admin server host
  - admin server port
  - admin user name
  - admin password
  - admin connect protocol (for example, t3 or t3s)
3. Install ADP from Management Service (OMS) home directory. The default location is: <ORACLE\_HOME>/Middleware/oms11g/ocamm
4. Locate the ocamm.zip file. You can either unzip the file on this machine or copy the ocamm.zip file to another machine where you want the ADP Manager to reside.

The ADP Manager should be installed as a non-root user (that is, "oracle") and be the same as the Management Agent install user.

Because of size, Oracle recommends that you install the ADP Manager (2 GB RAM), the OMS (2 GB RAM), and the ADP database (3 GB RAM) on different machines. This recommendation does not apply if the machine that houses the Management Service has enough CPUs (at least 1 per server) and RAM, for both the Management Service and the ADP Manager.

5. Run `Disk1/InstData/<platform>/VM/install.bin`. For Windows, use `install.exe`
6. Select the location to install into when prompted, do not create links. In the following steps, `<ADP_HOME>` is used to refer to the location you selected for the installation.
7. Start the configuration console by running `<ADP_HOME>/bin/config.sh`  
This may bring up an applet login window if it is run on Windows or on UNIX with X-windows. Ignore the UI and proceed to Step 8.
8. To access the Configuration tab from the Grid Control Home page, click **Targets**, then click **Middleware**. On the Middleware page, in the Related Links section, click **Application Dependency and Performance**. Go to the **Registration** tab, select **Manager**.
9. Enter in the IP address for the machine where you installed the ADP manager in step 4. Use 51099 as the default port. Click **Add** and then verify your configuration by clicking **Test Connection**.
10. Click **Configuration** and select the **Database Configuration** node. Edit the existing default database configuration **MyOracle**
11. Enter your database access credentials
12. Click **Test Connection** to verify database connection and then click **OK**
13. Shut down the configuration console by running `<ADP_HOME>/bin/acshut.sh`
14. Wait about 30 seconds after `acshut.sh` exits for the shutdown to complete
15. If you are monitoring a WebLogic server, deploy (from the ADP/OCAMM install directory) `<ADP_HOME>/deploy/HttpDeployer.ear` to your admin server. This step is optional for OC4J.
16. Start the ADP Manager by running `<ADP_HOME>/bin/acsera.sh` (for example, `nohup ./acsera.sh &"`)
17. Return to the ADP UI, click **Configuration** tab, and then select **Resource Configuration** node.
18. Click **Create New Resource**
19. Enter resource name Product and Version. (**NOTE:** For WebLogic versions 10.3.2 and higher, use version "11.1", which corresponds to the Oracle Fusion Middleware version number.) Click **Continue**.
20. Click **Configure** and enter your application server access information For an Oracle WebLogic Server, the fields listed are the following:
  - Protocol: Either `t3` or `t3s`, depending on the protocol used to access JMX MBeans and EJBs on the monitored domain
  - Host: DNS name or IP address of the WebLogic Admin server for the monitored domain. (Note: If a `<listen-address>` is specified for the admin server, the name listed here must match that parameter.

- Port: Port which the WebLogic Admin server is listening for t3 connections.
- Http Host: DNS name or IP address of WebLogic server where the HttpDeployer.ear has been deployed (usually the admin server).
- Http Port: Port where the Weblogic server where the HttpDeployer.ear has been deployed is listening for connections
- Http Protocol: Either http or https depending on the configuration of the WebLogic server where the HttpDeployer.ear has been deployed
- Admin Server User ID: User name to access the WebLogic admin server
- New Password: Password for the username used to WebLogic admin server
- Retype New Password: Repeated password for the username used to access the WebLogic admin server
- BPEL JNDI Factory: Only used for SOA Suite 10g (Not used for 11g). Ignored in all other cases. For SOA Suite 10g, the BPEL JNDI factory used to access the BPEL API. Typically, leave at the default value.
- BPEL JNDI URL: Only used for SOA Suite 10g (NOT used for 11g). Ignored in all other cases.
- JNDI URL used to access the SOA Suite 10g BPEL API. Typically, follow the pattern of the default value and change the host and port to match.
- BPEL Console User Name: Only used for SOA Suite 10g (not used for 11g). Ignored in all other cases. Username used to access the BPEL console.
- BPEL Console Password: Only used for SOA Suite 10g (not used for 11g). Ignored in all other cases. Password associated with the username used to access the BPEL console.
- ESB protocol: Only used for SOA Suite 10g (NOT used for 11g). Ignored in all other cases. URL used to access the SOA Suite 10g ESB API. Either http or https depending on the configuration of the ESB.
- ESB Host: Only used for SOA Suite 10g (NOT used for 11g). Ignored in all other cases. DNS name or IP address used to access the SOA Suite 10g ESB API.
- ESB Port: Only used for SOA Suite 10g (NOT used for 11g). Ignored in all other cases. Port used to access the SOA Suite 10g ESB API.
- ESB Console User Name: Only used for SOA Suite 10g (NOT used for 11g). Ignored in all other cases. User name used to access the ESB console.
- ESB Console Password: Only used for SOA Suite 10g (NOT used for 11g). Ignored in all other cases. Password associated with the user used to access the ESB console
- Agent Keystore Password: DO NOT change this value unless you have installed a custom agent keystore into the ADP Manager.
- Agent Truststore Password: DO NOT change this value unless you have installed a custom agent truststore into the ADP Manager.
- Provider Home: Typically, this field is left blank and the ADP Manager will auto-detect the provider home. If there are deployment problems, this field may be used to manually override the root path where WebLogic client jars are downloaded from the WebLogic admin server. If such a manual override is necessary, enter in an absolute path to the WebLogic home (WL\_HOME) on the admin server specified.

21. Click **OK** to return to the previous screen
22. Click **Test Connection**. If you are monitoring a WebLogic server, wait for the `<ADP_HOME>/lib/bea/<version>_runtime` directory to be created and populated with jar files. This could take a few minutes. If nothing happens, check `log/manager-log#.csv` for exceptions.
23. Click **Back** to return to the previous screen and then click **Deploy**
24. On the subsequent screen, select **Deploy** from the Command menu and then select the servers/cluster you want to deploy the Agent to. Ensure that the admin server is always selected.  
  
Note: In the case of WebLogic Portal, AdminServer is always selected. Please unselect / select Nodemanager in case of weblogic based on how do configure the Weblogic manager to start.
25. Click **Continue**
26. Wait for deployment to complete. Verify that the UI reports **Deployment Successful**; if not, check for exceptions in `<ADP_HOME>log/manager-log#.csv`  
  
To view the status on the same page, click **Refresh**. Note that you might have to click **Refresh** a few times as it may take a while to deploy and update the status. Depending on the number of managed servers in your domain, it could take several minutes (up to 60 seconds per managed server) for this operation to complete. Stay on this page and monitor the manager log until some status is returned.
27. Click **Back** to return to the previous screen and then click **Save**
28. Close the browser-based console
29. Shut down the ADP Manager by running `<ADP_HOME>/bin/acshut.sh`
30. Wait about 30 seconds after `acshut.sh` exits for the shutdown to complete
31. Restart the application servers you deployed the Agent to. The ADP installer adds jar files to the system classpath and adds parameters to the Java options when deployed. An application server restart is required for those to take effect.
32. Start the ADP Manager by running `<ADP_HOME>/bin/acsera.sh` (for example, `nohup ./acsera.sh &"`)
33. Wait for discovery to occur. You can check that discovery is happening by looking to see if `<ADP_HOME>/darchive/server_archive` is created and growing, and then that `<ADP_HOME>/darchive/app_archive` is created and growing. If you think something is going wrong, you can check `log/manager-log#.csv` for exceptions, but this process can take several minutes.
34. To return to the ADP pages from the Grid Control Home page, click **Targets**, then click **Middleware**. On the Middleware page, in the Related Links section, click **Application Dependency and Performance**. Go to the **Monitoring** tab.
35. Verify that all servers to which you deployed the ADP Agent are up and running. Click the CAMM node in the Monitoring tab. The Agent information is available in the Main Display window.  
  
For metrics to be reported, the agent status must be in the REPORTING state. (If Agent Installed is true, the Agent Status will start in the DEPLOYED state, then go to CONNECTED, and then become REPORTING.) Check the manager and agent logs for exceptions if this is not the case.



## 10.2 Pre-Installation Requirements

Prior to installing ADP, install the Oracle Management Service.

To save yourself time during the installation and configuration processes, check compliance with the following dependencies. The following list uses examples of environment dependencies for WebLogic 8.1, 9.2, 10.0, and 10.3.2.

1. Check Support Hardware
  - CPU - At least 1 CPU (2 Ghz Intel-equivalent or better) for the ADP Manager
  - RAM - At least 2GB RAM for the ADP Manager
  - Free HD Space - At least 20 GB free disk for the ADP Manager
2. Check Supported OS
  - OS
  - Patch Level
  - JRE - Clients must have the Java 1.6 Plugin installed
3. Check Network
  - No Firewalls (same subnet)
  - If Firewalls - ports are open The CAMM manager must be able to communicate to the monitored application servers on their t3 (or t3s if enabled) ports directly. The monitored application servers must be able to communicate with the CAMM manager on ports 51099 and 55003 directly.
  - No multi-homing interface conflicts
  - IP interfaces are static
4. Check browsers

The following browsers are supported:

  - Mozilla Firefox 3.0 and higher
  - Internet Explorer 7 and higher
  - Safari 4.0 and higher
5. Check Web Logic configuration
  - WebLogic 8.1, 9.x, 10.x
  - One cluster per WLI domain
  - Applications deployed and running
  - There is process load
  - t3 channels are enabled
6. Check Database
  - Check availability of Oracle database - Ensure that the schema has been created for ADP data and that connectivity is available from the ADP Manager.

### 10.2.1 Environment Considerations

Consider the following environment considerations:

- [Section 10.2.1.1, "Access and Connectivity"](#)
- [Section 10.2.1.2, "ADP Database"](#)

### 10.2.1.1 Access and Connectivity

Determine whether you have access and connections to the monitored application platform, network, and DHCP.

**10.2.1.1.1 Access to Monitored Application Platform** With respect to the target application platform to be monitored, ADP requires the following level of access:

- System Access to the Oracle SOA Suite/Oracle WebLogic/IBM WebSphere Application domain environment (such as ability to deploy to the domain and restart the domain)
- System Administration capabilities to modify WebLogic/WebSphere startup files.

Note that ADP accesses the target application platform as application users created with administrative privileges. Creating an operating system user (such as a UNIX user) is not necessary other than securing the ADP installation and configurations.

The Oracle SOA Suite/Oracle WebLogic/IBM WebSphere Administration domain server and the Managed Servers must be accessible from the machine where ADP is being installed.

**10.2.1.1.2 Network Connectivity** Firewalls provide security for networks and applications by preventing certain connectivity between machines, servers, and applications. ADP is remotely monitoring performance and availability of the Oracle SOA Suite/Oracle WebLogic/IBM WebSphere runtime environment. If firewalls or proxies are present in the network topology, they must be configured to allow communication traffic between ADP and the Oracle SOA Suite/Oracle WebLogic/IBM WebSphere runtime environment.

Oracle recommends that the ADP dedicated machine be deployed into the same subnet, within the firewall perimeter.

If a firewall is unavoidable, ensure that the ADP port is open and that t3 and RMI traffic is allowed. The ADP Manager communicates with the monitored application server with JMX and EJB/T3 calls. The ADP Agent communicates back to the ADP Manager using Sun RMI.

If you are using ipv6 addressing:

1. Install the application domain environment and ADP in the ipv6 machine, and
2. Comment out the following line in the config.sh and acsera.sh files:

```
java.net.preferIPv4Stack=true
```

**10.2.1.1.3 DHCP** Oracle recommends that the ADP dedicated machine, the Java EE Application Admin Server, and Managed Servers are assigned static IP addresses. Non-static configurations are more difficult to configure.

### 10.2.1.2 ADP Database

Ensure that the ADP database is properly installed and configured.

ADP can store application metadata, performance and availability metrics in either a dedicated database or a schema located in the Management Repository. The database and schema must be accessible by ADP. Currently, ADP supports Oracle 10g, Oracle 11g, and MySQL database platforms for the repository.

For an Oracle or MySQL database installation, a dedicated user database and schema should be created with adequate access rights and connectivity restrictions. Setup and configuration for the MySQL database are covered in [Section 10.3.5.2, "Configuring MySQL DBMS for ADP"](#).

## 10.2.2 Clustering and Application Domains

Application domain is a logical/administrative context for a collection of resources, such as a WebLogic cluster (or WebSphere cell) and/or standalone server instances. A single ADP Manager instance can also monitor mixed application domains of different vendor platforms. Thus, with a single Manager instance, a human operator can have a single, consistent view into a large, heterogeneous environment where, for example, WebLogic domains and WebSphere cells co-exist.

Because of this flexibility, a single instance of ADP is best dedicated to a single *environment* - as in production, or QA. Within each environment, application server platforms may be heterogeneous, based on different vendors (WebLogic/WebSphere) or different versions (WLS 9.2 versus 10.0), and have diverse deployment configurations (such as standalone servers, Oracle SOA Suite, WLS clusters and/or WebSphere cells).

Mixing environments within a single ADP instance (such as domains from production and from QA) is technically feasible but not recommended, since traffic patterns from the different environments tend to be different (live versus load testing) and can complicate advanced data analyses. In this case, deploying multiple instances of ADP would be the correct solution, where a dedicated ADP instance monitors each specific environment (Production, Staging, QA, and so on). You can find details about multiple ADP instances support in [Section 10.3.4, "Deploying ADP Components"](#).

### 10.2.2.1 WebLogic Clustering

WebLogic clustering includes clusters and the Node Manager.

**10.2.2.1.1 Clusters** ADP supports the monitoring of performance and availability across any number of servers, clusters, and machine configurations across multiple WebLogic domains. It will automatically detect application clustered deployments and dependencies and build metrics that relate to this deployment architecture. There are no limitations for the number of domains, clusters, machines, and servers a single ADP instance can monitor.

**10.2.2.1.2 Node Manager** Node Manager is used by the WebLogic Admin server to remotely start and stop WebLogic Managed Server instances. The ADP Agent Deployment makes some changes to startup parameters of a Managed Server. When using Node Manager, the ADP Agent Deployment detects that Node Manager is running and make those changes to the Node Manager server JVM startup parameters.

## 10.2.3 Statistics and Privileges To Obtain From Various Administrators

To save time during the installation and configuration of the ADP Manager and ADP Agent, get the privileges and field information you need *before* you start the process.

### 10.2.3.1 Database Administrator

Contact the Database Administrator (DBA) for the information needed to fill in the fields in the Resource section of the installation. You need the following values for the Oracle database instance:

- Database SID (service identifier)
- Database host and port
- Database user with CONNECT and RESOURCE roles

### 10.2.3.2 WebLogic Administrator

Contact the WebLogic Administrator (WebLogic Admin) for the following values:

- Admin Console host and port
- WebLogic administrator username and password
- If using the Oracle SOA Suite: BPEL and ESB console username, password, host, and port

### 10.2.3.3 WebSphere Administrator

Contact the WebSphere Administrator (WebSphere Admin) for the following values:

- Network Deployment Manager (dmgr) SOAP port
- WebSphere administrator username and password
- Client certificates (if global security is enabled)

### 10.2.3.4 OC4J Administrator

Contact the OC4J Administrator (OC4J Admin) for the following values:

- Application Server Control host and port
- OC4J administrator username and password
- OPMN request port
- If using the Oracle SOA Suite: BPEL and ESB console username, password, host, and port

## 10.3 Installation Process

This section describes the ADP installation process in detail. For generic installation information, refer to [Section 10.1, "Quick Steps for Installing and Configuring the ADP Manager"](#).

Topics include:

- [Section 10.3.1, "Installing ADP - Windows Installation"](#)
- [Section 10.3.2, "Installing ADP - Solaris, Linux, and AIX Installations"](#)
- [Section 10.3.3, "Setting Up ADP Data Repository"](#)
- [Section 10.3.4, "Deploying ADP Components"](#)
- [Section 10.3.5, "Installing and Configuring ADP Data Repository"](#)
- [Section 10.3.6, "Backing Up the ADP Database"](#)

### 10.3.1 Installing ADP - Windows Installation

Perform the following steps to install ADP on Windows. Note that at any time during the installation, you can click **Cancel** to cancel the installation, or click **Previous** to go back a screen.

1. Install ADP from the OMS home directory. The default location is: <ORACLE\_HOME>/Middleware/oms11g/ocamm  
The installation automatically starts. Choose the language in which you want the installation instructions.  
The Introduction screen appears.
2. Read the introduction text and click **Next**.  
The Choose Install Set screen appears.
3. The Install Folder screen appears.
4. Click **Next**. On the Choose Install Folder screen, indicate which folder will contain the installation. Either type the folder's location or accept the default C:\oracle\em11g folder.
5. Click **Next**. On the Choose Shortcut Folder screen, choose a Shortcut Folder in which to copy the ADP product icons. If desired, select the **Create Icons for All Users** box.
6. Click **Next**. The Pre-Installation Summary screen lists the options to be installed. If an option is not correct, click **Previous** to move back and correct the option. When the summary is correct, click **Next**.

### 10.3.2 Installing ADP - Solaris, Linux, and AIX Installations

Perform the following steps to install ADP on your UNIX environment. Note that at any time during the installation, you can type **back** to go back, and can cancel the installation by typing **quit**.

1. Install ADP from the OMS home directory. The default location is: <ORACLE\_HOME>/Middleware/oms11g/ocamm.
2. In the ocamm directory, unzip the camm.zip file and execute the install.bin program:

```
unzip camm.zip
./install.bin
```

**Note:** Ensure that *install.bin* has execution privileges and, if necessary, use *chmod a+x install.bin* command to make it executable.

The Introduction screen appears. Choose the language in which you want the installation instructions.

3. Read the introduction text and press the <ENTER> key to continue.
4. Click **Next**. From the **Install Folder** text, indicate which folder will contain the installation. Either type the folder's location or accept the default \$HOME/oracle/em11g.

```
Choose Install Folder
-----
```

```
Where would you like to install?
```

```
Default Install Folder: /root/oracle/em11g
```

```
ENTER AN ABSOLUTE PATH, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
```

```
:
```

Choose the directory in which to install ADP and press the <ENTER> key. You will be prompted of where to install the ADP product icons.

5. Click **Next**. Indicate which directory will contain the Link locations.

```
Choose Link Location
```

```
-----
```

```
Where would you like to create links?
```

```
->1- Default: /root
  2- In your home folder
  3- Choose another location...

  4- Don't create links
```

```
ENTER THE NUMBER OF AN OPTION ABOVE, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
:
```

Choose the desired directory and press the <ENTER> key. You will be shown a summary of your installation choices

Choose option 4 - **Don't create links**. Run startup scripts out of the bin directory as the install location.

6. Verify that your installation choices are correct. If they are not, type **back** to go back and make choices. If you are satisfied, press the <ENTER> key to begin the installation.
7. Press <ENTER> to finish after the installation completes.

### 10.3.3 Setting Up ADP Data Repository

ADP requires an RDBMS be setup as the data repository for runtime metrics collection. Oracle 10g, Oracle 11g, and MySQL RDBMS are supported for this purpose. Details for manual installation and configuration of the Data Repository are described in [Section 10.3.5, "Installing and Configuring ADP Data Repository"](#).

### 10.3.4 Deploying ADP Components

Deploying ADP components involves the following steps:

- Configuring ADP with information about the target application platform. For each monitoring environment, a single instance of ADP can monitor multiple application servers or clusters.
- Deploying the ADP Agent components to the target application server instances or clusters (for example, managed servers in a cluster of a domain, and so on).

To deploy ADP components, refer to [Section 10.3.4.1, "Configuring ADP"](#).

#### 10.3.4.1 Configuring ADP

In ADP, a monitored target application platform, whether it is an individual application server instance or a cluster in a management domain, is called a *Resource*.

Oracle highly recommends that you register and update the configuration of your server resources through the ADP Manager.

1. Start the ADP Manager.

- Windows: From the Start menu, select **Oracle**, then select **ADP**.
  - UNIX: In the /bin directory, invoke the following command:  

```
nohup ./config.sh &
```
2. On the Grid Control Home page, click **Targets**, then click **Middleware**. On the Middleware page, select **Application Dependency and Performance** in the Related Links section.
  3. Select the **Configuration** tab and select **Resource Configuration**.
  4. Click the **Create New Resource** button in the main pane.
  5. Name your resource and select your application server product and version.
  6. Click **Continue**.
  7. Click **Configure** in the middle of the main pane.
  8. Type in your application server details. Ignore the BPEL and ESB related options if you do not have the Oracle SOA Suite.

---



---

**Important:** DO NOT modify the fields for Agent Keystore Password and Agent Truststore Password.

---



---

9. Click **OK**.
10. Click **Save**. (If you omit this step, your configuration will be lost.)

#### 10.3.4.2 Deploying ADP Agents on the WebLogic Platform

The following sections describe how to deploy ADP Agents on the WebLogic platform.

**10.3.4.2.1 HttpDeployer.ear Procedure** Use the WebLogic Admin console to deploy the em11g/deploy/HttpDeployer.ear application.

**10.3.4.2.2 Deploying the ADP Java Agent** Using the GUI-based deployer as documented in the [Section 10.1, "Quick Steps for Installing and Configuring the ADP Manager"](#) is recommended.

#### 10.3.4.3 Deploying ADP Agents on the WebSphere Platform

Deployment of ADP Agent for WebSphere platform is a two-phase process. First you need to install ADP IBM Deployer application responsible for ADP Agent libraries deployment initial handshake between ADP and ADP Agent, then deploy the Agent libraries on the target system. There are two options to achieve this:

- Automatic deployment using ADP websphereDeployer script
- Manually installing all the supporting artifacts

**10.3.4.3.1 Required IBM WebSphere Libraries** Once WebSphere is registered via the ADP Administration UI, the libraries required by ADP to connect to WebSphere must be defined in the actual classpath. This is done through the Resource Configuration UI.

Note that an installation must be available directly on the machine or via a NFS/SMB mount. The following property would need to be modified to mirror the absolute path to the WebSphere Application Server (WAS) home directory. The necessary libraries will then be loaded on the classpath accordingly.

*Example of wsHome setting:*

```
<ns1:configParameter>
  <ns1:key>wsHome</ns1:key>
  <ns1:value>C:/Progra~1/IBM/WebSphere/AppServer</ns1:value>
</ns1:configParameter>
```

**10.3.4.3.2 Automatic ADP Agent Deployment for WebSphere Platform** The following sections explain how to automatically deploy the ADP Agent for the WebSphere platform.

#### Deploying WebSphere File Transfer Application

If you are using a network deployment manager (dmgr), you should skip this step. Otherwise (if running a standalone WAS server), ensure the WebSphere File Transfer Application is installed and running. If not, deploy filetransfer.ear to the WebSphere application server. This is required to enable the ADP Agent automatic deployment process. If the ADP Agent is being installed manually, the file transfer application is not required.

#### Deploying ADP Java Agent for WebSphere Platform

Using the GUI-based deployer as documented in the [Chapter 10.1, "Quick Steps for Installing and Configuring the ADP Manager"](#) is recommended.

**10.3.4.3.3 Deploying ADP Java Agent for Oracle Platform** Use the GUI-based deployer as documented in [Section 10.1, "Quick Steps for Installing and Configuring the ADP Manager"](#).

#### 10.3.4.4 Running Multiple ADP Instances

There will always be a *default* instance of the ADP (the installed instance). For each additional Manager instance needed, first create an instance directory underneath ACSERA\_HOME. (ACSERA\_HOME is the directory that ADP was installed into, for example, C:\oracle\em11g.) Copy the installed config, mcccconfig, and schema directories under the new instance directory.

In INSTANCE\_DIR/config/Acsera.properties, set the following properties to unique, instance-specific values:

- RMI.Registry.Port = other than the original one (51099 by default)
- RMI.RemoteServiceController.ServerPort= other than the original one (55000 by default)
- RMI.JavaProvider.ServerPort= set the port to be more than 100 counts greater than the original new instance. For example, if the original port is 55003, increase the RMI.JavaProvider.ServerPort to 55103.

The default value for RMI.Registry.Port is 51099. It is recommended that additional Manager instances use values counting down from the default (51098, 51097, and so on.)

Update dbconfig.xml for the target database (use a different database schema for the instance).

If there is a JDK installed under the ACSERA\_HOME directory, then the scripts will use that JDK. Otherwise, JAVA\_HOME must be set to the directory of the installed JDK.

The following commands may be used to control non-default Manager instances:



Windows	Linux/AIX/Solaris/HP-UX
bin/acsera.bat	bin/acsera.sh
bin/acshut.bat	bin/acshut.sh
bin/deployer.bat	bin/deployer.sh
bin/standalone.bat	bin/standalone.sh

The acsera.\*, acshut.\* and standalone.\* scripts take an instance name as the first argument. Leaving out the instance name causes the script to target the default instance.

**10.3.4.4.1 Front-End Configuration** The configure new ports, perform the following steps:

1. From the Grid Control Home page, click **Targets**, then click **Middleware**. On the Middleware page, select **Application Dependency and Performance** in the Related Links section.
2. On the ADP page, click the **Registration** tab and register a new manager using a new port.

## 10.3.5 Installing and Configuring ADP Data Repository

ADP maintains a database (can be Oracle or MySQL) of information collected on the system it is monitoring. This database can be contained on the same machine on which ADP is running. It can also be hosted on a remote machine. The following paragraphs describe how to configure the ADP Data Repository.

The following sections describe procedures that may be performed by the ADP administrator to customize the database configuration.

### 10.3.5.1 Configuring Oracle DBMS for ADP

ADP currently supports the Oracle 11g database for its runtime repository. In order to set this up, the following steps are necessary in order to setup and configure the Oracle DBMS for ADP. ADP will initialize the database upon connecting to it and generate the required tables.

1. Install Oracle DB: 11g on a separate machine
2. Create a new user in database (preferably "ADP" or something distinct)
3. Set the System Global Area to have at least 1 GB (1275068416 bytes)
4. Increase the number of processes in database from 150 to 300 (optional; this is primarily for an OracleXE database).

Execute the following commands using Oracle SQL\*Plus:

- a. connect / as sysdba; (this will connect to the database as DBA)
- b. show parameter processes;

NAME	TYPE	VALUE
aq_tm_processes	integer	0
db_writer_processes	integer	1
gcs_server_processes	integer	0

NAME	TYPE	VALUE
job_queue_processes	integer	10
log_archive_max_processes	integer	2
processes	integer	150

- c. alter system set processes=300 scope=spfile; (Run this to increase the number of processes. You can set it to a higher number if you want.)
- d. shutdown immediate (for shutdown the server)
- e. startup (startup the server)
- f. show parameter processes; (run this to verify the changes took effect)

Use of the GUI-based configuration tool described in the [Section 10.1, "Quick Steps for Installing and Configuring the ADP Manager"](#) at the beginning of this manual is recommended. For manual database configuration, fields in dbconfig.xml may be modified.

### 10.3.5.2 Configuring MySQL DBMS for ADP

ADP currently supports MySQL 4.1 database and higher for its runtime repository. The following steps are necessary to setup and configure the Oracle DBMS for ADP. ADP initializes the database upon connecting to it and generates the required tables.

1. Install MySQL 4.1 or higher on a separate machine
2. Create a new user in the database (preferably "Oracle" or something distinct) and grant the user the appropriate privileges
3. Tune memory for performance as indicated below:

On a 1 GB server shared by ADP and MySQL, the following parameter should be set in the my.ini file to increase database subsystem performance:

```
set-variable=key_buffer=128M
```

On a 2 GB server shared by ADP and MySQL or on a dedicated to MySQL 1GB server, the parameter should be set as:

```
set-variable=key_buffer=256M
```

### 10.3.6 Backing Up the ADP Database

Oracle recommends that regular backups be made of the database utilized as a runtime repository for ADP. The Oracle database provides the EXP and IMP tools that allow you to quickly backup and restore the database if necessary. The most convenient method for backing up the MySQL database is using the mysqldump command. By default, this is found in c:\mysql\bin on Windows, or /mysql/bin on UNIX.

## 10.4 Post-Installation Requirements

This section explains the following post-installation requirements:

- [Section 10.4.1, "IBM WebSphere Post-Deployment Requirements"](#)
- [Section 10.4.2, "Configuring Oracle SOA Suite for Secure Connectivity"](#)

- [Section 10.4.3, "Configuring Oracle WebLogic Server or Oracle WebLogic Portal \(WLP\) for Secure Connectivity"](#)
- [Section 10.4.4, "Importing a Certificate into the Manager's Keystore"](#)
- [Section 10.4.5, "Configuring the ADP Agent When WebLogic Is Installed As a Windows Service"](#)

---

**Note:** You must register the CAMM Manager with the Oracle Management Service. Otherwise, the ADP UI will not be available.

---

## 10.4.1 IBM WebSphere Post-Deployment Requirements

The following post-deployment requirements are specific to ADP deployments on IBM WebSphere Application Server.

### 10.4.1.1 Configuring ADP for WebSphere Application Server 6.1 Secured Connections

The main goal is to add the signer certificate of each administration server to ADP's truststore, which is needed by each resource to connect to the server. This allows ADP to trust the server when making secured (SSL) connections to the server. Without this trust, the SSL handshake will fail.

When using the default ADP truststore, the server's signer certificate would be added to AcseraManagerTrust.jks. This procedure assumes that the customer is using the default key.p12 and trust.p12 keystores for their security support. If a different trust store is being used, refer to that trust store instead.

1. Exporting the administration server's signer certificate for resource.

If the administration server is the deployment manager in a WebSphere Application Server ND, export signer certificate from trust.p12 of the deployment manager located at path

```
<WAS_HOME>\profiles\Dmgr01\config\cells\<CellName>\trust.p12
```

If the administration server is a standalone server, export the signer certificate from trust.p12 of the standalone server located on the following path:

```
<WAS_HOME>\profiles\AppSrv01\config\cells\<CellName>\nodes\<NodeName>\trust.p12
```

To export, run the following command:

```
JAVA_HOME/bin/keytool -export -keystore <trust path>
-storepass WebAS -storetype PKCS12 -alias default_signer
-file servercert
```

**Note:** When exporting a PKCS12 store type, run keytool from an IBM JDK since it has support for this format type.

2. Import the administration server's signer certificate into the ADP truststore.

Import the exported certification according to the information in [Section 10.4.4, "Importing a Certificate into the Manager's Keystore"](#).

### 10.4.1.2 Configuring WebSphere 5.1 for Secure Connectivity

To run ADP against WebSphere with enabled Global Security, perform the following steps:

1. Identify the `com.ibm.ssl.keyStore` and `com.ibm.ssl.trustStore` files in `soap.client.props` and `sas.client.props` in `[WAS_HOME]/properties` as follows:
  - a. Copy the indicated keystore and truststore files to the ADP Manager.
  - b. Import the files following the instructions in [Section 10.4.4, "Importing a Certificate into the Manager's Keystore"](#).

**Example 10–1 soap.client.props**

```
com.ibm.SOAP.securityEnabled=true
com.ibm.SOAP.loginUserId=admin
com.ibm.SOAP.loginPassword=test
```

**com.ibm.ssl.keyStore=**

```
com.ibm.ssl.keyStorePassword=acserajava
```

**com.ibm.ssl.trustStore=**

```
com.ibm.ssl.trustStorePassword=acserajava
```

**Example 10–2 sas.client.props**

```
com.ibm.CORBA.securityEnabled=true
```

```
com.ibm.ssl.keyStoreType=JKS
```

**com.ibm.ssl.keyStore=**

```
com.ibm.ssl.keyStorePassword=acserajava
```

```
com.ibm.ssl.trustStoreType=JKS
```

**com.ibm.ssl.trustStore=**

```
com.ibm.ssl.trustStorePassword=acserajava
```

2. If you encounter security exceptions in the ADP EJB when the application server starts, you may need to update the `[WAS_HOME]/properties/server.policy` file and append the configuration that follows.

**Example 10–3 server.policy**

```
// Allow the Acsera Agent all permissions
grant codeBase "file:${was.install.root}/AcseraAgent/lib/-" {
permission java.security.AllPermission;
};

// Allow the Acsera Deployer EJBs all permissions
grant codeBase "file:${was.install.root}/installedApps/[node]/[Acsera app
name].ear/-" {
permission java.security.AllPermission;
};
```

Normally, using the `websphereDeployer` command, the ADP deployer EJBs would be deployed in the WebSphere server environment with the application name of the form:

```
Acsera_<node name>_<server name>
```

For example, this is an application name of a deployer deployed on node `a6-7` and server `WebSphere_Portal`.

```
Acsera_a6-7_WebSphere_Portal
```

## 10.4.2 Configuring Oracle SOA Suite for Secure Connectivity

The Oracle SOA Suite may be configured to support RMIS (RMI over SSL) connectivity. In this case, ADP can be configured to use this secure connection. To configure ADP to do this, perform the following steps:

1. On the Oracle SOA Suite install, look at ORACLE\_HOME/j2ee/<instance>/config/rmi.xml, locate the <ssl-config> element, and identify the path in the keystore attribute.
2. Copy the keystore file indicated to ADP manager's config directory (for example, em10/config)
3. Import this keystore file following the instructions in [Section 10.4.4, "Importing a Certificate into the Manager's Keystore"](#).

## 10.4.3 Configuring Oracle WebLogic Server or Oracle WebLogic Portal (WLP) for Secure Connectivity

To configure Oracle WebLogic Server 10.0 to handle connectivity using t3s, the location of the keystore files needs to be updated through the console.

1. Log in to the WebLogic Server console and select the servers under the Environment Servers list that is displayed which you plan to manage with ADP.
2. Select a server from the server list.
3. Select the keystores tab click **Load & Edit** to update the Keystore
4. Make the following changes. Identify the keystore and truststore file paths from the following properties:

### Identity

Custom Identity Keystore

### Trust

Custom Trust Keystore: location of the trust file

5. Repeat steps 2 through 4 for additional server instances that will be managed.
6. Copy the identified keystore and truststore files to the ADP manager.
7. Copy the BEA\_HOME/license.bea to the ADP manager's config directory (for example, em11g/config)
8. Import the keystore and truststore files following the instructions in [Section 10.4.4, "Importing a Certificate into the Manager's Keystore"](#)
9. Locate the following properties in the Acsera.properties file and set them as follows:

```
weblogic.security.TrustKeyStore=CustomTrust
weblogic.security.CustomTrustKeyStoreFileName=AcseraManagerTrust.jks
weblogic.security.CustomTrustKeyStorePassPhrase=acseramanager
```

## 10.4.4 Importing a Certificate into the Manager's Keystore

To import entries from a keystore or truststore, perform the following steps, replacing *ServerStoreFile.jks* with the keystore or truststore from your application server. (Skip steps 1 and 2 if you are importing certificate files from WAS 6.1 as described in [Section 10.4.1.1, "Configuring ADP for WebSphere Application Server 6.1 Secured"](#))

**Connections".) You will generally need to complete these steps twice, once for the keystore and once for the truststore.**

**1. List the key aliases in the keystore/trustfile file from the server**

```
keytool -list -keystore ServerStoreFile.jks -storepass
DemoIdentityKeyStorePassPhrase
```

**Output:**

```
Keystore type: jks
Keystore provider: SUN
```

Your keystore contains 1 entry:

```
demoidentity, Wed Nov 19 13:34:56 PST 2008, keyEntry, Certificate fingerprint
(MD5): 36:06:C2:44:31:0A:28:FC:06:19:F7:AB:C0:7D:27:6A
```

**2. Export a key entry to an intermediate file**

```
keytool -export -alias demoidentity -keystore ServerStoreFile.jks -storepass
DemoIdentityKeyStorePassPhrase -file demo103
```

**Output:**

Certificate stored in file <demo103>

**3. Import the key into the ADP store file (either AcseraManagerKey.jks or AcseraManagerTrust.jks in the ADP manager's config directory)**

```
keytool -import -alias demoidentity1 -keystore AcseraManagerKey.jks
-storepass acseramanager -file demo103
```

**Output:**

```
Owner: CN=b91, OU=FOR TESTING ONLY, O=MyOrganization, L=MyTown, ST=MyState,
C=US
Issuer: CN=CertGenCAB, OU=FOR TESTING ONLY, O=MyOrganization, L=MyTown,
ST=MyState, C=US
Serial number: 510fb3d4b2872e3a093d436fcbe9b24b
Valid from: Tue Nov 18 13:34:47 PST 2008 until: Sun Nov 19 13:34:47 PST 2023
Certificate fingerprints:
    MD5: 36:06:C2:44:31:0A:28:FC:06:19:F7:AB:C0:7D:27:6A
    SHA1: BB:85:6D:4C:0B:4A:92:63:CA:5E:E9:A8:54:42:80:2D:0D:BE:7C:91
Trust this certificate? [no]: yes
Certificate was added to keystore
```

**4. Verify that the key was imported successfully**

```
keytool -list -keystore AcseraManagerKey.jks -storepass acseramanager
```

**Output:**

```
Keystore type: jks
Keystore provider: SUN
```

Your keystore contains 3 entries:

```
demoidentity1, Wed Apr 01 13:03:21 PST 2009, trustedCertEntry,
Certificate fingerprint (MD5): 36:06:C2:44:31:0A:28:FC:06:19:F7:AB:C0:7D:27:6A
demoidentity, Fri Mar 13 15:15:06 PST 2009, trustedCertEntry,
Certificate fingerprint (MD5): 0B:11:02:B5:44:0D:2A:CC:7F:C5:30:5C:1A:C9:A1:6C
mykey, Thu May 19 16:57:36 PDT 2005, keyEntry,
```

Certificate fingerprint (MD5): 5D:B0:EC:28:14:33:26:1F:44:F5:BE:DD:A8:50:15:9D

**5.** Repeat steps 2 through 4 for each key entry listed in step 1.

At present with ADP running with a bundled Sun HotSpot JDK, it is not possible for ADP to configure with PKCS12 type key/trust stores for secured connections. IBM JDK has built-in enhancements that allow it to work with PKCS12 key/trust stores, such as WebSphere 6.1's default key.p12 and trust.p12 stores. Also, there is a WebSphere 6.1 automatic function that is enabled with the property `com.ibm.ssl.enableSignerExchangePrompt=true` that allows a client connecting to a secure WebSphere port that allows automatic download of server's signer certificate and update of client's truststore. However, this automatic function is only available when ADP is running with an IBM JDK which is not the case at present. This is the reason why we need to follow the above procedure to connect with a secured WebSphere 6.1.

### 10.4.5 Configuring the ADP Agent When WebLogic Is Installed As a Windows Service

When the monitored WebLogic server is installed as a Windows Service, the automatic startup changes to deploy the ADP Agent need to be manually applied to the registry entries that control WebLogic startup.

The parameters which need to be changed are in the Windows registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\Current ControlSet\Services\${ServiceName}\Parameters
```

Users should then consult the file on the ADP Manager:

```
deploy/agent/bea/bin/agentoptions.bat (for WebLogic 8.1.x)
deploy/agent/bea9/bin/agentoptions.bat (for WebLogic 9.x and higher)
```

Inspect this file and resolve the net results of its execution as Parameters in the registry.

Note that the `beaaj.jar` named on the `%EXT_POST_CLASSPATH%` variable needs to be placed on the server's classpath.





## Configuring Request Monitoring

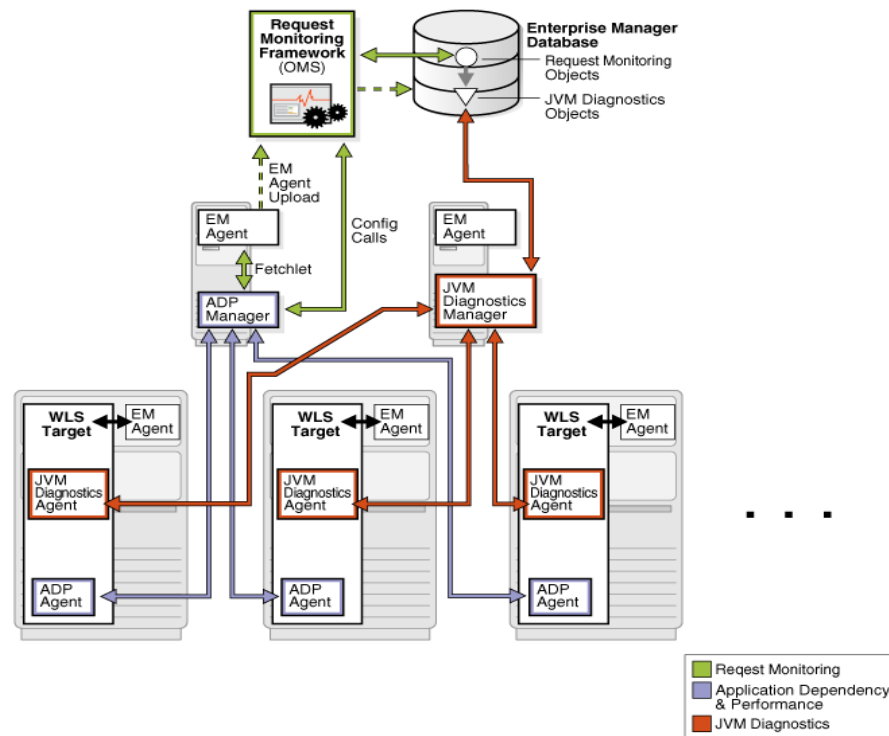
The Request Monitoring feature provides end-to-end visibility into requests for Java applications deployed to WebLogic Server. It establishes a clear view of a request flowing through a multi-tier java environment. This chapter describes the procedure to setup and enable the Request Monitoring feature in Enterprise Manager Grid Control. This chapter covers the following:

- [Section 11.1, "Request Monitoring Architecture"](#)
- [Section 11.2, "Prerequisites"](#)
- [Section 11.3, "Configuring Request Monitoring"](#)

### 11.1 Request Monitoring Architecture

The following diagram shows the request monitoring architecture.

**Figure 11–1 Request Monitoring Architecture**



The JVM Diagnostics Agent and the ADP Agent are deployed on the WebLogic Server targets that are being monitored. These agents collect JVM Diagnostics and Application metrics and periodically report them to the JVM Diagnostics and ADP Managers. The Management Agent (EM Agent) on the ADP Manager machine, periodically runs a Fetchlet to retrieve the metrics from the ADP Manager and uploads them to the Management Repository. The Request Monitoring framework processes these metrics and uses them to monitor and analyze requests.

## 11.2 Prerequisites

Before you enable Request Monitoring, you must do the following:

- Identify the WebLogic Server targets and J2EE applications that are to be monitored.
- Ensure that the necessary hardware for the ADP Manager, ADP database, and JVM Diagnostics Manager is available.

## 11.3 Configuring Request Monitoring

To configure and enable request monitoring, follow these steps:

1. Deploy the Oracle Management Service and the Management Repository database. For details, see the *Enterprise Manager Basic Installation and Configuration Guide*.
2. Install the ADP Manager, and deploy the ADP Agent on the WebLogic Server targets. For details on installing the ADP Manager, see the [Chapter 10, "Installing and Configuring Application Dependency and Performance"](#). The ADP Manager must be installed as the same user as the Management Agent (as described in the step below).
3. Deploy the Management Agent on all the machines on which the ADP Managers have been installed.
4. Deploy the Management Agent on the machine on which the WebLogic server target to be monitored is running. In Enterprise Manager Grid Control, click **Targets** and **Middleware**. On the Middleware page, check if the WebLogic Server target to be monitored has been discovered and registered as a target. If it has not been registered, add the WebLogic Server as a target and register it. For details on discovering and registering the WebLogic Server target, see the *Oracle Fusion Middleware Administration Guide*. If this WebLogic server target is configured to connect to a database running on a remote machine, ensure that the Management Agent is deployed on the remote machine, and confirm that this database is listed as a target in Enterprise Manager Grid Control.
5. Register the ADP Manager. Navigate to the Middleware and click the **Application Dependency and Performance** link. In the Application Dependency and Performance page, click the **Registration** tab on the right panel. [Figure 11–2, "RMI Configuration Page"](#) is displayed.

Figure 11–2 RMI Configuration Page

Middleware Home > Application Dependency and Performance

Use time frame:  1 hour Start: 7/22/2010 2:33 PM End: 7/22/2010 3:33 PM End time is current system time

Monitoring Configuration Registration

Managers

- b-25
- e2e-Medrec**
- b-63

**RMI Configuration**

Manager Name: e2e-Medrec

Host (IP or Name): hp-ca.us.oracle.com

Port: 51099

Upload  Enable Request Monitoring

Home Directory: /home/oracle/camm1lg\_900

Key Store File: AcseraManagerKey.jks

Key Store Password: \*\*\*\*\*

Trust Store File: AcseraManagerTrust.jks

Trust Store Password: \*\*\*\*\*

Alias: mykey

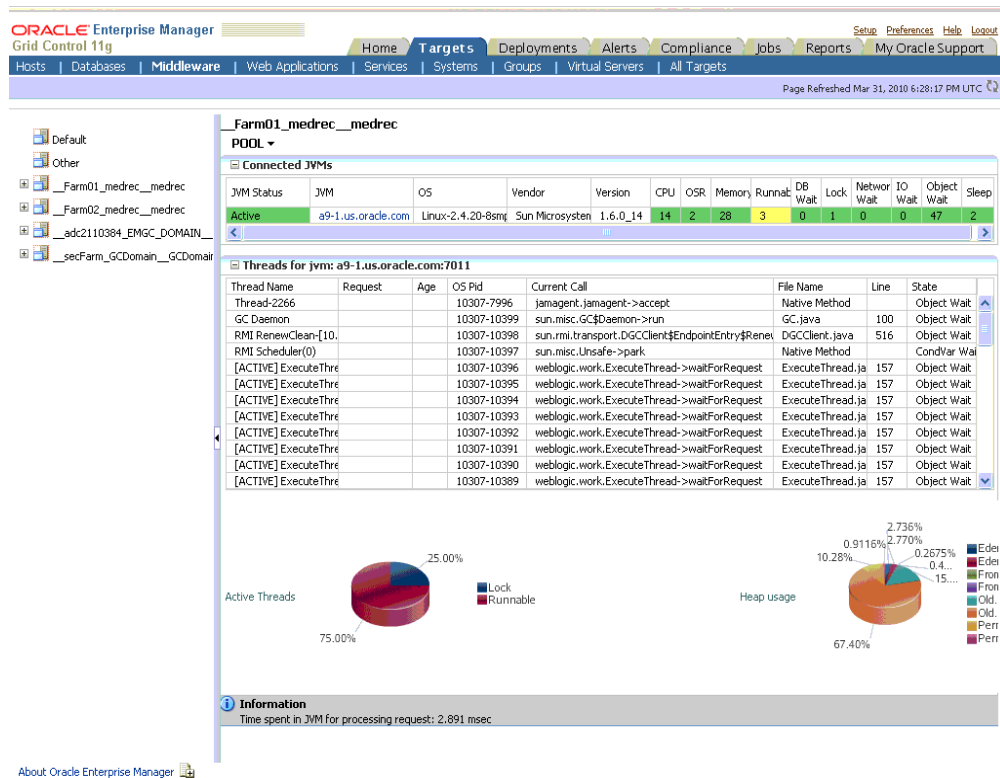
Agent: hp-ca.us.oracle.com:1831

Test Connect Add Remove Update

Enter the name of the ADP Manager to be registered and the fully qualified host name of the machine that the ADP Manager is running on. The default port 51099 is pre-populated for you. To ensure that this ADP Manager can collect request monitoring data, you must check the **Enable Request Monitoring** checkbox. Click **Test Connect** to check if you can successfully connect to the ADP Manager. Select the Management Agent from the Agent drop-down list and click **Add** to complete the ADP Manager registration. For more details on registering the ADP Manager, see the [Chapter 10, "Installing and Configuring Application Dependency and Performance"](#).

6. Install the JVM Diagnostics Manager and deploy the JVM Diagnostics Agent on WebLogic Server targets. For more details, see the [Chapter 9, "Configuring JVM Diagnostics"](#).
7. To validate that the JVM Diagnostics Manager is communicating with the JVM Diagnostics Agent, click the **JVM Diagnostics** link on the Middleware page. Click on a target in the left hand panel, for example `__Farm01_medrec__medrec`. In the right hand panel, click **Pool > Threads > Real-Time Analysis**. The following screen is displayed:

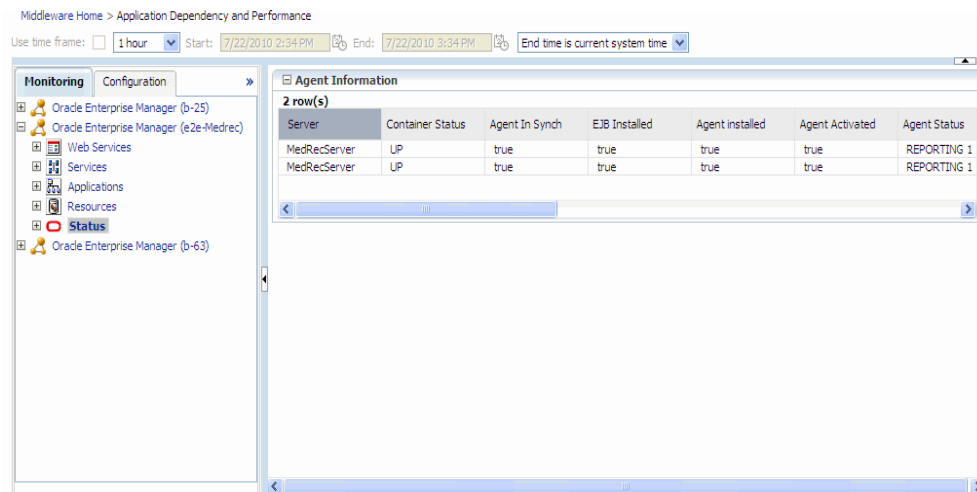
Figure 11–3 JVM Diagnostics All Threads Page



On this page, you can view the threads for each connected JVM.

8. Restart the ADP Manager and the WebLogic Server targets.
9. Click **Application Dependency and Performance** link on the Middleware page. Click the **Monitoring** tab on the right panel and select the **Status** option from the menu. The following screen is displayed.

Figure 11–4 Agent Information Page



You can validate that the ADP Agent has discovered the Java EE target by checking the Agent Status on this page.

10. Request Monitoring setup is now complete. To view and configure requests, grant the Request Monitoring Administrator privilege which provides full access to the Request Monitoring pages and allows the user to modify the request configuration settings. For richer Request Monitoring functionality, we recommend that you grant the following privileges:
  - View any Target privilege which allows user to drill down to the Target Home page and to view the target metrics in the Topology view.
  - JVM Diagnostics privilege which allows user to drill down to the JVM Diagnostics pages from the Topology page.
11. After the appropriate privileges have been granted, click the **Request Monitoring** link on the Middleware page and then click the **Request Definitions** tab. The following screen is displayed.

**Figure 11–5 Request Definitions Page**

Request Monitoring

Request Performance **Request Definitions**

Create Group... Partition... Edit... Remove...  Group Requests  View Only Key Requests

Name	Key	Critical Threshold (sec)	Warning Threshold (sec)	Last Modified	Modified By
medrec				5/3/2010 12:08 PM	SYSMAN
/mmedrec/Faces Servlet	✓	0.3	0.1	6/30/2010 6:40 AM	SYSMAN
/mmedrec/Faces Servlet (name1=value1)		0.3	0.1	6/25/2010 7:10 PM	SYSMAN
/mmedrec/Faces Servlet (name2=value2)		0.2	0.1	6/25/2010 7:48 PM	SYSMAN
/mmedrec/patient/viewLoginResult.jsp		3.0	2.0	6/22/2010 5:09 PM	SYSMAN
/mmedrec/Faces Servlet (name1=value2)		0.2	0.1	6/25/2010 5:24 PM	SYSMAN
physician				5/3/2010 12:08 PM	SYSMAN
soa-infra				5/11/2010 3:53 PM	SYSMAN
test				6/25/2010 2:53 PM	SYSMAN
mywebapp2				6/29/2010 5:53 PM	SYSMAN

On this page, you can:

- View and edit requests or request groups.
- Create or edit request groups.
- Partition an existing request by defining additional parameters for the request.
- Set request response time violation thresholds (critical and warning)



# Part IV

---

## JVM Diagnostics

The chapters in this part provide information regarding JVM Diagnostics.

Part III contains the following chapters:

- [Chapter 12, "Introduction to JVM Diagnostics"](#)
- [Chapter 13, "Using JVM Diagnostics"](#)
- [Chapter 14, "Troubleshooting JVM Diagnostics"](#)





---

---

## Introduction to JVM Diagnostics

This chapter provides an overview of JVM Diagnostics. It contains the following sections:

- [Section 12.1, "Overview"](#)
- [Section 12.2, "New Features in this Release"](#)
- [Section 12.3, "Supported Platforms and JVMs"](#)
- [Section 12.4, "User Roles"](#)

### 12.1 Overview

Mission critical Java applications often suffer from availability and performance problems. Developers and IT administrators spend a lot of time diagnosing the root cause of these problems. Many times, the problems occurring in production environments either cannot be reproduced or may take too long to reproduce in other environments. This can cause severe impact on the business.

Oracle Enterprise Manager Grid Control 11g R1's JVM Diagnostics enables administrators to diagnose performance problems in Java application 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 JVM Diagnostics, administrators will be able 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. The key features of JVM Diagnostics are:

- [Section 12.1.1, "Java Activity Monitoring and Diagnostics with Low Overhead"](#)
- [Section 12.1.2, "In-depth Visibility of JVM Activity"](#)
- [Section 12.1.3, "Real Time Transaction Tracing"](#)
- [Section 12.1.4, "Cross-Tier Correlation with Oracle Databases"](#)
- [Section 12.1.5, "Memory Leak Detection and Analysis"](#)
- [Section 12.1.6, "JVM Pooling"](#)
- [Section 12.1.7, "Real-time and Historical Diagnostics"](#)

### 12.1.1 Java Activity Monitoring and Diagnostics with Low Overhead

JVM Diagnostics provides in-depth monitoring of Java applications without slowing them down. It helps you to identify the slowest requests, slowest methods, requests waiting on I/O, requests using a lot of CPU cycles, and requests waiting on database calls. It also identifies the end-user requests that have been impacted by resource bottlenecks. Application resources that are causing the performance bottleneck are also visible.

### 12.1.2 In-depth Visibility of JVM Activity

JVM Diagnostics provides immediate visibility into the Java stack. You can monitor thread states and Java method/line numbers in real time and you can proactively identify issues rather than diagnosing issues like application crashes, memory leaks, and application hangs after they occur.

### 12.1.3 Real Time Transaction Tracing

If a particular request is hanging or if the entire application is slow, administrators can perform a real-time transaction trace to view current Java application activity. You can see the offending threads and their execution call stacks. You can also analyze various bottleneck resources such as how much time a thread spent in waiting for a database lock. Complex problems such as activity in one thread (or request) affecting the activity in the other thread or rest of the JVM can be found very quickly.

Sometimes the monitoring interval (normally 10 seconds) that is in use is too coarse grained. The Java thread of interest may be too short lived or the amount of monitoring data collected may be insufficient. In such cases, you can run a JVM Trace to get fine-grained details of the JVM activity. This feature allows you to monitor your Java application at a very high frequency (default of once every 50ms) for a short period of time. This allows you to identify interdependency of threads, bottleneck resources (DB, I/O, CPU, Locks, Network) & top methods.

### 12.1.4 Cross-Tier Correlation with Oracle Databases

JVM Diagnostics facilitates tracing of Java requests to the associated database sessions and vice-versa enabling rapid resolution of problems that span different tiers. It highlights the slowest SQL queries and helps administrators to tune SQL and the database to improve the performance application. This facilitates smooth communication between the database administrators and application administrators by isolating the problems to the database or the application tier.

### 12.1.5 Memory Leak Detection and Analysis

Memory leaks lead to application slowdowns and eventually cause applications to crash. JVM Diagnostics alerts administrators on abnormalities in Java memory consumption. Administrators can use JVM Diagnostics and take heap dumps in production applications without disturbing the application. They can take multiple heap dumps over a period of time, analyze the differences between the heap dumps and identify the object causing the memory leak. Heap analysis can be performed even across different application versions. Differential Heap Analysis with multiple heap dumps makes it easy to identify memory leaks.

## 12.1.6 JVM Pooling

JVM Diagnostics allows administrators to group sets of JVMs together into JVM pools. This provides the console user with a single view across all related JVMs. Hence all JVM's that make up a single application or a single cluster may be grouped together in an application. This allows administrators to visualize problems naturally and intuitively.

## 12.1.7 Real-time and Historical Diagnostics

With JVM Diagnostics, you can perform real-time and historical diagnostics on your Java applications. This provides you with detailed insight on the root causes of production problems without having to reproduce the same problem in a Test or QA environment. You can play back transactions interactively from the browser and view the time spent in the network and the server. You can also view the response time breakdown by Servlet, JSP, EJB, JDBC, and SQL layers.

Apart from the real-time data, you can also analyze historical data to diagnose problems that occurred in the past. You can view historical data that shows the time taken by end-user requests and the breakdown by Servlet, JSP, EJB, JDBC, and SQL layers.

## 12.2 New Features in this Release

This section lists the new JVM Diagnostics features in Oracle Enterprise Manager Grid Control 11g R1.

- JVM Diagnostics is now integrated with Oracle Enterprise Manager Grid Control.
  - Some of the pages in the old console have been converted to the ADF format.
  - The Enterprise Manager repository is now used for JVM Diagnostics.
  - Enterprise Manager Grid Control users can now perform JVM Diagnostics tasks depending on their privileges.
- Secure communication between JVM Diagnostics Manager and Agent is now supported.

### 12.2.1 One-off Patch Enhancements

The features listed in this section are available only if you have applied the one-off patch.

- Multiple JVM Diagnostics Managers: You can now deploy multiple JVM Diagnostics Managers and connect one or more agents to these managers. For more details, refer to the *Scaling Out by Deploying Multiple JVM Diagnostics Managers* section.
- Level Zero (L0) Tracing: If the Application Dependency and Performance Agent is running and Request Monitoring has been configured, additional data is displayed in the JVM Diagnostics screens. For example, you may see:
  - Average Age on Top Requests for Domain (Pool) Summary page. Click a Request Name to view the request flow details.
  - In the Domain (Pool) or Real Time Analysis page, the Age, ECID, and RID is also displayed.
- SQL Identifier Support - You can now perform queries and search for a SQL ID corresponding to a particular SQL text or SQL hash value.

- Application Development Framework (ADF): With ADF support, entity objects and view objects are also displayed in the top metrics section
- Modifying the `web.xml` File: You now have an option to modify the `web.xml` file for the JVM Diagnostics Agent. For details, see [Section 13.7.7, "Download Agent Page"](#).

## 12.3 Supported Platforms and JVMs

The JVM Diagnostics Manager is supported on:

- Windows (2000 or later)
- Linux x86 (RHEL 3 or later)
- Solaris SPARC (8 or later)

The JVM Diagnostics Database Agent supports Oracle 8i or later on the following platforms:

- Linux x86, Linux x86\_64 and IA64
- Solaris SPARC and x86
- AIX PowerPC
- HP-UX 11 on PA-RISC and IA64

The JVM Diagnostics Agent supports the following JVMs:

- Sun JDK - 1.3, 1.4, 1.5, and 1.6
- JRockit - 1.4, and 1.5
- HP JVM - 1.4, 1.5, and 1.6
- IBM JVM - 1.4, 1.5, and 1.6

See the *My Oracle Support* site for the latest certification details.

---

---

**Note:**

- We recommend that you use Sun JDK 1.6.0\_18 32-bit for the JVM Diagnostics Manager.
  - You can use 64-bit JVM for the JVM Diagnostics Agent.
- 
- 

## 12.4 User Roles

To use JVM Diagnostics, you must have either of the following Enterprise Manager system privileges:

- JVM Diagnostics User: Allows you to view JVM Diagnostics data.
- JVM Diagnostics Administrator: Allows you to manage JVM Diagnostics operations.

You can define these privileges in the Setup pages. For more details on defining these privileges, see the *Enterprise Manager Security* chapter in the *Administration Guide*.

---

---

## Using JVM Diagnostics

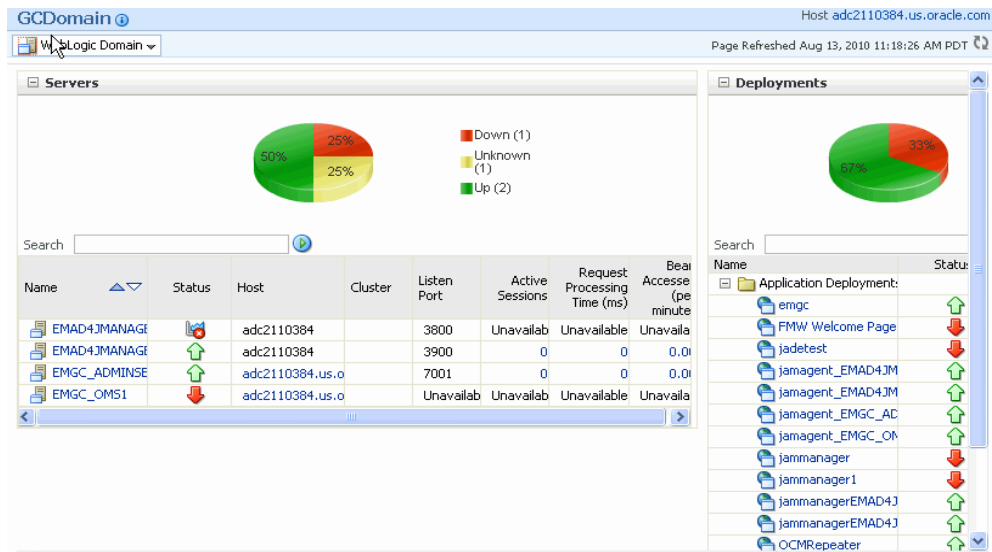
This chapter describes the various tasks you can perform using JVM Diagnostics. In particular, it contains the following:

- [Section 13.1, "Accessing the JVM Diagnostics Pages"](#)
- [Section 13.2, "Viewing the Domain \(Pool\) Summary and Details Pages"](#)
- [Section 13.3, "Viewing the Domain \(Pool\) Real Time Thread Analysis Page"](#)
- [Section 13.4, "Viewing the JVM Summary and Details Pages"](#)
- [Section 13.5, "Viewing the JVM Real Time Thread Analysis Page"](#)
- [Section 13.6, "Viewing the JVM Real Time Memory Analysis Page"](#)
- [Section 13.7, "Using the Setup Pages"](#)
- [Section 13.8, "Analyzing the JVM Heaps"](#)
- [Section 13.9, "Using the Database Pages"](#)

### 13.1 Accessing the JVM Diagnostics Pages

After the JVM Diagnostics Manager has been installed, you can start using the features. Navigate to the Middleware tab in Enterprise Manager and click on a WebLogic target. The WebLogic Domain Home page is displayed.

Figure 13–1 WebLogic Domain Page

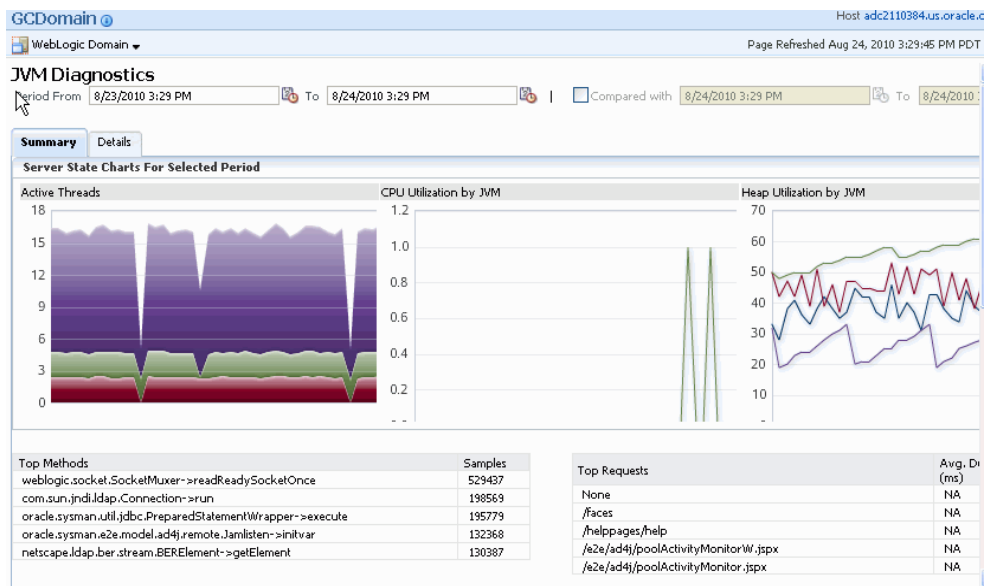


To start using JVM Diagnostics, select the JVM Diagnostics option from the WebLogic Domain drop-down menu and choose the appropriate option.

## 13.2 Viewing the Domain (Pool) Summary and Details Pages

You can view the summary and detailed information of the JVM pools in the selected WebLogic domain on this page. You can also compare the JVM pool data across two periods in the domain. To view this page, click the **Summary** option from the JVM Diagnostics menu.

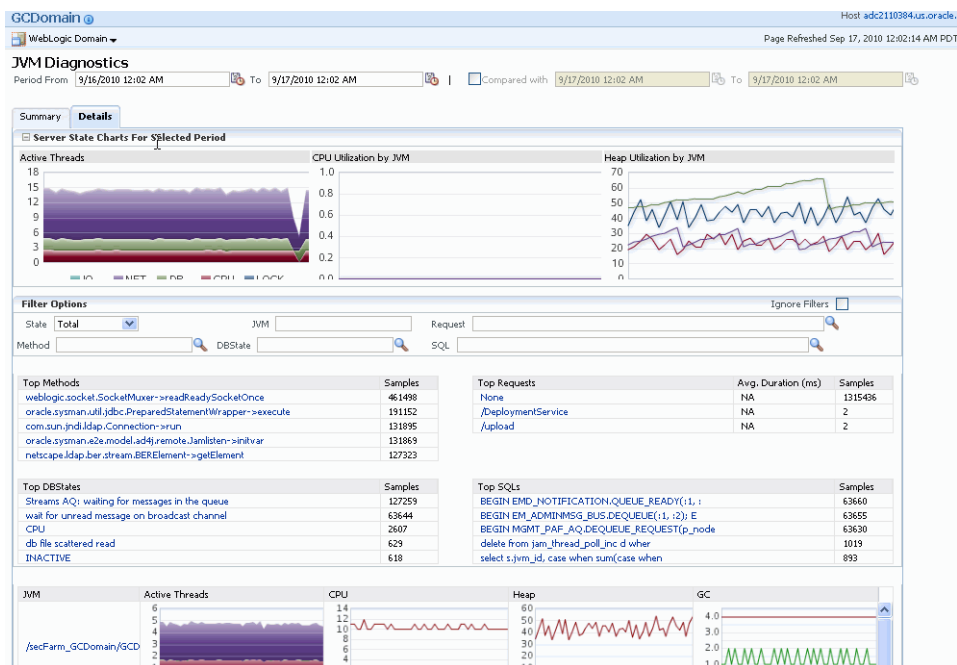
Figure 13–2 Domain Summary Page



This page contains the Summary and Details tabs. The Summary tab shows the summary details of the JVM pools which include the Server State Charts, and a list of Top Methods and Top Requests.

Click the **Details** tab to view the Details page.

**Figure 13–3 Domain Details Page**

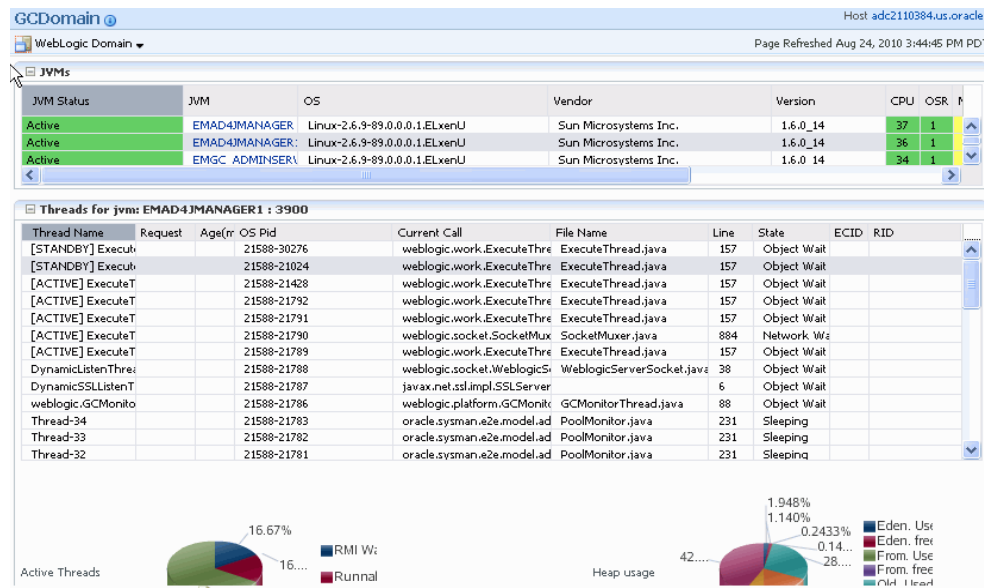


You can view the Server State Charts, list of Top Methods, Top Requests, Top DBStates, and Top SQLs. You can filter the data that is displayed by specifying various criteria such as Method Name, JVM Name, Thread State, DBState, and so on. You can also view server state charts for each JVM in the pool.

### 13.3 Viewing the Domain (Pool) Real Time Thread Analysis Page

This page shows the real-time data for all the JVMs in the selected domain. To view this page, click **Threads** and **Real-Time Analysis** from the JVM Diagnostics menu.

Figure 13–4 Domain Real-Time Thread Analysis Page



The first table on this page shows the list of JVMs and the current status of each JVM. The current activity of the JVM including CPU usage, memory, number of threads waiting for a database response, number of threads waiting for synchronization lock, and other details are displayed.

Click on a JVM link in the JVM column to view the list of threads for the JVM. You can also see two charts, the first one displays the active threads in the JVM, the second one displays the heap status for the JVM.

## 13.4 Viewing the JVM Summary and Details Pages

This page shows the summary and detailed information for a specific JVM. To view this page, navigate to a WebLogic Domain Home page, select a JVM from the left panel, click **JVM Diagnostics** and the **Summary** option from the menu.



Figure 13–5 JVM Summary Page



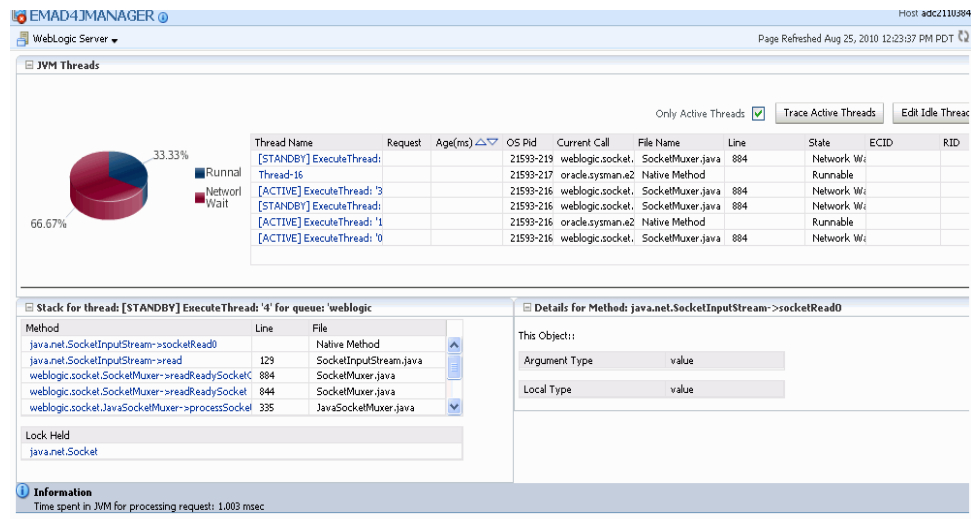
This page contains the Summary and Details tabs. The Summary tab shows the summary details of the JVM pools which include the Server State Charts, Top Methods, and Top Requests charts.

Click the **Details** tab to view the Details page. You can view the Server State Charts, list of Active Threads by State, Top Methods, Top Requests, Top DBStates, and Top SQLs. You can filter the data that is displayed by specifying various criteria such as Method Name, JVM Name, Thread State, DBState, and so on.

## 13.5 Viewing the JVM Real Time Thread Analysis Page

This page shows the real-time data for a selected JVM. To view this page, navigate to a WebLogic Domain Home page, select a JVM from the left panel and click **JVM Diagnostics**. From the JVM Diagnostics menu, click **Threads** and **Real-Time Analysis**.

Figure 13–6 JVM Real-Time Analysis Page



This page shows the following:

- JVM Threads:** This table shows detailed information for all threads running in the JVM. Check the **Only Active Threads** checkbox to view only the active threads running in the JVM. Click **Trace Active Threads** to monitor all the active threads at a high frequency for a short period of time. For threads in DB Wait state, you can drill down to the DB Session page. Click **Edit Idle Thread Rule** to drill down to the Idle Thread Rule page. On this page, you can create Idle Thread Rules and add threads to the rule. All threads part of the rule will be marked as idle threads and will not be monitored.
- Thread State Chart:** This chart groups threads according to their state. For example, the chart shows threads in network wait state, threads in runnable state, object wait state and so on.
- Thread Stack:** Click on a Thread Name in the JVM Thread table to see the call stack for the thread.
- Method Locals:** Click on the Method Name to see method locals in bottom right panel.
- Object Details:** Click on an Object Name to see the object details.

## 13.6 Viewing the JVM Real Time Memory Analysis Page

This page shows the real time organization of all objects in the JVM Heap. To navigate to this page, click the **JVM Diagnostics** link in the Middleware page. Click **All** in the left panel to do a real time memory analysis of all the objects in the JVM Heap.

## 13.7 Using the Setup Pages

This section describes the pages under the Setup menu. To access these pages, click **Setup** option from the JVM Diagnostics menu.

## 13.7.1 Console Setup Page

This page is used to set console parameters and reset lost passwords. You can specify the following parameters on this page:

- **Console Log Level, Cross Tier Log Level:** Verbosity of console log messages. Increase only when advised by Oracle Support.
- **Agent Request Timeout, Agent Loop Request Timeout:** Increase if the monitored JVMs are extremely busy and the console times out and disconnects waiting for their response.
- **Require Login for Java Tab:** Enable this checkbox to restrict access to other (non-setup Tabs). Users will need to login with valid accounts (created from the Users screen) and will have access to all non-setup screens. The setup screens can only be accessed by the ADMIN user.
- **Enable Monitoring:** With this checkbox enabled, JVMs belonging to selected pools will be polled for active requests periodically. This data is available on the NOC and in the last 90 seconds/24 hrs charts.
- **Email Alerts:** Email alerts based upon threshold violations (requires monitoring to be enabled).
- **SNMP Host for Alerts:** SNMP Host to which the alerts should be sent.

Click **Save** to set the new parameters or **Reset** to use the existing parameters.

## 13.7.2 JVM Pool Info

You can group sets of JVMs into JVM pools that provide monitoring information across all related JVMs in a single view. This table displays all the JVM pools in the system. You can view all available pools, delete existing pools or add new pools. The **Default** and **Other** pools cannot be deleted. Any user defined pools can be deleted.

Each row displays whether polling is currently enabled for the pool, the polling interval, and the email notification recipient. If the Polling Enabled flag is set to Y, JVMs belonging to this pool will be polled for active requests periodically based on the Poll Interval. You can click on the **Edit** icon to edit the JVM pool or click on **New Pool** to add a new JVM Pool.

### 13.7.2.1 Updating a JVM Pool

To update a JVM Pool, click the **Edit** icon in the Show JVM Pools page. In the Edit JVM Pool page, you can specify the Poll Interval, enable monitoring for this pool, and send email alerts to a specified list.

### 13.7.2.2 Update Pool Thresholds

Threshold violations can have a **Level** of **R** (Red) or **Y** (Yellow). The **Metric** is the attribute being monitored and the **Threshold** is the value against which that metric is being compared. A violation occurs when the threshold is exceeded after a minimum number of samples have been monitored. Threshold units are % CPU Utilization for Machine CPU (CPU), JVM CPU (CPU), % Heap Utilization for Heap, number of threads in that state for others. 3 to 5 threads active in DB for more than 50% of the samples will cause an amber (Y) alert for DB, while 6+ threads active in DB wait state for more than 50% of the samples will cause a red alert for DB.

**Action URL** is the URL to be invoked when a threshold violation occurs. This includes internal and external URLs. The Action URL can be any valid URL on a remote

system. It can be used to trace a particular thread, all active threads, or dump a heap in response to a threshold violation.

The Action URL should be a valid URL as called from a browser. If not specified, default parameters for the `traceThread` and `heapdump` will be added to the URL. If the Action URL specifies a host that is different from the current console host, you must navigate to the Console Setup page and uncheck the **Require Login for Java Tab** checkbox.

A sample URL template for Tracing Threads is given below:

```
http://localhost/jvmTraceActive.jsp?traceThread=allactive&JVM_THREAD_ID=&JVM_ID=1&JVM_MACHINE=machine_name&JVM_PORT=8080&pollInterval=50&pollDuration=10&samplesfilename=traceactive&detailsfile=on
```

where:

- `traceThread` is the name of the thread to be traced. If you want to trace all the active threads, specify the value as `traceThread=allactive`. This is the value in the Thread column in the View All Threads page.
- `JVM_THREAD_ID` is the ID of the thread to be traced. If you want to trace all active threads or if `traceThread=allactive`, the `JVM_THREAD_ID` parameter must be blank. This is the value specified in the OS PID column in the View All Threads page.
- `JVM_ID`, `JVM_MACHINE`, `JVM_PORT` are the ID, machine name, and port number of the JVM Agent on which the thread or threads to be trace are running. For example, if the text displayed at the top of the View All Threads page is - Name: machine\_name:8080 ID: 1: Linux-2.6.9-55.0.0.0.2.EL: Sun Microsystems Inc.: 1.6.0\_07
  - `machine_name` is the `JVM_MACHINE`
  - 8080 is the `JVM_PORT`
  - 1 is the `JVM_ID` (If the `JVM_ID` parameter is not supplied in the URL, it is automatically appended by the Console)
- `PollInterval` is the time interval (in milliseconds) between successive samples. The default value is 50.
- `PollDuration` is the duration (in seconds) for the trace. The default value is 10.
- `Samplesfilename` is the prefix of the trace sample file name created on the console machine. The default value is `traceactive` and a unique identifier is added as a suffix to each trace file that is created.
- `Detailsfile` is the sample thread stack in addition to the state. You must set this to On if you require complete thread details. The default is On.

A sample URL for heapdump is given below:

```
http://localhost/jvmHeapDump.jsp?JVM_ID=1&JVM_MACHINE=machine_name&JVM_PORT=8080&JVM_DUMP_FILE=heapdump <http://localhost/jvmHeapDump.jsp?JVM_ID=1&JVM_MACHINE=machine_name&JVM_PORT=8080&JVM_DUMP_FILE=heapdump>
```

The following parameters are required in the URL for **heapdump**.

- `JVM_ID`, `JVM_MACHINE`, and `JVM_PORT` are the same parameters as specified for the Tracing Threads template.

- `JVM_DUMP_FILE` is the prefix of the heap dump file name on the agent. A unique identifier is added as a suffix to each heap dump file.

### 13.7.3 Databases Page

Use this screen to view information about the databases being monitored/profiled currently or to add additional databases for the same. Click the **Edit** icon to edit an existing database or click **Register NewDB** to register a new database.

### 13.7.4 JVMs and Managers Page

This page lists all the JVMs that have connected to the console. All the registered JVM Diagnostics Managers along with IP address, port number, build version, and status are listed on this page. You can edit the JVM parameters using the **Edit** icon, or delete old JVMs that no longer connect to the console.

In the Edit JVM Information page, you can assign a JVM to a different pool, change its log level (if recommended by Support) or change the temporary directory where the heap snapshots are dumped.

### 13.7.5 Monitoring

Once you have updated a JVM pool with polling enabled, you can monitor the JVMs in that pool. Click on the **Start Monitoring** link above to start monitoring. This can also be done from the Console Setup Screen. You can enable or disable monitoring of pools or change their polling intervals by updating the pool properties.

You can mark a thread as idle by adding it to an Idle Thread Rule. All threads that have been marked as idle will not be monitored. Click **New Rule** to create a new Idle Thread Rule. The Rule Type can be:

- **Monitor (Waiting on Lock)**: Select this type if you want to ignore all threads that are locked.
- **Current Call**: Select this type if you want to ignore all threads that are making a call to the selected function.

The Rule Value should contain the class name, method, followed by class+method. An example of a Current Call is `weblogic.socket.PosixSocketMuxer->processSockets`. An example of a Monitor (Waiting on Lock) is `weblogic.socket.PosixSocketMuxer$1`.

All threads that meet the criteria specified in the Idle Thread Rule will not appear in the View Active Threads screen.

### 13.7.6 Trace Files Page

Use this page to:

- Delete trace files from the traces you have run.
- Load a trace file from another console. You'll need to move the trace file(s) to the `jamserv/trace` directory for them to be visible in this screen.

### 13.7.7 Download Agent Page

On this page, you can manually download the various binaries such as Java Agent, JVM Diagnostics Manager, Database Agent, Load Heap, and deploy them.

---

---

**Note:** If you are using WebLogic Server 10.x and above, we recommend you use the automated deployment procedure as described in [Section 9.5.1, "Automated Deployment Procedure"](#).

---

---

### 13.7.7.1 Downloading the Java Agent

On the Download Agent page, click the **Download** icon next to the **Java Agent (AD4J Agent/Jamagent) WAR file** component. In the JAM Agent web.xml Parameters page, select an Available Manager from the list, modify the Tuning, Target Association, and Logging parameters and click **Download**. The `jamagent.war` will be downloaded onto the specified system.

## 13.8 Analyzing the JVM Heaps

This section describes the pages under the Saved Heap Snapshots menu option. To access these pages, click **Saved Heap Snapshots** option from the JVM Diagnostics menu.

### 13.8.1 All Heaps Page

This page displays a list of all the heaps that have been loaded into the repository. You can see heap details like JVM Name, vendor, heap size, heap used. Click on the link in the % column to drill down to a specific heap and view the memory usage of the heap. Click the **Edit** icon to edit the Heap Snapshot information.

### 13.8.2 Heap Usage Page

This page provides a summary of heap usage by different types of objects in the heap. The type of object (Klass, Array, Instance), the total number and size of objects in each heap.

### 13.8.3 Heap Usage by Roots

This page shows the number of objects and amount of heap memory reachable by each class of root. The name of the root, the total number of objects reachable from the root and the adjusted memory (used to identify memory leak hot-spots) reachable from the root are displayed.

- Click the link in the Root column to view the top reachable objects in the heap.
- The View Summary option is available if you reach this page after selecting the heaps to be compared. Click this option to do a bottom up analysis of the heap.

### 13.8.4 Top 40 Objects Page

This page shows the top 40 objects reachable from a root. The objects are sorted in descending order by the ascending memory reachable from the object (or the difference of the adjusted memory reachable when comparing two heaps). The internal root identifier of the object, type of object, signature, number of children reachable from the object, and total and adjusted memory from the object are displayed. Click on the **Signature** link to drill down to the Heap Object Information page.

If you reach this page while comparing two heaps, you will see the type of object, number of objects of each type, number of children reachable from the object, adjusted memory reachable from this object, and the delta. The Delta Adj column shows the

difference in the total memory reachable from these objects. Click the **Delta Adj** link to compare the objects in the two heaps. If there are multiple objects of a given time/signature present in a given heap, this action will display a selection page to choose an object from each heap to compare.

---



---

**Note:** If the second heap does not have the corresponding objects of the same type and signature as the first heap, these columns in the second heap will be blank.

---



---

### 13.8.5 Heap Object Information Page

This page shows information about the objects in the heap. You can see the following:

- **Heap Object Information:** This table shows information about a specific object in the heap snapshot.
- **Object Information:** This table shows information about all object's children (like fields and array entries). Click on the signature to examine the child.
- **Object Children:** This table shows information about all object's parent (from where this object is reachable). Click on the signature to examine the parent.
- **Roots:** This table shows any roots from which this object is directly reachable.

### 13.8.6 Compare Heaps Page

On this page, the list of available of heaps for comparison are displayed. Click on the % column to select the second heap for comparison. The Show Heap Usage by Roots page is displayed. Click **View Summary** to see the details of the heaps being compared.

## 13.9 Using the Database Pages

This section describes the pages under the JDBC Correlation menu option. After you have deployed and registered the database agent as described in [Section 9.7, "Deploying the Database Agent"](#), you can view the activity on the registered databases using this option.

To access these pages, click **JDBC Correlation** option under the JVM Diagnostics menu. The following pages are available:

- **DB Activity:** This page shows a summary of databases monitored by the console. Click on the **View Details** link to view the database session details.
- **DB Waits:** This page shows the session details of the selected database. Click on the **Event** or **Idle Event** link to drill down to the DB Waits page.
- **Current IO:** This page shows the current IO activity on the selected database.
- **Current CPU:** This page shows the current CPU activity on the selected database.
- **Current Locks:** This page provides a summary about the sessions holding and waiting for locks.
- **Execution Plan:** This page shows the SQL Statement Execution plan. Click **Get Plan** to view the execution plan and the table statistics.

- **Find Sessions:** This screen provides attributes and details for the database session waiting for the lock. Click the **Thread Information** link to get the Java details for the screen.

## 13.10 Tracing JVM Threads

To trace JVM threads, you can:

- Click **JVM Diagnostics > Threads > Trace Now**.
- Click **Threads > Real Time Analysis**. On the Real Time Analysis page, click **Trace Active Threads** to navigate to the Trace Active Threads page.

### 13.10.1 Traces Active Threads Page

You can trace all active threads on this page. During tracing, the state and stack of the target thread is sampled at set time intervals for the desired duration. To access this page, click **Trace Active Threads** in the JVM Real Time Thread Analysis page. See [Section 13.5, "Viewing the JVM Real Time Thread Analysis Page"](#) for details.

You can specify the following parameters before you start the trace:

- **Poll Interval:** The time interval between successive samples.
- **Duration:** The duration of the trace.
- **Trace Thread Details:** Sample thread stack in addition to the state will be traced.
- **Try Changing Thread:** If a thread stack changes during a sample (this can happen when a thread is using CPU), JVM Diagnostics will skip that thread for that sample. If you find missing samples, use this feature to retrace the changed stacks.
- **Include Network Waits:** Indicates if idle network waits should be included in the trace.
- **Trace All Threads:** Select this checkbox to include both idle and active threads in the trace.
- **Allow Trace Interrupt:** Allows you to interrupt the trace process.

After the tracing has been successfully completed, click **Goto Saved Trace** to view the trace results.

### 13.10.2 Trace Summary Page

This page shows the state, method, and stack information for a trace. To access this page, click **Goto Saved Trace** on the Trace Active Threads page.

- **JVM Trace Results:** This section provides information about when and where the trace was run. It also shows any filters that have been applied to the results of tracing.
- **Resource Usage:** This section shows the system activities such as CPU utilization, heap usage, and garbage collection during tracing.
- **Top States:** This shows the JVM State Analysis chart for the duration of the trace. The Y axis shows how many threads were in that state at that time.
- **Top Methods:** This shows the relative time spent on different methods during the trace.



- **Threads by State:** This table Threads breaks down the relative time spent in different states for each method. The **Thread** link displays filters just for a specific thread.
- **Requests by State:** This table shows the number of samples in each state for each request.
- **Methods by State:** This table breaks down the relative time spent in different state for each method. The **Method** link shows the call stacks, which include the method.

### 13.10.3 Available Threads Page

This table lists all the traces that have been loaded into the repository using the **Trace Active Threads** option. The Thread column indicates if all active threads or a specific thread has been traced. The number of samples taken during the trace are displayed. Click the **Samples** link to drill down to the Trace Summary page. Click the **Edit** icon to modify the trace information for a set of samples.

### 13.10.4 Thread Summary Page

This page shows the state, method, and stack information for a specific thread in a trace. To access this page, click a **Thread** link in the Threads by State column in the Trace Summary page.

- **JVM Trace Results:** This section provides information about when and where the trace was run. It also shows any filters that have been applied to the results of tracing.
- **Resource Usage:** This section shows the system activities such as CPU utilization, heap usage, and garbage collection during tracing.
- **Top States:** This shows the JVM State Analysis chart for the duration of the trace. The Y axis shows how many threads were in that state at that time.
- **Top Methods:** This shows the relative time spent on different methods during the trace.
- **Threads by State:** This table breaks down the relative time spent in different states for each method. The **Thread** link displays filters just for a specific thread.
- **Requests by State:** This table shows the number of samples in each state for each request.
- **Methods by State:** This table breaks down the relative time spent in different state for each method. The **Method** link shows the call stacks, which include the method.



---

---

## Troubleshooting JVM Diagnostics

This chapter describes the errors you may encounter while deploying and using JVM Diagnostics and the workaround steps you can follow to resolve each of them. It contains the following sections:

- [Section 14.1, "Agent Automated Deployment Errors"](#)
- [Section 14.2, "Secure Communication Errors"](#)
- [Section 14.3, "Cross Tier Functionality Errors"](#)
- [Section 14.4, "Trace Errors"](#)
- [Section 14.5, "Deployment Script Execution Errors"](#)
- [Section 14.5.1, "Deployment on 64-bit JVMs"](#)
- [Section 14.6, "LoadHeap Errors"](#)
- [Section 14.7, "Errors on JVM Diagnostics UI Pages"](#)
- [Section 14.8, "Frequently Asked Questions"](#)

### 14.1 Agent Automated Deployment Errors

This section lists errors that occur during the automated deployment of the JVM Diagnostics Agent.

## 14.1.1 DeployAd4jAgentOnTarget Step Errors

**Table 14–1 DeployAD4JAgentOn Target Step Errors**

Error Message	Workaround Steps
<p>New SOA Composite deployed on the SOA Server from JDeveloper are not displayed automatically in Enterprise anager Grid Control. war file &lt;war file name&gt; is already deployed. Use Option force to redploy</p>	<p>Navigate to the Agent Deployment page, check the <b>Force</b> checkbox and click <b>Deploy</b>.</p>
<p>Caused by: java.lang.SecurityException: User: &lt;username&gt;, failed to be authenticated</p>	<p>This error occurs if the username and password for the WebLogic Administration server is incorrect.</p> <p>Specify the correct credentials for the WebLogic Administration server and click Deploy on the Agent Deployment page.</p>
<pre>Exception in thread "main" java.lang.NoClassDefFoundError: javax/enterprise/deploy/spi/exceptions/TargetExc eption Caused by: java.lang.ClassNotFoundException: javax.enterprise.deploy.spi.exceptions TargetException at java.net.URLClassLoader\$1.run(URLClassLoader? .java:200) at java.security.AccessController.doPrivileged(Nati ve Method) at java.net.URLClassLoader.findClass(URLClassLoader? .java:188) at java.lang.ClassLoader.loadClass(ClassLoader? .java:307) at sun.misc.Launcher\$AppClassLoader.loadClass(Launc her.java:301) at java.lang.ClassLoader.loadClass(ClassLoader? .java:252) at java.lang.ClassLoader.loadClassInternal(ClassLoa der? .java:320) Could not find the main class: oracle.sysman.e2e.model.ad4j.util.RebuildWar. Program will exit.</pre>	<p><b>Reason:</b> This error may occur if:</p> <ul style="list-style-type: none"> <li>■ The path to the <code>weblogic.jar</code> (present on the machine on which the Management Agent is running) as specified on the Agent Auto Deployment page.</li> <li>■ The user does not have Read permissions on the directory on which the <code>weblogic.jar</code> file is present.</li> </ul>

## 14.1.2 CreateTempDir Step Errors

**Table 14–2 CreateTempDir Step Errors**

Error Message	Workaround Steps
RemoteOperationException: ERROR: Invalid username and/or password"	If the Management Agent monitoring the domain is running on a Windows machine, you may encounter this error if you specify an incorrect username or password for this machine. This error may also occur if you do not have the logon as batch job privilege.
nmo set suid root	You may see this error if you do not have Read permissions on the binary and the entire path.  To resolve this error, enter the command, <code>chmod 7755 nmo</code> to change the permission of <code>emagent11g/bin/nmo</code> to <code>suid root</code> .
<pre>Exception in thread "main" java.lang.NoClassDefFoundError: javax/enterprise/deploy/spi/exceptions/TargetException Caused by: java.lang.ClassNotFoundException: javax.enterprise.deploy.spi.exceptions TargetException at java.net.URLClassLoader\$1.run(URLClassLoader? .java:200) at java.security.AccessController.doPrivileged(Native Method) at java.net.URLClassLoader.findClass(URLClassLoader? .java:188) at java.lang.ClassLoader.loadClass(ClassLoader? .java:307) at sun.misc.Launcher\$AppClassLoader.loadClass(Launcher.java:301) at java.lang.ClassLoader.loadClass(ClassLoader? .java:252) at java.lang.ClassLoader.loadClassInternal(ClassLoader? .java:320) Could not find the main class: oracle.sysman.e2e.model.ad4j.util.RebuildWar. Program will exit.</pre>	<p>This error occurs if:</p> <ul style="list-style-type: none"> <li>the path to the <code>weblogic.jar</code> (present on the machine on which the Management Agent is running) as specified on the Agent Auto Deployment page.</li> </ul>

## 14.1.3 Active State Error

**Table 14–3 CreateTempDir Step Errors**

Error Message	Workaround Steps
Error: Ensure that Ad4j Manager is Active	You can deploy the JVM Diagnostics Agent only after the JVM Diagnostics Manager has been installed and is in an active state.

## 14.2 Secure Communication Errors

**Table 14–4 Secure Communication Errors**

Error Message	Workaround Steps
<p>JAM Agent: loadNative Exception loading [/tmp/libjamagent_nz.so.1] /tmp/libjamagent_nz.so.1: libnzn11.so: cannot open shared object file: No such file or directory. JAM Agent: Please make sure that libnzn11 and libclntsh is in the LD_LIBRARY_PATH</p>	<p>This error occurs if the NZ libraries are not in LD_LIBRARY_PATH.</p> <p>To resolve this issue, add the libnzn11.so and libclntsh.so.11.1 files to the LD_LIBRARY_PATH.</p>
<p>Jam Communication : Open wallet error : 28759 Jam Communication : setup_auth failed 28759</p>	<p>This error occurs if an unrecognized wallet resource locator has been used to open the wallet.</p> <p>To resolve this error, ensure that the wallet location is preceded by file.</p>
<p>Jam Communication : Open wallet error : 28759 Jam Communication : setup_auth failed 28759</p>	<p>This error occurs when the wallet file cannot be opened by the JVM Diagnostics Agent.</p> <p>Ensure that the file path to the wallet is correct and the correct permissions have been provided. While specifying the wallet location in the web.xml file, do not specify the absolute path to the wallet.</p>
<p>Jam Communication : nz initialize SSL handshake error 28864 Jam Communication : nzos_Write error - 29031 AM Console: Invalid connection from 10.228.234.246.55859</p>	<p>These errors occur if the JVM Diagnostics Manager has been launched in non-secure mode.</p>
<pre> javax.crypto.IllegalBlockSizeException: Input length must be multiple of 8 when decrypting with padded cipher at com.sun.crypto.provider.SunJCE_f.b(DashoA13? *..) at com.sun.crypto.provider.DESedeCipher.engineDoFinal (DashoA13? *..) at javax.crypto.Cipher.doFinal(DashoA13? *..) at jamagent.jamagent.decrypt(jamagent.java:725) at jamagent.jamagent.decrypt(jamagent.java:741) at jamagent.jamagent.&lt;init&gt;(jamagent.java:799) at jamagent.jaminit.init(jaminit.java:50) </pre>	<p>This error occurs if the encrypted password specified for the JVM Diagnostics Agent is incorrect.</p> <p>Enter the correct password in web.xml file.</p>

**Table 14–4 (Cont.) Secure Communication Errors**

Error Message	Workaround Steps
Jam Communication : Setting up context Jam Communication : nz initialize SSL handshake error 29024 Jam Communication : nzos_Write error :- 29031	This error occurs if the wallet password is incorrect or the certificate for the JVM Diagnostics Agent has not been certified by the same CA as the JVM Diagnostics Manager.
Jam Communication : nz initialize SSL handshake error 28860 Jam Communication : nzos_Read error : 28865 Console Err code: -1716291680 Receiving data on port 3600 from 127.0.0.1:57187, FD 450: Broken pipe	
JAM Agent: Conn reset by console JVM Diagnostics Manager Error: JAM Communication: nz initialize SSL handshake error 29049 JAM Communication: nzos_Read error: 28865 Console Err code: -1716291680 Receiving data on port 3600 from 127.0.0.1:58221, FD 450: Transport endpoint is not connected	This error occurs if the JVM Diagnostics Agent is running in non-secure mode and is trying to connect to a secure JVM Diagnostics Manager

## 14.3 Cross Tier Functionality Errors

This section lists the errors that show the status of the JVM Diagnostics Manager.

**Table 14–5 Cross Tier Functionality Errors**

Error Message	Workaround Steps
<p>DBWait link not displayed on JVM Threads Real Time Analysis page. No data displayed in Top DBStates / SQLs tables.</p>	<p>Cross tier functionality errors may occur due the following:</p> <ul style="list-style-type: none"> <li>■ Incorrect database credentials</li> <li>■ Database Agent errors</li> </ul> <p>If the database credentials are incorrect:</p> <ul style="list-style-type: none"> <li>■ Navigate to <b>Middleware &gt; JVM Diagnostics &gt; Setup &gt; Databases</b> page and verify the credentials for the registered database to be monitored.</li> <li>■ Enter the id of user who has installed the database in the OS User field.</li> <li>■ Specify the database application user credentials in the DB User field.</li> <li>■ Specify the database system user credentials in the DB User (Explain Plan) field.</li> </ul> <p>If database agent errors occur, ensure that the database agent is running on a machine on which the database is installed with the correct IP address and port number.</p> <ul style="list-style-type: none"> <li>■ Navigate to the <b>Threads &gt; Real Time Analysis</b> page for the JVM. If you see a thread that is in the DB Wait state, it should be a hyperlink. Click on the hyperlink to drill down to the database details and view the database session information including the SQL statements executed.</li> <li>■ If the DB Wait state is not a hyperlink, navigate to the <b>Middleware &gt; JVM Diagnostics &gt; Setup</b> page and set the <b>Cross Tier Log Level</b> to 6 and send AD4J Manager logs report issue.</li> </ul>

## 14.4 Trace Errors

This section lists errors that occur during tracing.

**Table 14–6 Trace Errors**

Error Message	Workaround Steps
<p>weblogic.transaction.internal.TimedOutException: Transaction timed out after 30 seconds</p>	<p>This error occurs if the Poll Duration has a large value and results in a timeout.</p> <p>This error does not affect the Trace functionality and can be ignored.</p>

## 14.5 Deployment Script Execution Errors

This section lists the errors that occur when you run the deployment script.



Table 14–7 Script Execution Errors

Error Message	Workaround Steps
ScriptException: Error occurred while performing deploy: The action you performed timed out after 600,000 milliseconds	<p>This error occurs when you are deploying the JVM Diagnostics Agent. To resolve this issue check if the lock for the WebLogic Administration Console on which the JVM Diagnostics Manager has already been acquired. If it has been acquired, release it and run the script again.</p> <ul style="list-style-type: none"> <li>■ Login to the WebLogic Administration Console: <code>http://&lt;machine address&gt;:&lt;weblogic port&gt;/console</code>.</li> <li>■ If you see the <b>Activate Changes</b> and <b>Undo All Changes</b> buttons in the left pane, click on these buttons to clear them. If the buttons are not cleared, click <b>Undo All Changes</b> and run the script again.</li> </ul>

### 14.5.1 Deployment on 64-bit JVMs

This section lists the errors that occur when you try to deploy the JVM Diagnostics Manager on a 64-bit JVM.

Table 14–8 Deployment on 64-bit JVM Errors

Error Message	Workaround Steps
<pre>JAM Console: Only 32 bit JVM is supported. The shared lib might not be loaded on this platform &gt;&gt;java.library.path: /scratch/skbalakr/Oracle/Middleware/oms11g/lib JAM Console: loadNative Exception loading [/tmp/libJamConsole.so.1] /tmp/libJamConsole.so.1: ld.so.1: java: fatal: /tmp/libJamConsole.so.1: wrong ELF class: ELFCLASS32 (Possible cause: architecture word width mismatch) java.lang.UnsatisfiedLinkError: /tmp/libJamConsole.so.1: ld.so.1: java: fatal: /tmp/libJamConsole.so.1: wrong ELF class: ELFCLASS32 (Possible cause: architecture word width mismatch) at java.lang.ClassLoader\$NativeLibrary.load(Native Method) at java.lang.ClassLoader.loadLibrary0(ClassLoader? .java:1778) at java.lang.ClassLoader.loadLibrary(ClassLoader? .java:1674) at java.lang.Runtime.load0(Runtime.java:770) at java.lang.System.load(System.java:1003) at oracle.sysman.e2e.model.ad4j.remote.Jam.init(Jam .java:597) at oracle.sysman.e2e.model.ad4j.remote.servlet.AD4JManagerServlet.init(AD4JManagerServlet? .java:38)</pre>	<p>This error occurs as the JVM Diagnostics Manager can be deployed only on 32-bit JVMs.</p> <p>To resolve this issue, open the <code>DOMAIN_HOME&gt;/bin/startWeblogic.sh</code> (cmd) file and change value of <code>JAVA_VM</code> variable from <code>-d64</code> to <code>-d32</code>.</p>

## 14.5.2 JVM Diagnostics Manager Automated Deployment

This section lists the errors that occur during the automated deployment process.

**Table 14–9 JVM Diagnostics Manager Automated Deployment Errors**

Error Message	Workaround Steps
<pre>Failed to deploy the application with status failed. Current Status of your Deployment: Deployment command type: deploy Deployment State: failed Deployment Message: weblogic.management.ManagementException: [Deployer:149007] New source location, /scratch/skbalakr/Oracle/Middleware/oms11g/ad4j/ jammanager_dummy.ear,cannot be deployed to configured application,jammanager. The application source is at /scratch/skbalakr/Oracle/Middleware/oms11g/ad4j/ jammanager.ear. Changing the source location is not allowed for a previously attempted deployment. Try deploying without specifying the source. No stack trace available. This Exception occurred at Tue May 11 06:40:50 UTC 2010. weblogic.management.scripting.ScriptException: Error occured while performing deploy : Deployment Failed.</pre>	<p>Automated deployment fails when you try to change the source location for a previously attempted deployment.</p>
<pre>JAM Console: Only 32 bit JVM is supported. The shared lib might not be loaded on this platform java.library.path: /scratch/skbalakr/Oracle/Middleware/oms11g/lib JAM Console: loadNative Exception loading [/tmp/libJamConsole.so.1] /tmp/libJamConsole.so.1: ld.so.1: java: fatal: /tmp/ libJamConsole.so.1: wrong ELF class: ELFCLASS32 (Possible cause: architecture word width mismatch) java.lang.UnsatisfiedLinkError: /tmp/libJamConsole.so.1: ld.so.1: java: fatal: /tmp/libJamConsole.so.1: wrong ELF class: ELFCLASS32 (Possible cause: architecture word width mismatch) at java.lang.ClassLoader\$NativeLibrary.load(Native Method) at java.lang.ClassLoader.loadLibrary0(ClassLoader? .java:1778) at java.lang.ClassLoader.loadLibrary(ClassLoader? .java:1674) at java.lang.Runtime.load0(Runtime.java:770)</pre>	<p>This error occurs if you try to run the deployment script on a 64-bit JVM.</p> <p>To resolve this issue, open the <code>DOMAIN_HOME&gt;/bin/startWeblogic.sh</code> (cmd) file and change value of <code>JAVA_VM</code> variable from <code>-d64</code> to <code>-d32</code>.</p>

## 14.6 LoadHeap Errors

This section lists loadheap errors.

**Table 14–10 LoadHeap Errors**

Error Message	Workaround Steps
glibc detected * free(): invalid next size (fast): 0x0965d090" ./loadheap.sh: line 237: 32357 Aborted ./bin/\${bindir}/processlog in=\${infile} hdr=\${sumdata} obj=\${objdata} rel=\${reldata} root=\${rootdata} osum=\${objsumdata} rrel=\${rootrel} heap=\${heap_id} skip=\${skipgarbage} db=\${dbtype} \$* Error processing file /tmp/heapdump6.txt	Check if the heapdump operation has been successfully completed. Open the heapdump6.txt file and check if there is a heapdump finished string at the end of the file. If you see this string, load the finished dump file.
Heapdump already in progress, cannot take another heapdump	Check if the heapdump operation has been successfully completed. Open the heapdump6.txt file and check if there is a heapdump finished string at the end of the file. If you see this string, restart the jamagent and run the heapdump operation again.
loadheap.sh created unusable unique indexes.	Run the loadheap/sql/cleanup.sql shipped with loadheap.zip to fix the unique indexes.

## 14.7 Errors on JVM Diagnostics UI Pages

This section lists the user interface errors.

**Table 14–11 JVM Diagnostics UI Page Errors**

Error Message	Workaround Steps
JAM Console: Socket timed out after recv -- client adc2100083.us.oracle.com:7001 is not Active [0] secs JAM Console jamlooptimeout = [3]JAM CONSOLE: JVM 1 is not active JAM Cons Err Processing Request: 128 JVM 1 is not active jamDAL: jamreq returned 128 return status < 0 from jamDalInst.processRequest	To resolve this error, increase the Agent Request Timeout (secs) and Agent Loop Request Timeout (secs)
This page cannot be displayed at the current JVM optimization level.	When the optimization level is set to 0, the Show Heap (Memory) operation is very time consuming and is not supported.
Method locals on Real Time Thread page not displayed.	Method locals is not supported if optimization level is set to 0.
Agent is up and running but is not displayed in the real time pages.	If the log file shows JAMMANAGER: OLD AGENT or NULL POOL or wrong jamoptimization level, this indicates that the old jamagent/Dbagent is being used.  To resolve this issue, download the latest jamagent from <b>Setup &gt; Download</b> page.
You do not have the necessary privileges to view this page	Ensure that you have the required JVM Diagnostics Administrator or User privileges to view the JVM Diagnostics data.

## 14.8 Frequently Asked Questions

This section lists some of the questions you may have while using JVM Diagnostics. It includes the following:

- Location of the JVM Diagnostics Logs

- JVM Diagnostics Manager Status
- JVM Diagnostics Agent Status
- Monitoring Status
- Creating Less Privileged Users
- Usage of Try Changing Threads Parameter
- Significance of Optimization Levels
- Manually Deploying the JVM Diagnostics Agent
- Log Manager Level
- Repository Space Requirements

### 14.8.1 Location of the JVM Diagnostics Logs

You can find the JVM Diagnostics logs in the following locations:

- The JVM Diagnostics Manager Log file is located at `$T_WORK/gc_inst/user_projects/domains/GCDomain/servers/EMAD4JMANAGER/logs/EMAD4JMANAGER.out`
- UI related errors are logged in:
  - `$T_WORK/gc_inst/user_projects/domains/GCDomain/servers/EMGC_OMS1/logs/EMGC_OMS1.out`
  - `$T_WORK/gc_inst/user_projects/domains/GCDomain/servers/EMGC_OMS1/logs/EMGC_OMS1.log`
- Communication errors between the JVM Diagnostics Manager and the Console are logged in `$T_WORK/gc_inst/em/EMGC_OMS1/sysman/log/emoms.log`

### 14.8.2 JVM Diagnostics Manager Status

To check the status of the JVM Diagnostics Manager:

- Click **Middleware > JVM Diagnostics > Setup**. In the Console Setup page, check if the status of the JVM Diagnostics Manager is **Up** or **Down**.
- Check the status of all Managers at **Middleware > JVM Diagnostics > Setup > JVMs & Managers** page.
- Check the JVM Diagnostics Agent log file to verify the connection between Agent and the Manager. If you see an error - `JAM Agent ERROR: Cannot connect to Console:Connection refused`, this indicates that the JVM Diagnostics Manager is not running.
- Check if the message `JAM Console: Agent connection from:[Hostname]` is present in the JVM Diagnostics Manager log file. If this message appears, it indicates that the JVM Diagnostics Manager is running and is connected to the Agent.

### 14.8.3 JVM Diagnostics Agent Status

To check the status of the JVM Diagnostics Agent:

- Click **Middleware > JVM Diagnostics > Threads > Real Time Analysis** to view the pool to which the JVM belongs. Check the JVM Status in the Connected JVMs table.
  - If the status is **Not Active**, this indicates that the Agent is not connected to the Manager. Check the agent logs to verify if it is running and the IP address and port number of the Manager is correct.
  - If the status is **No AD4J Agent Deployed**, the JVM Diagnostics Agent must be deployed on that JVM.
- Navigate to the **Threads > Real Analysis** page for the JVM. If the JVM Diagnostics Agent is running, the active threads data must be visible. If the JVM Diagnostics Agent is not running, you will see a message - *JVM is inactive, Please try again after some time.*

#### 14.8.4 Monitoring Status

To verify if the JVM Diagnostics Manager is monitoring the data:

1. Click **Middleware > JVM Diagnostics > Setup**. In the Console Setup page, verify that the **Enable Monitoring** checkbox is checked.
2. Navigate to the Monitoring page under Setup and check if monitoring status is **On** for the Pool to which the JVM being monitored belongs.
3. Navigate to the JVM Pools page under Setup and verify if the **Poll Enabled** checkbox has been checked for the Pool to which the JVM being monitored belongs. Monitoring should now be enabled.

#### 14.8.5 Running the `create_jvm_diagnostic_db_user.sh` Script

You can run the `create_jvm_diagnostic_db_user.sh` script if you want to create less privileged users who can only load heaps using the `loadHeap` script.

#### 14.8.6 Usage of the Try Changing Threads Parameter

This parameter should be used only when the JVM is highly active.

#### 14.8.7 Significance of Optimization Levels

The JVM Diagnostics Agent supports three optimization levels:

- Level 0 indicates that the JVM Diagnostics Agent is using a JVMTI based engine. This level is supported for JDK 6 series on almost all supported platforms.
- Level 1 is a hybrid between level 0 and level 2. It is supported only for very few JDKs on selected platforms.
- Level 2 uses Runtime object analysis technique for monitoring. This implementation is usually complex and time consuming and is supported for selective JDKs and platforms.

#### 14.8.8 Manually Deploying the JVM Diagnostics Agent

To deploy the JVM Diagnostics Agent manually, follow these steps:

1. Create a JVM Pool with the same name as that of the WebLogic Server Domain target but replace each '/' (slash) with '\_\_\_' (two underscores). For example if the

WebLogic domain name is /sample\_EMGC\_DOMAIN/EMGC\_DOMAIN, the JVM Pool should be \_\_sample\_EMGC\_DOMAIN\_\_EMGCDOMAIN.

2. Check the **Poll Enabled** checkbox to start monitoring this JVM Pool.
3. Download the `jamagent.war` from the **JVM Diagnostics > Setup > Download** Page.
4. In the `WEB-INF/web.xml` file of the `jamagent.war`, update the value of the following properties and create the `jamagent.war` again.
  - `oracle.ad4j.groupidprop` – The name of WebLogic Server Domain target on which the Agent is to be deployed. This is shown on JVM Diagnostics Deployment page under the **Target Name** column.
  - `oracle.ad4j.jvmidprop` - The type of WebLogic Server Domain target on which the Agent is to be deployed. This is shown on JVM Diagnostics Deployment page under the **Target Type** column with the value `weblogic_j2eeserver`.
  - `jampool` – The name of the domain on which the Agent is being deployed. The "/" replaced by "\_\_" (forward slash is replaced by double underscore)
5. Deploy rebundled `jamagent.war` on the WebLogic Server Domain manually.
6. Navigate to **Middleware** page, select the WebLogic Server Domain target. The WebLogic Server Domain Home page is displayed. Select the JVM Diagnostics option from the menu

### 14.8.9 Log Manager Level

The default log manager level is 3. You can increase it a higher level only if you are encountering some issues.

### 14.8.10 Repository Space Requirements

For monitoring data, Oracle recommends 100 MB per JVM per day with the default setting of a 24 hour purge interval. This amount can vary based upon runtime factors (e.g depth of call stacks, etc) within your environment. Hence, you must check the tablespace growth periodically and if required, you may need to change the space requirements. This will ensure that database growth due to standard monitoring will occur smoothly without sudden spikes. Tablespace sizing can be affected by the following:

- **Heap Dumps:** Analyzing heaps requires a large amount of tablespace. As a standard practice, we recommend that you must have 5 times the size of heap dump file being loaded in your 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.
- **Thread Traces:** While these are smaller than heaps. they are loaded into the database automatically when a user initiates a trace at the console. The size of these threads can vary dramatically depending on the number of active threads during the trace, the duration of the trace, and the sample interval of the trace. This should usually be under 100MB but if several thread traces have been initiated, it could fill up the database quickly. Before initiating the traces, you must ensure that there is adequate space in the database.

# Part V

---

## Using Application Dependency and Performance

The chapters and appendixes in this part provide information regarding the usage of Application Dependency and Performance (ADP).

Part II contains the following chapters:

- [Chapter 15, "Introduction to Application Dependency and Performance"](#)
- [Chapter 16, "Exploring Application Dependency and Performance"](#)
- [Chapter 17, "Performance Analytics"](#)
- [Chapter 18, "Exporting Data"](#)
- [Chapter 19, "ADP Methodology"](#)
- [Section 20, "Frequently Asked Questions About ADP"](#)
- [Appendix A, "Configuration Directories and Files"](#)
- [Appendix B, "Support Matrix for Application Dependency and Performance"](#)





---

---

# Introduction to Application Dependency and Performance

The Application Dependency and Performance (ADP) pages within Enterprise Manager Grid Control analyze Java EE, SOA, and Portal applications to capture the complex relationships among various application building blocks in its Application Schema model - the core of the Oracle intelligent platform.

This chapter includes the following:

- [Section 15.1, "Overview"](#)
- [Section 15.2, "Architecture"](#)

## 15.1 Overview

Using the insights stored in Application Schema, ADP is able to deliver an Application Service Management (ASM) environment that self-customizes out-of-the-box, evolves with change, minimizes expert involvement, and delivers a holistic, service-oriented view across heterogeneous environments. ADP enables an enterprise to more efficiently manage distributed applications, attain management agility, and lower total cost of ownership.

See the following sections:

- [Section 15.1.1, "Managing Complex Java EE, SOA, and Portal Applications"](#)
- [Section 15.1.2, "Delivering a Service-Oriented View Across Environments"](#)
- [Section 15.1.3, "Avoiding Involvement from Java EE, SOA, Portal, and Application Experts"](#)
- [Section 15.1.4, "Eliminating Repetitive Do-It-Yourself \(DIY\) Manual Processes"](#)
- [Section 15.1.5, "ADP Solution"](#)

### 15.1.1 Managing Complex Java EE, SOA, and Portal Applications

Today's Java EE, SOA, and Portal applications enable enterprises to deliver mission-critical business functions to key constituencies - most often their customers, partners, and employees. These composite applications are assembled from many different Java EE components and exposed services distributed across a heterogeneous environment.

To be effective at managing today's complex, distributed Java EE, SOA, and Portal applications across a heterogeneous environment, enterprises must adopt an intelligent ASM platform with the following characteristics:

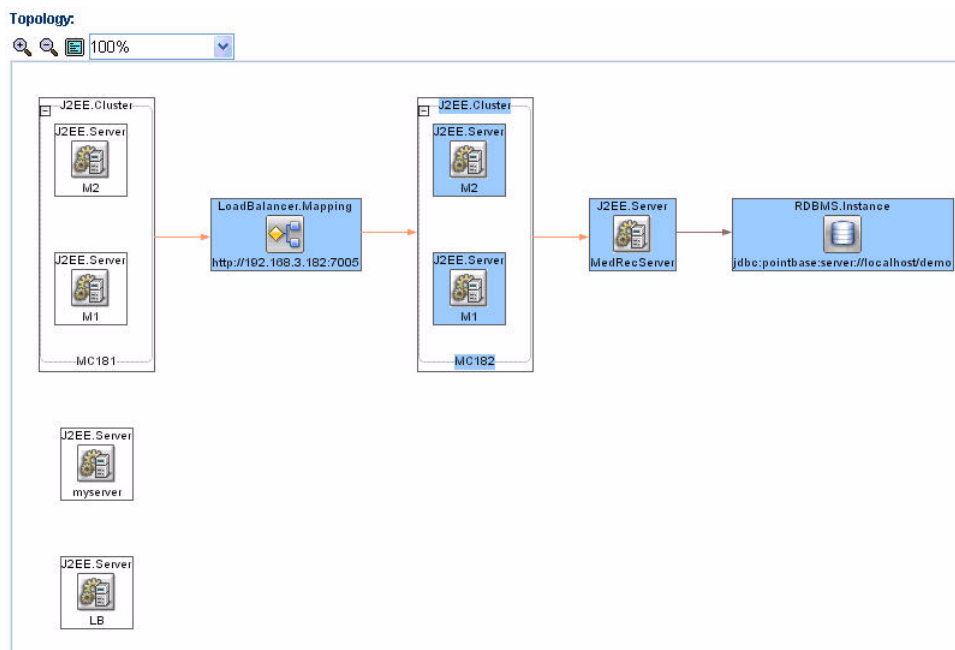
- Provides holistic, service-oriented views across heterogeneous environments  
An intelligent ASM platform must provide high-level service-oriented metrics that map to low-level technology-centric metrics. These measurements must be organized in a service-oriented fashion to deliver a unified, holistic view of the numerous interconnected application components deployed across heterogeneous environments.
- Requires minimal Java EE, SOA, Portal, and application expertise  
An intelligent ASM platform must have the ability to capture complex relationships among various interconnected components of today's Java EE, SOA, and Portal applications. This ability can help minimize reliance on Java EE, SOA, Portal, and application experts for setting up and maintaining effective APM environments.
- Eliminates repetitive DIY manual processes  
An intelligent ASM platform must eliminate repetitive DIY manual processes by delivering the ability to self-customize out-of-the-box and evolve with change. Elimination of these repetitive DIY manual processes is the only way to deal with rising complexity and rapid rate of change with ease.

### 15.1.2 Delivering a Service-Oriented View Across Environments

Today's mission-critical business functions are powered by Java EE, SOA, and Portal applications that comprise numerous interconnected components deployed across highly distributed environments. To manage these applications effectively, enterprises must first gain an understanding of the complex relationships among the business functions, associated interconnected components, and the underlying runtime environments. To enable clear and accurate understanding, IT organizations need holistic, service-oriented views that span across heterogeneous environments.

Furthermore, appropriate rendering of these views enables users at different levels of the organization to collaborate with each other and do their respective jobs more efficiently.

**Figure 15–1 Application Dependency and Performance Topology View in Enterprise Manager Grid Control**



Application Schema Navigation provides efficient ways for you to access relevant information using techniques like hierarchical traversal, architecture model navigation, string queries, drill down, drill out and more.

### 15.1.3 Avoiding Involvement from Java EE, SOA, Portal, and Application Experts

To manage Java EE, SOA, and Portal performance effectively, IT organizations must adopt an intelligent platform like ADP that requires minimal expertise to set up and maintain. Unlike conventional APM toolkits, ADP does not rely on human expertise to set up and maintain customized APM environments. Instead, ADP uses a unique model-driven approach that leverages the information stored in its Application Schema model to keep the involvement of experts to the minimum. ADP's unique ability to self-customize out-of-the-box and evolve with change makes it the perfect solution for managing not only custom enterprise applications, but also applications developed by external parties.

### 15.1.4 Eliminating Repetitive Do-It-Yourself (DIY) Manual Processes

Based on a unique model-driven approach, ADP eliminates repetitive DIY manual processes. To achieve this level of self-customization and continuous change adoption, ADP uses its AppsSchema modeling technology to perform the critical task of analyzing application structure and infrastructure configuration. After capturing these insights in the Application Schema model, ADP leverages this information to establish a fully customized ASM environment. To keep this environment up-to-date, ADP continuously updates the Application Schema model as new applications are deployed and changes are applied. ADP's unique ability to self-customize out-of-the-box and evolve with change enables fast time-to-value, low total-cost-of-ownership (TCO), and maximal return-on-investment (ROI).

### 15.1.5 ADP Solution

Oracle provides the industry's first intelligent ASM platform for Java EE, SOA, and Portal. Unlike conventional APM toolkits, ADP analyzes these applications and captures complex relationships among various application building blocks in its Application Schema model - the brain of this intelligent ASM platform.

Using the insights stored in the Application Schema model, ADP is able to deliver an ASM solution that self-customizes out-of-the-box, evolves with change, minimizes expert involvement, and delivers a holistic, service-oriented view across heterogeneous environments. Adopting an intelligent platform such as Oracle will enable the enterprise to more efficiently manage distributed applications, attain management agility, and lower total cost of ownership.

## 15.2 Architecture

ADP employs a multi-tier, fully distributed, configurable architecture to provide the scalability and flexibility to meet the changing needs of enterprise deployments.

ADP operates as a service on the machine and automatically begins running when the machine first boots, and remains on perpetually. ADP is typically installed on its own machine and dedicated to monitor a group of managed application servers.

To allow remote access to ADP through a browser, a web container is installed. This web container provisions the ADP UI applets to the browser and maintains communication with these applets. See [Figure 15-2](#).

---

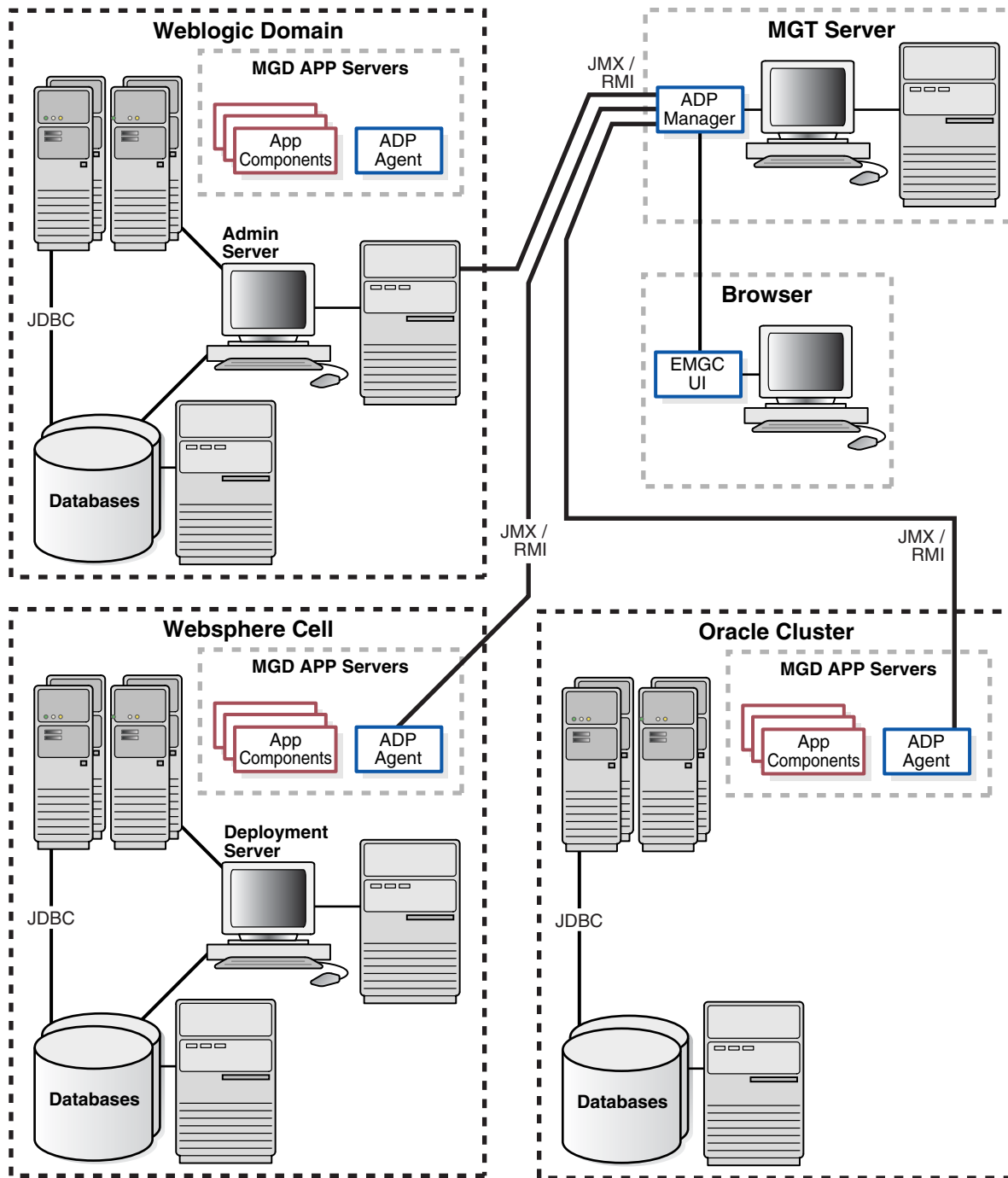
---

**Note:** By default, the browser functionality is turned off. To turn on the browser functionality, set the Tomcat properties to *true* in the ADP\_HOME/config/Acsra.properties file.

---

---

Figure 15–2 ADP Topology



The following core components are deployed to form the ADP ASM system.

### 15.2.1 ADP Java Agents

ADP Java Agents are the data collectors of the ADP ASM system. ADP Java Agents are deployed to all managed application servers to perform a series of tasks including collecting performance managements, tracking contextual relationships, and summarizing data in real-time while introducing as little overhead as possible. At the expiration of the predefined aggregation interval, these agents forward the summarized data to ADP for additional analysis. For various Java EE platforms such

as Oracle SOA Suite, Oracle WebLogic, and IBM WebSphere, ADP leverages their deployment infrastructures to quickly deploy the ADP Java Agents to all application servers.

## 15.2.2 ADP Manager

ADP Manager is the core analytical engine of the ADP ASM system. In real-time, ADP Manager performs complex mathematical modeling and statistical calculations with summarized data from all ADP Java Agents. ADP Manager can be configured with a backup to provide higher level of availability.

ADP Manager can also be configured without the UI component, also known as headless configuration.

### 15.2.2.1 ADP Manager and High Availability

Although the ADP Manager does not have high availability (HA) built into it, administrators can have a backup ADP Manager installed on a separate machine; this backup ADP Manager points to the same database but is disabled. If the production ADP Manager fails, the backup ADP Manager can then be enabled against the same database. The backup ADP Manager then rediscovers the application after the agent is redeployed from the backup ADP Manager, to the managed resources in order to synchronize them. All metrics are preserved, assuming the model does not change in the short timeframe it takes to bring the backup ADP Manager online.

The key with this backup procedure is to backup the database that ADP uses as its repository in order to preserve the historical data. On the modeling side, the backup ADP Manager has to rediscover the application which should happen automatically, as long as the resources are configured and the new agent has been deployed.

If historical data preservation is not a necessity, users can simply have another ADP Manager and database and swap agents reducing the backup effort considerably.

## 15.2.3 ADP Database

ADP stores its analyzed data and application models in the Management Repository. There is a separate schema for ADP in the Management Repository and this schema is external to the ADP installation.

## 15.2.4 ADP User Interface

The ADP User Interface (ADP UI) is the primary user interface for ADP users. Users can use ADP UI to set Service Level Objectives (SLOs), define actions, analyze monitoring data, and more. The ADP UI is fully configurable. To access ADP:

1. From the Grid Control Home page, click **Targets**, then click **Middleware**.
2. In the Related Links section of the Middleware page, select **Application Dependency and Performance**.

---

---

## Exploring Application Dependency and Performance

This chapter examines the following:

- [Section 16.1, "Exploring the User Interface"](#)
- [Section 16.2, "Exploring the Monitoring Tab"](#)
- [Section 16.3, "Exploring the Configuration Tab"](#)
- [Section 16.4, "Exploring the Registration Tab"](#)

### 16.1 Exploring the User Interface

This section explores the ADP User Interface. Topics include:

- [Section 16.1.1, "Accessing ADP"](#)
- [Section 16.1.2, "General ADP UI Elements"](#)
- [Section 16.1.3, "Drill Down in Operational Dashboard"](#)
- [Section 16.1.4, "Configuring Service Level Objectives \(SLOs\)"](#)
- [Section 16.1.5, "Configuring SLO Blackouts"](#)
- [Section 16.1.6, "Time Frame"](#)
- [Section 16.1.7, "Display Interval"](#)
- [Section 16.1.8, "Graphs and Data Items"](#)
- [Section 16.1.9, "Right-Click Operations on Tables and Graphs"](#)
- [Section 16.1.10, "Comparative View"](#)
- [Section 16.1.11, "Save as PDF"](#)
- [Section 16.1.12, "Easy Scroller"](#)
- [Section 16.1.13, "Zoom In and Zoom Out Toolbar"](#)
- [Section 16.1.14, "Custom Metrics"](#)
- [Section 16.1.15, "Functional View"](#)
- [Section 16.1.16, "Topology View"](#)
- [Section 16.1.17, "Architecture View"](#)
- [Section 16.1.18, "Metric Types"](#)

## 16.1.1 Accessing ADP

To access Enterprise Manager's Application Dependency and Performance (ADP) feature, do the following:

1. From the Grid Control Home page, click the **Targets** tab.
2. On the Targets page, click the **Middleware** tab.
3. On the Middleware page, either:
  - Click the **Application Dependency and Performance** link which is located in the Related Links section at the bottom of the page, or
  - Click the target in which you are interested. In the tree view, select a target under Application Deployments. In the right pane, expand the **Application Deployment** menu, select **Monitoring**, and then select **Application Dependency and Performance**.

## 16.1.2 General ADP UI Elements

ADP UI consists of the following core components:

- Navigation Pane (left)

There are three types of workspaces in the ADP navigation pane: *Monitoring*, *Configuration*, and *Registration*. In the Monitoring workspace, you can navigate the managed environment and monitored applications in a tree.

  - Use the Monitoring workspace to traverse the ADP tree model and identify abnormal activities.
  - Use the Configuration workspace to create, modify, and review various configuration settings for ADP.
  - Use the Registration workspace to register a Remote Method Invocation (RMI) manager.
- Main Display Window (right)

As you navigate through the ADP tree model and configuration categories, detailed performance information and configuration settings are displayed in the Main Display Window. You can refresh the Main Display Window at anytime by clicking the Refresh icon.

## 16.1.3 Drill Down in Operational Dashboard

The Operational Dashboard displays the health indicators for various key entities in the managed environment. ADP uses traditional traffic light colors to represent the health of these various key entities.

For each component, ADP uses the following health indicators to provide a comprehensive view. These health indicators are:

- Performance

The performance health indicator depicts the relative responsiveness of the monitored entity to the configured threshold.
- Availability

The availability health indicator informs you to what extent a particular entity is available to service requests.
- Errors



The errors health indicator informs you if the number of errors and exceptions encountered by this entity are approaching or violating the configured threshold.

- **Load**

The load health indicator depicts how many operations have been performed and requests have been served by a particular entity.

ADP is aware of clusters. As such, these indicators display overall health of a particular entity across the entire cluster.

## 16.1.4 Configuring Service Level Objectives (SLOs)

In ADP, thresholds configured for various measurements are called Service Level Objectives (SLOs). Configuring SLOs is a key activity for establishing and maintaining an effective performance monitoring system. It is easy to configure SLOs in ADP. Click the **Configuration** tab and select the **Service Level Objective Configuration** option.

### 16.1.4.1 Create New SLO

When you select Service Level Objectives Configuration, ADP displays the Service Level Objective Configuration window. This window allows you to apply existing SLOs or create new ones. When you click **Create New SLO**, ADP guides you through the process of setting up a new SLO.

The steps for SLO creation are as follow:

1. Either select a SLO file or create a new SLO file. ADP can store SLO configurations in different files to improve configuration portability.
2. Define the SLO Entity Type. ADP automatically selects the appropriate entity type for you based on the selected monitoring element. For example, if you want to set a SLO on a Portal Desktop element, ADP automatically sets the Entity Type for you.
3. Other information is filled in by default. Normally, there is no need to modify the SLO Entity values.
4. When you are done setting the SLO Entity Type values, click **Create New SLO** to go to the second step of the SLO creation process, Defining the SLO Parameters. Note: The (\*) character means Select All. It is recommended that you do not use the (\*) character.

### 16.1.4.2 Defining SLO Parameters

Follow these steps to define the SLO parameters:

1. Type in the name of the SLO parameter.
2. Select the performance metric.
3. Define the monitoring window size, which determines how long the condition must persist before generating an alert.
4. Set threshold values for the SLO.
5. Select what actions to take when a trigger is fired. A list of preconfigured actions is available in the view pane.
6. Add new actions by going to the Action Configuration node in the Configuration Workspace.
7. Click **Save** to set the SLO for this monitored element.

8. You can delete unwanted SLOs for any element from this window.

**16.1.4.2.1 Propagating Threshold Violation Events** ADP is designed to propagate threshold violation events up the hierarchy. Therefore, when a SLO is set on a lower level metric, the higher level health indicator light becomes activated. Additionally, the health indicator light for the application server that hosts this component also becomes active. Oracle calls this *containment approach* to SLO event propagation. When a lower level SLO is violated, the violation event propagates all the way up the hierarchy and changes the status of all containers for this event.

For example, say we defined a performance threshold on the average response time metric of the CaseManagement portlet. We would expect the css portal and cgServer health indicator lights to become active because the CaseManagement portlet is part of the css portal, css portal is part of the cssdemo application, and cssdemo application is deployed on cgServer.

**16.1.4.2.2 Types of SLOs** In addition to the containment concept, ADP categorizes SLOs into the following types:

- Performance
- Availability
- Error
- Load

In the example, the average response time metric is correctly categorized into the performance type.

If you set a SLO on a metric in the load category such as Portal Desktop Visit Count, you will see the activation of load health indicators for all containers of the desktop. In our example, we set a SLO on the Portal Desktop Visit Count of the csr desktop. This activates the load health indicators for css portal, cssdemo application, and cgServer instance.

**16.1.4.2.3 SLO Events Viewer** Right-click on any tree node and select **View Service Level Objective Events** to open a new window. You can see all the SLO violation events triggered for the selected entity. ADP automatically applies a filter to show only relevant events.

Once new SLOs are added, ADP updates the relevant graphs to visually display these new thresholds. [Table 16–1, "SLO Line Types"](#) explains the different line types.

**Table 16–1 SLO Line Types**

Line Description	Description
Solid Red Line	A violation threshold that triggers on high.
Solid Yellow Line	A cautionary threshold that triggers on high.
Dashed Red Line	A violation threshold that triggers on low.
Dashed Yellow Line	A cautionary threshold that triggers on low.

## 16.1.5 Configuring SLO Blackouts

Use this option to specify blackout time frames to prevent having a specified number of SLOs from being evaluated. You can prevent having unwanted alerts being fired during planned or unplanned down time.

1. From the Grid Control home page, click the **Targets** tab. On the targets page, click the **Middleware** tab. On the Middleware page, locate the Related Links section and click the **Application Dependency and Performance** link. Ensure the Configuration tab is highlighted
2. Select **SLO Blackout Configuration**.
3. You can view any existing SLO blackout.
4. Use this window to create, delete, or view the details of existing blackouts.

### Deleting SLO Blackout

1. Select an existing event on the list.
2. Click **Delete SLO Blackout**.
3. Confirm that you want to delete the entry and click **Yes**.

### Viewing SLO Blackout Summary List

1. Click **SLO Blackout Summary List**.
2. View the details on the existing SLO Blackout events.
3. Click **Show SLO Blackout List** to return to the previous window.

### Creating SLO Blackout

1. Click **Create SLO Blackout** to view the detail window.
2. On the SLO Blackout File page, type the name of the blackout in the New SLO Blackout File field. Click **Continue**.
3. On the SLO Blackout Configuration page, fill in the fields. Refer to [Table 16–2, "SLO Blackout Configuration"](#) for details.

**Table 16–2 SLO Blackout Configuration**

Column/Metric	Description
Blackout Name	Type in the name.
Description	Type in the description of the SLO you are creating.
Blackout By SLO File	Use to blackout at the file level. The SLO files display in a list where you can select them or cancel out of the window.  This option restricts the blackout to the SLO file name.
Blackout By SLOs	Use to blackout at the SLO level. The SLOs display in a list where you can select them or cancel out of the window.  This option restricts the blackout to the SLO name.
Blackout By Entity	Use to blackout at the entity type level. Click the <b>Blockout by Entity:</b> button to view the list of entity types. Select the entity.  This option restricts the blackout to the entity type selected.
year, month, date, hour, minute, duration	Use the guidelines to the right of these columns to enter the appropriate information.
recurring	Select how often you would like to run this blackout event from the list.

## 16.1.6 Time Frame

In ADP, you can specify the length of the time the window information is to be displayed. To specify the length of this time window, select the appropriate length in the Time Frame list. The following Time Frame values are available:

- 1 hour
- 2 hours
- 4 hours
- 8 hours
- 12 hours
- 24 hours

---

---

**Note:** The ADP default data collection interval is 60 seconds. As you adjust the data collection interval, ADP automatically adjusts the display time frames.

---

---

ADP automatically adjusts information displayed to fit the specified time window. You can drill down to see detailed performance information for a specific range of time.

For example, visualize the drill down process with two screen shots of the same graph with different Time Frames of the average response time for Portal campaign. The first graph has a Time Frame of 24 hours. The second graph has a Time Frame of 1 hour. By increasing the granularity of the Time Frame, you are performing a drill down operation.

For example, an IT Operations staff noticed abnormally high response time with Portal campaign subsystem. The person decided to investigate further to evaluate the extent of the problem. By changing the Time Frame from 24 hours to 1 hour, this user is able to see that between 14:17 and 14:18, the Portal campaign response time jumped from an average of 1000 milliseconds to 5000 milliseconds. While the problem did not persist, it may warrant additional investigation.

## 16.1.7 Display Interval

Display Interval, located above the Main Display window, indicates the start and end time for the data displayed in the Main Display Window. Display Intervals change as you change the following settings:

- Time Frame
- Interval Context
- Turning Off Time Frame Limitation

### 16.1.7.1 Time Frame

When you select a new Time Frame, the Display Interval automatically changes to fit the selected Time Frame. For example, if you were to change the Time Frame from 1 hour to 2 hours, the Start value of the Display Interval changes.

### 16.1.7.2 Interval Context

Display Interval can also be changed by setting the Interval Context. The settings for the Interval Context are:

- End Time Is Current System Time

The default Interval Context for ADP is to use the current system time as End value for the Display Interval. In this default setting, you have a sliding Display Interval and can see the latest performance information in the Main Display Window.

- End Time Is Fixed

You can also change the Interval Context setting to use a fixed time as the End value for the Display Interval. By selecting the fixed Interval Context, you can create a fixed time window to display performance data. The fixed time window is particularly useful for performing analytical tasks.

- Date/Time Selector

When you select to fix the End time for the Interval Context, the ADP UI enables a pair of Date/Time Selectors to allow you to set Start or End values for the Display Interval. Click the icon next to the Start and End times to open up the Date/Time Selector.

The Date/Time Selector allows you to set a specific Display Interval to fit your needs. Additionally, the Date/Time Selector enables ADP to compare current performance trends with historical data.

---

**Note:** Changing the start and end time do conceptually different things. Users are advised to always change their time frame by modifying the end time first, and then the start time. Changing the end time moves the window in time, whereas changing the start time increases/decreases the size of the window.

---

### 16.1.7.3 Turning Off Time Frame Limitation

To support the display of data for more than twenty four hours, ADP allows you to specify your own time frame for data display. To enable this feature, set **Interval Context** to **End time is fixed** and make sure the **Use time frame:** check box is unchecked. Turning off time frame limitation allows ADP to display eight days worth of data.

For example, when you specify the time frame to be eight days by adjusting the start and end times through the Date/Time Selector, ADP then adjusts its view to display eight days worth of data in a single graph. This feature allows you to perform trending analysis over time.

## 16.1.8 Graphs and Data Items

ADP displays performance information in various formats. Most commonly used display formats in ADP are tables and graphs.

- On graphs, you can gain more information about a data item by pointing the mouse over the interested item.
- Minimum and maximum response time measurements are stored in their database in addition to average response time measurements. The min and max metrics, if present, are displayed visually in the UI.
- For tables, you can perform a table sort by clicking the blue up/down arrow located in the column headings.

- On some screens, you can define the zoom in area using a click and drag operation. To zoom in, click and drag the mouse to the right. To zoom out, click and drag the mouse to the left.

**Tip:** For graphs with extreme outliers, graph details are lost due to automatic graph scaling. To work around this problem, you can use the graph zoom in feature to review these details. This feature is only available on some screens.

### 16.1.9 Right-Click Operations on Tables and Graphs

There are several simple operations you can perform on various tables and graphs in ADP. The following is a list of the right-click operations associated with tables and graphs.

Note that these operations are only available in Applet views.

**Table 16–3** *Right-click on Tables and Graphs*

Right-Click Operation	Description
Copy cell value	The right-click operation is available for tables only. This operation copies the cell value to enable common copy/paste operation (table only).
Export as CSV...	This right-click operation is available for both tables and graphs. This operation saves all the values in the table or graph as a comma separated value (CSV) file. The CSV file can later be imported into other applications such as Microsoft Excel.
Count number of rows	This right-click operation is available for tables only. This operation returns a count for the number of rows in the selected table (table only).
Fit to View	Make the table or graph file the entire pane.
Restore View	Restore the layout to the default state
Show/Hide Table Columns	Remove or un-remove columns from a table (table only)
Enable/Disable Series	Remove or un-remove a line from a graph (graphs only)

For example, you can use the Count number of rows right-click operation to get a total row count for any table.

### 16.1.10 Comparative View

ADP provides a number of analytical tools to enable performance analysis. One of these tools is the Comparative View. To access Comparative View, right-click the ADP Main Display Window and select **Create Comparative View**.

Note that this tool is only available in Applet views.

After the Comparative View window appears, you can use the Date / Time Selector to specify start and end times for each of the two windows in the Comparative View. You can use this tool to compare performance statistics of two different time frames.

**Tip:** You can use comparative views to determine if current performance of a specific application or component differs greatly from historical performance or baseline performance captured previously.

Comparative views are useful to evaluate current performance characteristics against historical performance characteristics.

### 16.1.11 Save as PDF

To improve collaboration among those who work on application performance issues, ADP provides the ability to save any view as a PDF file. To save a specific view as a PDF file, right-click on the ADP Main Display Window and select **Save this view as a PDF file**.

Note that this feature is only available in Applet views.

### 16.1.12 Easy Scroller

Easy Scroller is a feature to help you navigate different views in ADP. To bring up Easy Scroller, right-click on a view in the Main Display pane and select the **Easy Scroller** option. Drag the box within Easy Scroller to navigate.

Note that this feature is only available in Applet views.

### 16.1.13 Zoom In and Zoom Out Toolbar

For some of the views, ADP provides the zooming ability. This capability enables you to zoom into diagrams for more fine-grain details and zoom out for more coarse-grain structure.

On the Zoom In/Zoom Out Toolbar, the icon zooms in on the view by 10%, the icon zooms out on the view by 10%, and the icon returns the view back to normal size (100%). You can use the drop-down box to quickly zoom in or zoom out of the view.

Note that this feature is only available in Applet views.

### 16.1.14 Custom Metrics

While ADP intelligently selects relevant performance metrics based on its Application Schema model, you can further customize the monitoring environment by configuring additional custom metrics. In addition, you can use custom metrics in problem diagnostic situations where additional visibility is needed to pinpoint problem root cause.

To configure a new custom metric:

1. Click **Custom Metric Configuration** on the Configuration tab
2. Click the **Create Custom Metric** button.
3. On the Custom Metric File page, either choose an existing custom metric file or provide the name of a new custom metric file. Click **Continue**. ADP walks you through the configuration process.

Custom Metric Configuration page includes the following fields, see [Table 16-4, "Custom Metric Configuration Page"](#).

**Table 16-4 Custom Metric Configuration Page**

Field	Description
Name	This text field is for defining the display name for the custom metric.
Resource Name	This list is for defining the resource where the custom metric will be collected.
Class Name	This text field is for defining the fully qualified class name (package + class) associated with the custom metric.

**Table 16–4 (Cont.) Custom Metric Configuration Page**

Field	Description
Method Name	<p>This optional text field is for defining the method name associated with the custom metric.</p> <p><b>Usage:</b></p> <ol style="list-style-type: none"> <li>1. Type in * - ADP will instrument all methods.</li> <li>2. Provide comma separated list of methods with no wildcards - ADP will create method entities and only instruments these methods in the agent.</li> <li>3. Provide comma separated list of methods with wildcard prefixes or suffixes - ADP will instruct the agent to instrument the methods specified along with the wildcards.</li> <li>4. Provide 1) or 2) preceded by "!" to create an excluded list - ADP will instruct the agent to instrument all methods in the class not defined in the exclude list.</li> </ol> <p>Method field examples:</p> <ol style="list-style-type: none"> <li>1. methodA,methodB,methodC</li> <li>2. ejb*,*context,methodA</li> <li>3. !ejb*,*context,methodA</li> </ol>

After you define the custom metrics, restart the application server instances associated with these customizations. The new custom metrics will be listed under the Custom Metrics node in the ADP navigation tree.

The newly configured custom metric provides class level performance data, for example invocation count and response time.

### 16.1.15 Functional View

Functional View is a type of Application Schema Visualization - a visual way for ADP to represent the information stored in its Application Schema model. This view is designed to help you understand how business functions are assembled with various functional building blocks. [Table 16–5, "Functional View"](#) provides a list of functional views currently available in ADP.

**Table 16–5 Functional View**

Entity Type	Function View	Description
Process	Process Workflow View	This functional view depicts the workflow associated with the selected WLI and Oracle BPEL business process. It shows all the process nodes and the relationships among them.
Pageflow	Pageflow Functional View	This functional view depicts the logical flow associated with a JPS or Struts pageflow. It shows all the pages in a pageflow and the relationships among them.
OSB Proxy Service	Proxy Service Functional View	This functional view depicts the pipeline and stage flow associated with an OSB Proxy Service.

Depending on the type of entity selected, ADP displays different functional views. Right-click and select Display Functional View to bring up the relevant Functional View associated with the selected entity.

### 16.1.16 Topology View

Topology View is another type of Application Schema Visualization - a visual way for ADP to represent the information stored in its Application Schema model. This view is designed to help you understand how application environments are assembled with



various applications, application server instances, and shared resources. This information helps you map composite applications and their building blocks to application server instances and share resources.

The highest level topology view graphically depicts domains, external resources, and shared database resources. The applications used in the following examples are CSS and MedRec demos.

For example, you can have a topology with two ADP managed resources, CSS Domain and MedRec Domain, two external resources, and a shared database resource. The lines connecting various entities in the Topology Views depict calls made from one entity to another. You can get more information about a specific call by pointing the mouse over a specific line.

It is possible to hide different types of lines in the Topology View. To the line types, right-click the **Topology View** and highlight the **Edge types** option to reveal a list of different edge (arrow) types associated with the current Topology View.

### 16.1.17 Architecture View

Architecture View is another type of Application Schema Visualization; a visual way for ADP to represent the information stored in its Application Schema model. This view is designed to help you understand the structure and behaviors of Java EE, SOA, and Portal applications at the module and component level. Some Architecture Views also include built-in delay analysis to help identify potential bottlenecks in a given call path.

The Architecture View in ADP is capable of showing application structure and component relationships at two levels: module and component levels. At each level, ADP can show both active and potential call paths. [Table 16–6, "Various Types of Architecture View"](#) describes various types of Architecture Views.

Drill down on a specific application to launch into the Architecture View. This action demonstrates the logical progression of drilling from high level resource-centric topology view, down through application-centric topology view, to module-centric architecture view. Using this logical drill down, you can understand the structure of your application runtime environments and diagnose problems.

**Table 16–6** *Various Types of Architecture View*

Tab Name	Description
Module Level Execution	This is the default Architecture View at the module level. The Module Level Execution view shows the active calling relationships among various Java EE modules (EAR, WAR, JAR, and so on). Shared resources are also included.
Module Level	The Module Level view shows the potential calling relationships among various Java EE modules. Shared resources are also included. It should also be noted that any object that is not connected within the static view will not be included at this level and if there are no static connections at all between objects, every potential object relationship will be displayed. By default, the Module Level view is not enabled.
Component Level Execution	This is the default Architecture View at the component level. The Component Level Execution view shows the active calling relationships among different Java EE components (EJB, servlet, JSP, and so on). Shared resources are also included.
Component Level	The Component Level view shows the potential calling relationships among various Java EE components. Shared resources are also included. Similar to the module level, any object that is not connected within the static view will not be included at this level and if there are no static connections at all between objects, every potential object relationship will be displayed. By default, the Component Level view is not enabled.

These various types of architecture views are color coded in order to provide additional information. [Table 16–7, "Architecture View Color Codes"](#) lists color codes and their meanings.

**Table 16–7 Architecture View Color Codes**

Background Color	Description
Orange	The orange background color represents entry points into the application or module. The orange color also represents that these entities belong to the same application or module currently selected (in context).
Green	The green background color represents entry points into the application or module. The green color also represents that these entities belong to other applications or modules (out of context). The green color is also used to represent share resources.
White	The white background color represents that these entities belong to the same application and module currently selected (in context).
Blue	The blue background color represents that these entities belong to other applications and modules (out of context). The blue color also represents shared resources.

ADP graphically depicts active calling relationships among various Java EE modules and shared resources.

#### 16.1.17.1 Accessing the Architecture View

There are several ways to access the Architecture View. One way is through the Deployments node associated with a specific application under the Application Node. Application specific Architecture View can be accessed using the Deployments node on the Oracle Tree.

The last way to access the Architecture View is by right-clicking a managed entity and selecting the Architecture View. Right-click and select Architecture View to start the drill down process.

**16.1.17.1.1 Arrows in Architecture Views** The arrows connecting various entities in the Architecture Views depict calls made from one entity to another. You can get more information about a specific call by pointing the mouse over a specific arrow. Mousing over arrows shows the details of a specific call in Architecture View

It is possible to hide different types of arrows in the Architecture View. To do this, right-click on the Architecture View and highlight the Edge types option to reveal a list of different edge (arrow) types associated with current Architecture View. Unchecking a specific edge type hides all lines of that type in the Architecture View. Checking a specific edge type makes these lines appear.

To hide all lines not connected with a specific entity, select a monitored entity in the Architecture View, right-click and select **Hide other edges**. Highlight an entity and select **Hide other edges** to hide all arrows not connected to the managed entity.

**16.1.17.1.2 Architecture View Summary** The Architecture View Summary provides the delay analysis associated with the active call path displayed. The table and pie chart displayed in the right pane guides you to leading delay contributors in the displayed call path. Selecting a specific component in the call path brings up component specific information. You will see the following tabs:

- Summary tab first which includes high-level delay data for both inbound and outbound calls.

- The Instrumentation tab shows detailed method level performance data associated with the selected component. Click the Instrumentation tab to see detailed performance measurements and information at the method level.
- The Errors/Exceptions tab shows the errors metrics associated with the selected portal or BPEL process.
- The SQL Statement tab shows SQL statements and their performance data associated with the selected component.
- The Transactions tab shows the transaction events associated with the selected portal and children below. By default, the Transactions tab is not enabled.

## 16.1.18 Metric Types

Table 16–8, "Metric Types" describes various types of metrics provided by ADP.

**Table 16–8 Metric Types**

Examples	Metric Type	Metric Description
Active Sessions Completions Pending Requests Running Instances Max Capacity Messages High	Snapshot Count	A count of the monitored entity at a point in time. ADP plots these snapshot counts in trend graphs.
Requests Serviced Total Sessions [Processes] Aborted [Processes] Terminated [Method] Invocation Count Bytes Received	Aggregated Count	A count of the monitored entity incrementally aggregated from the beginning of display time window. ADP shows these aggregated counts in summary tables.
Response Time Elapse Time Connection Delay	Average Timing	<p>Calculated every sampling period (default 60 seconds), the average timing is calculated by dividing the total amount of time needed to complete the monitored business unit of work by the number of completed business units of work.</p> <p>ADP uses this data in the following two ways:</p> <ol style="list-style-type: none"> <li>1. Plot the average timings in trend graphs.</li> <li>2. Calculate average timing of this business unit of work for the display time window and display in a summary table.</li> </ol>
Min/Max	Minimum and Maximum Response Time Measurement	Minimum and maximum response time measurements found per collection sampling intervals. These are stored in their embedded database in addition to average response time measurements. The default is 60 seconds.

## 16.2 Exploring the Monitoring Tab

When ADP is pointed to a Oracle WebLogic domain, IBM WebSphere cell, or an Oracle SOA Suite cluster, it automatically discovers information about this particular domain including all deployed applications, configuration, resources, and others. ADP displays this information in the Monitoring tab under Oracle Enterprise Manager.

Each node represents a construct in the platforms monitored by ADP. Each construct is described in this section.

This section includes the following topics:

- [Section 16.2.1, "Oracle WebLogic Portals"](#)
- [Section 16.2.2, "WebSphere Portals"](#)
- [Section 16.2.3, "Oracle BPEL Processes"](#)
- [Section 16.2.4, "Oracle ESB"](#)
- [Section 16.2.5, "Oracle WebCenter"](#)
- [Section 16.2.6, "Processes"](#)
- [Section 16.2.7, "Web Services"](#)
- [Section 16.2.8, "Pageflows"](#)
- [Section 16.2.9, "Services"](#)
- [Section 16.2.10, "WSRP Producers"](#)
- [Section 16.2.11, "Integration"](#)
- [Section 16.2.12, "Applications"](#)
- [Section 16.2.13, "Oracle WebLogic Resources"](#)
- [Section 16.2.14, "WebSphere Resources"](#)
- [Section 16.2.15, "Oracle Resources"](#)
- [Section 16.2.16, "Custom Metrics"](#)
- [Section 16.2.17, "CAMM Node"](#)
- [Section 16.2.18, "Service Component Architecture \(SCA\)"](#)

## 16.2.1 Oracle WebLogic Portals

The Portals node under Oracle Enterprise Manager contains information about all deployed WebLogic Portal applications in the managed domains. The Portals node is organized hierarchically using the same framework developers use to build these Portal applications. The minimum and maximum response time measurements are stored in the database in addition to the average response time measurements. These metrics, if present, display visually in the Main Display Window on the right.

For WebLogic Portal, this hierarchy contains the following ([Table 16–9, "WebLogic Portal Hierarchy"](#)):

**Table 16–9 WebLogic Portal Hierarchy**

Component	Description
Portals	The Portal is the logical containment unit for a Portal application. A typical Portal can contain a few desktops, several books, tens of pages, and hundreds of portlets.
Desktops	The desktop is the top-level container for the portal components included in that specific view of the portal.  Portal administrators can create new desktops beyond what portal developers create in WebLogic Workshop.
Books	The top-level book contains all sub-books, pages, and portlets. The top-level book defines the initial menu navigation style used for the desktop. For each sub-book you add to a desktop you can select a different navigation style.

**Table 16–9 (Cont.) WebLogic Portal Hierarchy**

Component	Description
Pages	Pages and sub-books are the navigable containers used for organizing portlets.
Portlets	Portlets are the containers that surface Web content and applications in your desktops.

When you click the Portals node under Oracle Enterprise Manager, ADP displays summary information on active portal applications. This summary includes the following ([Table 16–10, "Tree Summary"](#)):

**Table 16–10 Tree Summary**

Metric	Description
Portal web application activity	A summary of user sessions for a specific portal application
Portal completions	Total number of requests fulfilled by a specific portal application
Portal response time (ms)	Average response time for a specific portal application
Portal entitlement response time	Average response time of WebLogic Portal entitlement subsystem for a specific portal application
Portal campaign response time	Average response time of WebLogic Portal campaign subsystem for a specific portal application

For Portal web application activity and Portal performance, ADP displays information in both table and graph formats. For the other metrics, ADP shows the information in graph format. When you click the plus (+) icon next to the Portals node, ADP expands the tree to show all managed portal applications currently deployed on the WebLogic domain.

You can also see information specific to a particular portal application. By selecting a specific portal application, all information displayed in the Main Display Window changes to only show data relevant to this new context. For example, when a user selects a particular portal application under the Portals node, the Main Display Window only shows information specific to that portal application.

At the Portal level, you can navigate to different levels of the portal application by using different tabs. Use the tabs available to quickly access lower level components. [Table 16–11, "Portal Level Tab Description"](#) provides a list of the tabs available for portal level nodes and their descriptions.

**Table 16–11 Portal Level Tab Description**

Tab	Description
Summary	Performance summary specific to the selected portal.
Desktops	Performance summary for all the desktops associated with the selected portal.
Headers	Performance summary for all the headers associated with the selected portal.
Books	Performance summary for all the books associated with the selected portal.
Pages	Performance summary for all the pages associated with the selected portal.
Portlets	Performance summary for all the portlets associated with the selected portal.
Footers	Performance summary for all the footers associated with the selected portal.
WSRP Topology	View WSRP consumer-producer relationships and WSRP deployment topology.
Analysis	Two performance analytics - Multi-Point Regression Analysis performed at the portal level and Entity Performance Ranking performed at the portlet level.

**Table 16–11 (Cont.) Portal Level Tab Description**

Tab	Description
Events	SLO violation events associated with the selected portal.
Errors/Exceptions	Errors metrics associated with the selected portal.
Transactions	Transaction events associated with the selected portal and children below. By default, the Transactions tab is not enabled. To enable the Transactions tab, enable the <code>UIProvider.Modes=product,trace</code> property in the <code>Acsera.properties</code> file before starting the manager.

### 16.2.1.1 Desktops

Expand a particular portal application further to reach the Desktops node. By selecting the Desktops node, ADP provides a list of currently active desktops associated with that portal application.

This Desktop Summary includes the following metrics:

**Table 16–12 Desktop Summary Metrics**

Metrics	Description
Desktop arrivals	Total number of requests for a specific desktop
Desktop completions	Total number of requests fulfilled by a specific desktop
Desktop response time (ms)	Average response time for a specific desktop.

**Note:** Portal desktops are end-user facing entities. Metrics such as Desktop hits and response time represents request arrival rate and application performance respectively. Violations in thresholds set on these metrics would indicate unacceptable end-user experience.

ADP displays these metrics in both table and graph formats.

For example, when you have two active desktops, you can drill down further to a specific desktop by expanding the Desktops node. Again, clicking on the plus (+) icon expands the tree view for you.

When you select a node in the expanded tree to get more information specific for that desktop, ADP changes information in the Main Display Window to reflect the new context.

ADP not only shows the performance metrics associated with a specific node, but it also displays other relevant settings for that node. For example, there can be pre-configured Service Level Objectives (SLOs). These SLOs are displayed in the graphs as red lines.

Expand the desktop node to see Header, Footer, and Books. You can see detailed information for these components by clicking on the appropriate nodes.

**16.2.1.1.1 Display Portal Desktop - Desktop Structure Viewer** One of the unique capabilities of ADP is its automatic discovery and modeling of deployed applications. The Desktop Structure Viewer provides visibility into how a portal desktop is organized. To activate the Desktop Structure Viewer, right-click on a specific desktop. Select the Display Portal Desktop menu option to access the Desktop Structure Viewer.

After the Desktop Structure Viewer appears, you can navigate through the portal desktop structure by clicking on the appropriate book, page, or portlet. The ability to

see portal desktop structure using the same perspective as portal end-users is a unique value especially for the IT support staff.

With the Desktop Structure Viewer, the IT support staff can speak the same language with end-users while at the same time looking at performance oriented information for a specific component. The IT support staff can also use the Desktop Structure Viewer to isolate a particular performance problem. By drilling down from the top-level desktop to individual portlets, the IT support staff can get more insight into which components are having performance problems.

The Desktop Structure Viewer consists of two main panes. The pane on the left is the Desktop Structure pane. This pane allows you to graphically navigate the portal desktop. The pane on the right is called the Main Display Window. The Main Display window displays performance information in the context of the selected component in the Desktop Structure pane. As you navigate through the portal desktop and click different components, the Main Display Window provides information relevant for that selected context.

The Main Display Window shows relevant performance metrics for different portal desktop components - desktop, books, pages, and portlets.

Since ADP understands the WebLogic Portal framework and knows that a pageflow can be associated with a portlet, it is designed to allow easy access to the Pageflow Viewer from the Desktop Structure Viewer.

To activate the Pageflow Viewer, double-click the interesting portlet. In turn you can double-click the portlet in the Desktop Structure pane to open the appropriate pageflow in the Pageflow Viewer.

#### **16.2.1.2 Portlet Drill Down**

You can drill down on a portlet in the portal desktop view to activate the Display Architecture View.

1. Select a portlet under a node.
2. Double-click on a name to see the Portal Desktop Status page.
3. In the Portal Desktop Status window, right-click on a service box to select Display Architecture View.
4. See [Section 17.4, "Drill Down - Bottleneck Analysis"](#) on how to use the architecture view.

#### **16.2.1.3 Pageflow Viewer**

The Pageflow Viewer has two panes. The pane on the right is the Main Display Window. The Main Display Window shows information corresponding to the item selected in the left pane. The left pane shows either the Flow View or the Component View. You can choose to see either the Flow View or the Component View by selecting the appropriate tab.

The Main Display Window changes to show information relevant to the selected item in either the Flow View or the Component View.

#### **16.2.1.4 Books**

Expand a particular portal desktop further to see the Books node. By selecting the Books node, ADP provides a list of currently active books associated with the specific desktop.

This Books Summary includes the following metrics ([Table 16–13, "Book Summary Metrics"](#)):

**Table 16–13 Book Summary Metrics**

Metrics	Description
Book completions	Total number of requests fulfilled by a specific book
Book response time (ms)	Average response time for a specific book

ADP displays these metrics in both table and graph formats. For example, you can have two active books for the portal desktop. These active books are listed in the table and plotted in the graphs.

You can drill down further to a specific book by expanding its node. Click the plus (+) icon to expand the tree view. Expand the Books node to see a list of specific books configured.

When you select a particular active book, the Main Display Window shows the relevant information in that context.

### 16.2.1.5 Pages

Expand a particular book to see the Pages node. By selecting the Pages node, ADP provides a list of currently active pages associated with the specific book.

This Pages Summary includes the following metrics ([Table 16–14, "Pages Summary Metrics"](#)):

**Table 16–14 Pages Summary Metrics**

Metrics	Description
Page completions	Total number of requests fulfilled by a specific page
Page response time (ms)	Average response time for a specific page

ADP displays these metrics in both table and graph formats. For example, you can have one active page for a book. The active page is listed in the table and plotted in the graphs.

You can drill down further to a specific page by expanding the Pages node. Click the plus (+) icon to expand the tree view. This reveals the next level of components - Portlets.

### 16.2.1.6 Portlets

Expand a particular page to see the Portlets node. Select a Portlets node to view a list of currently active portlets associated with the specific page.

This Portlets Summary includes the following metrics ([Table 16–21, "Portlet Metrics"](#)):

**Table 16–15 Portlet Metrics**

Metrics	Description
Portlet completions	Total number of requests fulfilled by a specific portlet.
Portlet response time (ms)	Average response time for a specific portlet.



ADP displays these metrics in both table and graph formats. For example, you can have four active portlets for a particular page. These active portlets are listed in the table and plotted in the graphs.

Drill down further to a specific page by expanding the Portlets node. Click the (plus) + icon to expand the tree view. This provides additional information about the page.

## 16.2.2 WebSphere Portals

The Portals node under Oracle Enterprise Manager contains information about all deployed WebSphere Portal applications in the managed cells. The Portals node is organized hierarchically using the same framework developers use to build these Portal applications. The minimum and maximum response time measurements are stored in the database in addition to the average response time measurements. These metrics, if present, display visually in the window on the right pane.

For WebSphere Portal, this hierarchy contains the following (Table 16-16, "WebSphere Portal Hierarchy"):

**Table 16-16 WebSphere Portal Hierarchy**

Component	Description
Portals	The Portal is the logical containment unit for a Portal application. A typical Portal can contain a few desktops, several of books, tens of pages, and hundreds of portlets.
WebSphere	The WebSphere is the top-level container for the portal components included in that specific view of the portal.  Portal administrators can create new desktops beyond what portal developers create in WebLogic Workshop.
Virtual Portals	The top-level book contains all sub-books, pages, and portlets. The top-level book defines the initial menu navigation style used for the desktop. For each sub-book you add to a desktop you can select a different navigation style.
Content Root	Pages and sub-books are the navigable containers used for organizing portlets.
Header	Portlets are the containers that surface Web content and applications in your desktops.
Pages	Pages are containers within virtual portals, books, and sub-books. Pages often contain labels and portlets.
Labels	Labels are markers defining content within page containers.

When you click the Portals node under Oracle Enterprise Manager, ADP displays summary information on active portal applications. This summary includes the following (Table 16-17, "WebSphere Tree Summary"):

**Table 16-17 WebSphere Tree Summary**

Metrics	Description
Portal web application activity	A summary of user sessions for a specific portal application.
Portal completions	Total number of requests fulfilled by a specific portal application.
Portal response time (ms)	Average response time for a specific portal application.

For Portal web application activity and Portal performance, ADP displays information in both table and graph formats. For the other metrics, ADP shows the information in graph format.

When you click the plus (+) icon next to the Portals node, ADP expands the tree to show all managed portal applications currently deployed on the WebLogic domain.

You can also see information specific to a particular portal application. By selecting a specific portal application, all information displayed in the Main Display Window changes to only show data relevant to this new context. For example, when a user selects the *WebSphere* portal application under the Portals node, the Main Display Window only shows information specific to WebSphere portal application.

At the Portal level, you can navigate to different levels of the portal application by using different tabs. Use the tabs available to quickly access lower level components.

The following is a list of the tabs available for portal level nodes and their descriptions (Table 16–18, "Portal Level Tab Descriptions").

**Table 16–18 Portal Level Tab Descriptions**

Tab	Description
Summary	Performance summary specific to the selected portal
Analysis	Two performance analytics - Multi-Point Regression Analysis performed at the portal level and Entity Performance Ranking performed at the portlet level
Events	SLO violation events associated with the selected portal
WSRP Topology	View WSRP consumer-producer relationships and WSRP deployment topology
Errors/Exceptions	Errors metrics associated with the selected portal
Instrumentation	Includes performance data by different types of instrumentation probe points. There are different tabs available: Class, Method, Errors/Exceptions and Transactions. Each tab includes basic information such as Probe Point Name, Invocation Count, and Response Time. This detailed performance data can help you identify low-level bottlenecks. Refer <a href="#">Section 16.2.6.3, "Instrumentation"</a> for more details.

### 16.2.2.1 Virtual Portals

To reach the Virtual Portals node, further expand a particular portal application. By selecting this node, ADP provides a list of currently active portals associated with that portal application. See to view the Summary for the WebSphere portal application.

This Summary includes the following metrics (Table 16–19, "Virtual Portals Summary Metrics"):

**Table 16–19 Virtual Portals Summary Metrics**

Metrics	Description
Virtual Portal Completions	Total number of requests fulfilled by a specific portal.
Virtual Portal Response Time (ms)	Average response time for a specific portal.

ADP displays these metrics in both table and graph formats. For example, you can have one active portal for the WebSphere portal application. The Content Root is listed in the table and Plotted in the graphs.

You can drill down further to specific portlets by expanding the Content Root node. Again, clicking on the plus (+) icon expands the tree view for you. ADP changes information in the Main Display Window to reflect new context - portlet.

ADP not only shows the performance metrics associated with a specific node, but it also displays other relevant settings for that node.

**16.2.2.1.1 Display Virtual Portal - Structure Viewer** One of the unique capabilities of ADP is its automatic discovery and modeling of deployed applications. The Structure Viewer provides visibility into how a portal desktop is organized. To activate the Virtual

Portal Viewer, right-click on a specific portal. Select the Display Virtual Portal menu option to access the viewer.

After the Structure Viewer appears, you can navigate through the portal structure by clicking the appropriate header. The ability to see the portal structure using the same perspective as portal end-users is a unique value especially for the IT support staff.

With the Structure Viewer, the IT support staff can speak the same language with end-users while at the same time looking at performance oriented information for a specific component. The IT support staff can also use the Structure Viewer to isolate a particular performance problem. By drilling down from the top-level desktop to individual portlets, the IT support staff can get more insight into which components are having performance problems.

The Structure Viewer consists of two main panes. The pane on the left is the Structure Viewer pane. This pane allows you to graphically navigate the portal desktop. The pane on the right is called the Main Display Window. The Main Display window displays performance information in the context of the selected component in the Desktop Structure pane. As you navigate through the portal desktop and click different components, the Main Display Window provides information relevant for that selected context.

The Main Display Window shows relevant performance metrics for different portal components.

### 16.2.2.2 Pages

Expand a particular portal to see the Pages node. By selecting the Pages node, ADP provides a list of currently active pages associated with the specific book.

This Pages Summary includes the following metrics ([Table 16–20, "Pages Summary Metrics"](#)):

**Table 16–20 Pages Summary Metrics**

Metrics	Description
Page Completions	Total number of requests fulfilled by a specific page
Page Response Time (ms)	Average response time for a specific page

ADP displays these metrics in both table and graph formats.

### 16.2.2.3 Portlets

Expand a particular page to see the Portlets node. Select a Portlets node to view a list of currently active portlets associated with the specific page.

This Portlets Summary includes the following metrics ([Table 16–21, "Portlet Metrics"](#)):

**Table 16–21 Portlet Metrics**

Metrics	Description
Portlet Completions	Total number of requests fulfilled by a specific portlet
Portlet Response Time (ms)	Average response time for a specific portlet

ADP displays these metrics in both table and graph formats. For example, you can have four active portlets for the Content Root page. These active portlets are listed in the table and plotted in the graphs.

Drill down further to a specific page by expanding the Portlets node. Click the plus (+) icon to expand the tree view.

### 16.2.3 Oracle BPEL Processes

The BPEL Processes node in the navigation tree contains information about all deployed Oracle BPEL processes within the managed domain. ADP organizes information for various process nodes into domains.

In the right-hand pane, you can view the minimum and maximum response time measurements stored in the database in addition to the average response time, arrivals, errors, and completions measurements. These metrics, if present, display visually in the window on the right pane.

When you select the root of the BPEL Processes tree, ADP displays the BPEL Processes Summary in the Main Display Window.

The BPEL Process Summary includes the following ([Table 16–22, "BPEL Process Summary Metrics"](#)):

**Table 16–22 BPEL Process Summary Metrics**

Metrics	Description
Domain	Name of the OC4J domain container
Process	Name of the BPEL process
Arrivals	Total number of currently running instances for a specific BPEL process
Response Time (ms)	Average response time in milliseconds for a specific BPEL process
Completions	Total number of fulfilled requests for a specific BPEL process. A Completed status represents a BPEL process instance that has finished normally.
Errors	Total number of aborted instances of a specific BPEL process
Min Response Time (ms)	Minimum average response time in milliseconds for a specific BPEL process
Max Response Time (ms)	Maximum average response time in milliseconds for a specific BPEL process

ADP presents these metrics in a table format in the Main Display Window when you select the BPEL Processes node. Graphical representations of two metrics, Arrivals and Completions, are displayed below the table.

When you click the plus (+) icon next to the domains sub-node under the main BPEL Processes node, ADP expands the tree to show all managed BPEL domains currently deployed on that particular Oracle SOA Suite instance.

You can see information specific to a particular process. By selecting a specific process, all information displayed in the Main Display Window changes to only show data relevant to this new context.

To see the BPEL process work flow associated with a BPEL process, select the node, right-click and select the Display Functional View option. ADP displays the appropriate functional work flow diagram and associated performance data in a new pop-up window.

See [Table 16–23, "BPEL Functional View Summary"](#) for BPEL Functional View summary.

**Table 16–23 BPEL Functional View Summary**

Column/Metric	Description
Activity	Name of a specific activity in the BPEL process
Type	Control Type for a specific node
Arrivals	Number of requests that have arrived for a specific node
Response Time (ms)	Average response time for a specific node
Completions	Number of completed requests for a specific node
Errors	Number of aborted instances for a specific node
Response Time Min (ms)	Minimum response time for a specific node
Response Time Max (ms)	Maximum response time for a specific node

By looking at this summary table, you can determine which BPEL process node is running slowly and whether there are errors.

In addition to the summary, the following views are available for a node:

- Delay Analysis view
- Metadata view
- Partner Links view
- Partner Link Type Role view
- Partner Link Bindings view
- Modeled Entities view
- Topology view

You can get to these views by selecting the appropriate tab.

### 16.2.3.1 Delay Analysis View

Delay Analysis gives you a bird's eye view of a specific BPEL process. You can see what nodes in the BPEL process are taking up a majority of the average elapsed time. The red bar indicates the slowest BPEL process group or BPEL process node. The blue represents the time spent for the particular nodes.

### 16.2.3.2 Metadata View

The Metadata view displays the tables containing specific metadata associated with the selected active BPEL process being displayed in the left-hand pane. Information provided in this view includes caller and called class metadata information as well as general summarized metadata in relation to the BPEL process and the associated web services. [Table 16–24, "Metadata View Summary"](#) explains the metadata.

**Table 16–24 Metadata View Summary**

Column/Metric	Description
SummaryTable -Process	Name of the BPEL process node
SummaryTable -Web Service	Name of the web service being called from the BPEL process
SummaryTable -Version	Version of the web service being called from the BPEL process
SummaryTable -Location	Location of the web service being called from the BPEL process

**Table 16–24 (Cont.) Metadata View Summary**

Column/Metric	Description
Caller Table - Caller Class	Class name for the caller class that is calling the BPEL process
Caller Table - Caller Method	Class method for the caller class that is calling the BPEL process
Caller Table -Target Host	Target host that the caller class targeted to instantiate the BPEL process
Caller Table -Target Port	Target port that the caller class targeted to instantiate the BPEL process
Caller Table -Target URL	Target URL that the call class targeted to instantiate the BPEL process
Caller Table - Invocation Count	Number of invocations of the BPEL process instantiated by the caller class
Caller Table - Response Time	Average response time of the BPEL process instantiated by the caller class
Called Clients Table - Called Class	Class name of the class that was called by the BPEL process
Called Clients Table - Target URL	Target URL of the class that was called by the BPEL process
Called Clients Table - Invocation Count	Number of invocations made from the BPEL Process to the called class.
Called Clients Table - Response Time	Response time of the called class

### 16.2.3.3 Partner Links View

The partner links view provides detailed information on the various roles related to how and why the partner link service is being utilized. The information provided includes both the caller and callee roles, as well as the partner link type. See [Table 16–25, "Partner Links View Summary"](#).

**Table 16–25 Partner Links View Summary**

Column/Metric	Description
Partner Link	Name of the partner link
My Role	Role in regards to the BPEL process calling the partner link service
Partner Role	Role of the partner link service
Partner Link Type	Partner link category (type) of the service being called

### 16.2.3.4 Partner Link Type Role View

See [Table 16–26, "Partner Link Type Role View Summary"](#) describes the columns in the Partner Link Type Role view.

**Table 16–26 Partner Link Type Role View Summary**

Column/Metric	Description
Name	Name of the partner link
Link Type Name	Category (type) of the partner link
Port Type	Partner link service URL

### 16.2.3.5 Partner Link Bindings View

The Partner Link Bindings view provides insight into the actual roles and types of the partner link instances which represent web services that have been bound by the BPEL process. See [Table 16–27, "Partner Link Bindings View Summary"](#).

**Table 16–27 Partner Link Bindings View Summary**

Column/Metric	Description
Partner Link Role	Defines the web service role that the BPEL process will communicate with
Partner Link Type	Defines the web service type that the BPEL process will communicate with
WebService PortType	Name of the web service
WebService Port Namespace ID	URL of the webservice instance

### 16.2.3.6 Modeled Entities View

The modeled entities view consist of a list and count of the general entities as catalogued during the discovery phase of the resource configuration. The tables contain both a total entity count as well as a breakdown of the entity count by entity type. See [Table 16–28, "Modeled Entities Summary"](#).

**Table 16–28 Modeled Entities Summary**

Column/Metric	Description
Total Entities Modeled Table - Total	Total entities (static label)
Total Entities Modeled Table - Count	Total number of entities catalogued during the discovery phase of the BPEL process
Modeled Entities Table - Entity Type	Entity type being catalogued as part of the discovery phase of the BPEL process
Modeled Entities Table - Count	Total number of entities catalogued during the discovery phase of the BPEL process for a particular entity type

### 16.2.3.7 Topology View

The Topology View utilizes the modeled entities that were captured during the discovery process to provide a bird's eye view of all of the various high-level relationships between BPEL processes, web services, and business services. You can toggle between static and dynamic relationship views using the tabs at the top of the Topology pane.

### 16.2.3.8 Node Hierarchy

Expanding a particular BPEL process further, the first item you see is the Node Hierarchy node. By selecting the Node Hierarchy node, ADP provides a list of nodes associated with the specific process.

When you click the plus (+) icon next to a specific Node Hierarchy node, ADP expands the tree to show BPEL process nodes in the Node Hierarchy. Click an individual BPEL process node to see the load and performance of the selected node in the Main Display Window.

The BPEL process node information also includes the name of the method invoked. This information is displayed as part of the summary table at the top of the main view window.

## 16.2.4 Oracle ESB

The Oracle ESB node under Oracle Enterprise Manager contains information about all of the deployed Oracle ESB servers running in the managed domain. ADP organizes the information for various Oracle ESB nodes into various categories.

When you select the root of the ESB tree, ADP displays the ESB Summary in the Main Display Window.

The ESB Summary includes the following (Table 16–29, "ESB Summary Metrics"):

**Table 16–29 ESB Summary Metrics**

<b>Metric</b>	<b>Description</b>
ESB System	Name of ESB System
ESB Service	Name of the ESB Service identifier
Arrivals	Total number of ESB service instance arrivals
Completions	Total number of ESB service instance completions
Response Time	Total number of completed instances for a specific BPEL process. A Completed status represents a BPEL process instance that has finished normally.

ADP presents these metrics in a table format in the Main Display Window when you select the ESB node. When you click the plus (+) icon next to the ESB Systems sub-node under the main ESB node, ADP expands the tree to show all managed ESB Systems currently deployed on that particular Oracle SOA Suite instance.

You can see information specific to a particular ESB System. By selecting a specific ESB System, all information displayed in the Main Display Window changes to only show data and the topology relevant to this new context.

By looking at the summary table, you can find out which ESB node is running slowly and whether there are errors.

Besides the summary, the following views are available for the Node Hierarchy node:

- Service Details view
- Service Parent Details view
- Service Definition view
- Service Operations view
- Operation Routing Rules view
- Topology view

You can get to these views by selecting the appropriate tab.

#### 16.2.4.1 Service Details View

The Service Details view provides specific information related to the details of the bound service process instances. Instance IDs and other descriptive details are included as part of this view. See Table 16–30, "Service Details View Summary".

**Table 16–30 Service Details View Summary**

<b>Column/Metric</b>	<b>Description</b>
Service Name	Name of the ESB service
GUID	GUID of the ESB service
Qname	Canonical qualified name for the bound ESB service
Description	Description of the ESB service



### 16.2.4.2 Service Parent Details View

The Parent Service Details view provides specific information related to the details of the parent of the bound service process instances. Instance IDs, roles, and other descriptive details are included as part of this view. See [Table 16–31, "Service Parent Details View Summary"](#).

**Table 16–31 Service Parent Details View Summary**

Column/Metric	Description
Service Name	Name of the parent ESB service
ParentGUID	GUID of the parent ESB service
ParentQname	Canonical qualified name for the parent of the bound ESB service
ParentType	Parent type of the parent ESB service
MyRole	Role of the caller of the parent ESB service instance
ParentRole	Role of the callee of the parent ESB service instance

### 16.2.4.3 Service Definition View

The Service Definition view contains information regarding the bound ESB service including the Business Service (ESB) WSDL and Port Type as well as the associated URLs. See [Table 16–32, "Service Definition View Summary"](#).

**Table 16–32 Service Definition View Summary**

Column/Metric	Description
Service Name	Name of the ESB service
BusinessServiceWSDL	URL of the Business Service WSDL
BusinessServicePortType	Port type of the Business Service
ConcreteServiceWSDL	URL of the Concrete Service WSFL
ConcreteServiceURI	URI for the concrete service

### 16.2.4.4 Service Operations View

The Service Operations views provides details regarding the various method operations being executed. All information is provided in regards to the metadata associated with a specific business service instance. See [Table 16–33, "Service Operations View Summary"](#).

**Table 16–33 Service Operations View Summary**

Column/Metric	Description
Service Name	Name of the ESB service
Name	Service operation name being executed
GUID	GUID of the ESB service
Qname	Canonical qualified name for the bound ESB service
Element	Associated element within the ESB Service
SchemaLocation	Schema location for the associated ESB service
Type	Type of ESB service operation

### 16.2.4.5 Operation Routing Rules View

The Operation Routing Rules view provides various details regarding the operation routing rules for Business Service operations. This includes the specific instance business service names being utilized for operations. See [Table 16–34, "Operation Routing Rules View Summary"](#).

**Table 16–34 Operation Routing Rules View Summary**

Column/Metric	Description
Service Name	Name of the ESB service
Name	Instance name ID of the ESB service instance
GUID	GUID of the ESB service instance

## 16.2.5 Oracle WebCenter

Oracle WebCenter provides a set of features and services (for example, portlets, customization, and content integration) that simplify the process of reaching a solution with JSF applications. This solution brings information from multiple sources into a single interface, simplifying transactions and providing everything users need to support a given task within the application itself.

**Table 16–35 WebCenter Tree Summary**

Component	Description
ADF Taskflows	ADF task flows provide a modular approach for defining control flow in an application. See <a href="#">Section 16.2.5.1, "ADF Task Flows"</a> .
JSF Pages	JSF page definition files define the binding objects that populate the data in UI components at runtime. See <a href="#">Section 16.2.5.2, "JSF Pages"</a> .
Portlets	Portlets are the containers that surface Web content and applications on desktops. See <a href="#">Section 16.2.5.3, "Portlets"</a> .

### 16.2.5.1 ADF Task Flows

Instead of representing an application as a single large JSF page flow, you can break it up into a collection of reusable task flows. Each task flow contains a portion of the application's navigational graph. The nodes in the task flows are activities. An activity node represents a simple logical operation such as displaying a view, executing application logic, or calling another task flow. The transactions between the activities are called control flow cases. A task flow consists of activities and control flow cases that define the transitions between activities.

**16.2.5.1.1 User-Defined Taskflows** The following taskflows are available in WebCenter.

**Table 16–36 Taskflow Activities**

Activity Name	Description
Managed Beans	A backing bean that is managed by the JSF framework and used during the JSF page lifecycle.
Taskflow Method Calls	Invokes a method, typically a method on a managed bean.
Taskflow Views	Displays a JSF page or page fragment. Multiple view activities can represent the same page or same page fragment.
Taskflow URL Views	Redirects the root view port (for example, a browserpage) to any URL-addressable resource, even from within the context of an ADF region.

**Table 16–36 (Cont.) Taskflow Activities**

Activity Name	Description
Taskflow Calls	Calls an ADF bounded task flow from an ADFunbounded task flow or another bounded task flow
Routers	Evaluates an EL expression and returns an outcome based on the value of the expression. For example, a router in a credit check task flow might evaluate the return value from a previous method call and generate success, failure, or retry outcomes based on various cases. These outcomes can then be used to route control to other activities in the task flow.

**16.2.5.1.2 Web 2.0 Service** Oracle WebCenter provides a wide range of Web 2.0 capabilities, including discussion forums, wikis, blogs, content services, RSS, presence, instant messaging, linking, tagging, and search. Both developers and business users can easily add these services to their pages to maximize productivity.

**Table 16–37 Taskflow Activities**

Activity Name	Description
Managed Beans	A backing bean that is managed by the JSF framework and used during the JSF page lifecycle.
Taskflow Method Calls	Invokes a method, typically a method on a managed bean.
Taskflow Views	Displays a JSF page or page fragment. Multiple view activities can represent the same page or same page fragment.
Taskflow URL Views	Redirects the root view port (for example, a browserpage) to any URL-addressable resource, even from within the context of an ADF region.
Taskflow Calls	Calls an ADF bounded task flow from an ADFunbounded task flow or another bounded task flow
Routers	Evaluates an EL expression and returns an outcome based on the value of the expression. For example, a router in a credit check task flow might evaluate the return value from a previous method call and generate success, failure, or retry outcomes based on various cases. These outcomes can then be used to route control to other activities in the task flow.

### 16.2.5.2 JSF Pages

A typical JSF application couples a backing bean with each page in the application. The backing bean defines properties and methods that are associated with the UI components used on the page. The UI component's value is bound to the bean's property.

A Managed Bean is a backing bean that is managed by the JSF framework and used during the JSF page lifecycle.

### 16.2.5.3 Portlets

Portlets can display excerpts of other Web sites, generate summaries of key information, perform searches, and access assembled collections of information from a variety of data sources. You can use the portlets that Oracle or third parties provide, or create your own programmatically. Oracle WebCenter supports WSRP 1.0, WSRP 2.0, JSR 168, and Oracle PDK-Java. You can include any portlets adhering to those standards in your WebCenter applications.

## 16.2.6 Processes

The Processes node under Oracle Enterprise Manager contains information about all deployed WebLogic business processes in the managed domain. ADP organizes information for various process nodes into the following major categories:

- Node Hierarchy
- Persistent Containers
- Instrumentation

The minimum and maximum response time measurements are stored in the database in addition to the average response time measurements. These metrics, if present, display visually in the window on the right pane.

When you select the root of the Processes tree, ADP displays the Processes Summary in the Main Display Window. See [Table 16–38, "Process Summary Metrics"](#).

**Table 16–38 Process Summary Metrics**

Metrics	Description
Process	Name of process
Running	Total number of currently running instances for a specific process
Suspended	Total number of suspended instances for a specific process. A Suspended request from a user is a common cause for a process instance to go into a Suspended state.
Frozen	Total number of frozen instances for a specific process
Completed	Total number of completed instances for a specific process. A Completed status represents a process instance that has finished normally.
Aborted	Total number of aborted instances for a specific process
Terminated	Total number of terminated instances for a specific process. An external Terminate request would terminate a process instance.
Average Execution Time (ms)	Average execution completion time for a specific process

**Tip:** Statistics on the number of process instances with Terminated, Aborted, and Frozen states can indicate abnormal operation of the WebLogic Integration application or container. It is possible to unfreeze Frozen process instances from WLI Console.

ADP presents these metrics in a table format in the Main Display Window when you select the Processes node. Graphical representations of two metrics, Running Instances and Average Execution Time, are displayed below the table.

When you click the plus (+) icon next to the Processes node, ADP expands the tree to show all managed processes currently deployed on the WebLogic domain.

You can see information specific to a particular process by selecting a specific process. All information displayed in the Main Display Window changes to only show data relevant to this new context.

To see the process work flow associated with a particular process, select the process node, right-click and select the Display Functional View option. ADP displays the appropriate functional work flow diagram and associated performance data in a new pop-up window.

### 16.2.6.1 Node Hierarchy

When expanding a particular process further, the first item you see is the Node Hierarchy node. By selecting the Node Hierarchy node, ADP provides a list of nodes associated with the specific process. See [Table 16–39, "Node Hierarchy Summary"](#).

**Table 16–39 Node Hierarchy Summary**

Column/Metric	Description
Node	Name of a specific node
ID	Process Node ID for a specific node
Type	Control Type for a specific node
Method	Node Method Name for a specific node
Arrivals	Number of Requests Arrived for a specific node
Active	Number of Active Instances for a specific node
Elapsed Time (ms)	Average Time Elapsed to Complete an Instance for a specific node
Completions	Number of Completed Instances for a specific node
Aborts	Number of Aborted Instances for a specific node
Exceptions	Number of Exception Encountered for a specific nod.

By looking at this summary table, you can determine which process node is running slowly and whether there are aborts or exceptions.

The following additional views are available for the Node Hierarchy node:

- Delay Analysis view
- Events view

You can get to these views by selecting the appropriate tab.

**16.2.6.1.1 Delay Analysis View** Delay Analysis gives you a bird's eye view of a specific process. You can see what nodes in the process are taking up a majority of the average elapsed time. The red bar indicates the slowest process group or process node. The blue represents the time spent for the particular nodes.

**16.2.6.1.2 Events View** The Events view shows a list of SLO violations events relevant to this process in a table format. The Events view table includes the following information ([Table 16–40, "Events View Summary"](#)):

**Table 16–40 Events View Summary**

Column/Metric	Description
Start Time	Start time for the process instance that violated a SLO
Entity Name	Name of the process node that violated a SLO
SLO Name	Name of the violated SLO
Service URI	URI of the process that violated a SLO
Application	Name of the application that violated a SLO
Event Type	Violation type (violation or cautionary)
Entity Type	Violation Metric type
SLO Threshold	Type of threshold (high or low)

**Table 16–40 (Cont.) Events View Summary**

Column/Metric	Description
SLO Trigger Value	Value that triggered a SLO violation

When you click the plus (+) icon next to a specific Node Hierarchy node, ADP expands the tree to show process nodes in the Node Hierarchy. Click an individual process node to see the load and performance of the selected node in the Main Display Window.

The process node information also includes the name of the method invoked. This information is displayed as part of the summary table at the top of the main view window.

### 16.2.6.2 Persistent Containers

When you expand a particular process further, the Persistent Containers node is included. By selecting the Persistent Containers node, ADP provides a list of persistence performance statistics relevant to the selected process.

As you select the root of the Persistent Containers tree, a summary of all Persistent Containers relevant to the selected process is presented. For example, a summary can contain the following high level items:

- Container persistence invocations
- Container persistence response time (milliseconds)
- Entity EJB activity
- Entity EJB cache
- Entity EJB transactions
- Entity EJB locking

These items are displayed in both table and graph formats.

The Persistent Containers Summary includes different tables:

- Entity EJB Activity
- Entity EJB Cache
- Entity EJB Transactions
- Entity EJB Locking

**16.2.6.2.1 Entity EJB Activity Table** Entity EJB Activity table ([Table 16–41, "Entity EJB Activity Table"](#)) includes the following information:

**Table 16–41 Entity EJB Activity Table**

Metrics	Description
EJB	Name of the Entity EJB
In Use	Number of instances for a specific Entity EJB currently being used from the free pool. [Snapshot Count]
Idle	Number of instances for a specific Entity EJB currently in the idle state in the free pool. These bean instances are available for use. [Snapshot Count]
Waits	Number of Threads currently waiting for a specific Entity EJB bean instance from the free pool [Snapshot Count]

**Table 16–41 (Cont.) Entity EJB Activity Table**

Metrics	Description
Timeouts	Total number of Threads that have timed out waiting for an available bean instance from the free pool [Aggregated Count]

**Tip:** Pay attention to Waits and Timeouts metrics. Activities in the Waits metric and increasing count in the Timeouts metric are signs that requests waiting to be serviced by the EJB container. Ideally, 0 should be indicated for these metrics.

**16.2.6.2.2 Entity EJB Cache Table** Entity EJB Cache table ([Table 16–42, "Entity EJB Cache Table"](#)) includes the following information:

**Table 16–42 Entity EJB Cache Table**

Metrics	Description
EJB	Name of the Entity EJB
Hits	Total number of times an attempt to access the Entity EJB instance from the cache succeeded [Aggregated Count]
Accesses	Total number of attempts to access the Entity EJB instance from the cache [Aggregated Count]
Size	Number of beans instances from this EJB Home currently in the EJB cache [Snapshot Count]
Activations	Total number of beans from this EJB Home that have been activated [Aggregated Count]
Passivations	Total number of beans from this EJB Home that have been passivated [Aggregated Count]

**Tip:** Passivation (serializing EJB state information to disk) and activation (reconstituting EJB state information from disk) are resource intensive operations. Ideally, it is preferable to see low level of activity in these metrics.

**16.2.6.2.3 Entity EJB Transactions Table** Entity EJB Transactions table ([Table 16–43, "Entity EJB Transactions Table"](#)) includes the following information:

**Table 16–43 Entity EJB Transactions Table**

Metrics	Description
EJB	Name of the Entity EJB
Commits	Total number of transactions that have been committed for this EJB. [Aggregated Count]
Rollbacks	Total number of transactions that have been rolled back for this EJB. [Aggregated Count]
Timeouts	Total number of transactions that have timed out for this EJB. [Aggregated Count]

**Tip:** High number of EJB Transaction Rollbacks may indicate problems with the data used - for some reason the target database is unable to commit the change. High number of EJB Transaction Timeouts may indicate problems accessing the database including network outage, database lock contention, database outage, and more.

**16.2.6.2.4 Entity EJB Locking Table** Entity EJB Locking table ([Table 16–44, "Entity EJB Locking Table"](#)) includes the following information:

**Table 16–44 Entity EJB Locking Table**

Metrics	Description
EJB	Name of the Entity EJB
Entries	Number of Entity EJB instances currently locked [Snapshot Count]
Lock Accesses	Total number of attempts to obtain a lock on an Entity EJB instance [Aggregated Count]
Current Waiters	Number of Threads that currently waiting for a lock on an Entity EJB instance [Snapshot Count]
Total Waiters	Total number Threads that have waited for a lock on an Entity EJB instance [Aggregated Count]
Timeouts	Total number Threads that have timed out waiting for a lock on an Entity EJB instance [Aggregated Count]

**Tip:** Pay attention to Current Waiters and Timeouts. These metrics can indicate possible performance problems caused by EJB Locking. Ideally, 0s should be displayed for these metrics.

By looking at the activities related to Persistence Containers, you can determine if EJB persistence calls are causing performance problems.

### 16.2.6.3 Instrumentation

When expanding a particular process further, the last item you see is the Instrumentation node. Click the plus (+) icon next to Instrumentation to expand the tree to reveal the following categories of instrumentation:

- Class
- Methods
- Errors/Exceptions
- Transactions

The Class node in the Instrumentation tree provides the following information ([Table 16–45, "Class Node"](#)):

**Table 16–45 Class Node**

Column/Metric	Description
Probe Point	Class name in which instrumentation probe point is inserted
Response Time (ms)	Average response time for a specific class
Invocation Count	Number of times a specific class is called

The Method node in the Instrumentation tree provides the following information ([Table 16–46, "Method Node"](#)):

**Table 16–46 Method Node**

Column/Metric	Description
Probe Point	Method name in which instrumentation probe point is inserted
Response Time (ms)	Average response time for a specific method
Invocation Count	Number of times a specific method is called



The Errors/Exceptions and Transactions are described in [Section 16.1.17, "Architecture View"](#).

## 16.2.7 Web Services

The Web Services node in the navigation tree contains information about all deployed Web Services in the managed domain. By selecting the Web Services node under Oracle Enterprise Manager, ADP shows the Web Services Summary in the Main Display Window.

This summary view lists all discovered web services and their associated URL entry points. Below this list, ADP lists out all active web services and their performance data (invocation count and response time).

When you click the plus (+) icon next to the Web Services node, ADP expands the tree to show all monitored web services currently deployed on the WebLogic domain.

When you select a specific web service, ADP displays performance data associated with the selected web service. Click the plus (+) icon next to a specific web service to expand the tree to show all public operations associated with that web service.

The Operations table provides the following information ([Table 16-47, "Operations Table"](#)):

**Table 16-47 Operations Table**

Column/Metric	Description
Operation	Name of the web service operation
Invocation Count	Number of times the operation is called
Response Time (ms)	Average response time for the operation in milliseconds
Delay (ms)	Overall delay contributed by the operation in milliseconds

## 16.2.8 Pageflows

The Pageflows node in the navigation tree contains information about all deployed pageflows in the managed domain. By selecting the Pageflows node under Oracle Enterprise Manager, ADP shows the Pageflows Summary in the Main Display Window.

## 16.2.9 Services

The Services node in the navigation tree contains information about all external entry points into the managed domain. ADP currently monitors the following types of services:

- HTTP
- EJBs
- JDBC

Selecting each service type reveals service summary in the Main Display Window.

The minimum and maximum response time measurements are stored in the database in addition to the average response time measurements. These metrics, if present, display visually in the window in the right pane.

ADP displays entry point activity summary associated with the selected EJB service.

**Tip:** Setting thresholds at some of these entry points enables ADP to monitor the performance of key business services. When a violation event occurs, you can begin investigating from the Service node.

### 16.2.9.1 HTTP

Expanding the HTTP node under the Services node reveals a list of discovered HTTP based entry points into the managed domain. HTTP service end points include JSPs, struts actions, and servlet mappings. These discovered HTTP entry points are listed by their root context. When you select a specific HTTP entry point, ADP displays the associated summary in the Main Display Window.

When a specific file is selected, ADP displays more detailed performance data.

Method level performance data is displayed when you select a specific HTTP service entry point.

**Table 16–48 HTTP Performance Summary**

Column/Metric	Description
Servlet	Name of the servlet associated with the selected service
Method	Name of the method invoked by external call
Arrivals	Total number of requests received by this method
Invocation Count	Total number of method invocations
Response Time (ms)	Average method response time in milliseconds

### 16.2.9.2 EJBs

To view the performance summary for EJBs invoked from outside the JVM, click the EJBs node.

**Table 16–49 EJB Performance Summary**

Column/Metric	Description
EJB	Name of the EJB
Invocation Count	Number of times the EJB is called
Response Time (ms)	Average response time for the EJB in milliseconds
Delay (ms)	Overall delay contributed by the EJB in milliseconds

**Tip:** As a general rule, external calls that terminate in EJBs are RMI calls. Web services calls that ultimately terminate in EJBs use SOAP and enter the application server via HTTP.

### 16.2.9.3 JDBC

To bring up the performance summary for JDBC operations invoked from outside of the JVM, click the JDBC node.

**Table 16–50 JDBC Performance Summary**

Column/Metric	Description
SQL Statement	Generalized SQL Statement executed by the JDBC operation
Class	Name of the class used in the JDBC operation
Method	Name of the method used in the JDBC operation

**Table 16–50 (Cont.) JDBC Performance Summary**

Column/Metric	Description
Invocation Count	Number of times the JDBC operation is called
Response Time (ms)	Average response time for the JDBC operation in millisecond
Delay (ms)	Overall delay contributed by the JDBC operation in milliseconds

## 16.2.10 WSRP Producers

The Web Services Remote Portlet (WSRP) Producers node in the navigation tree contains information about the WebLogic WSRP consumer - producer relationships in the managed domain. By selecting an entity in the WSRP node, ADP displays the performance measurements for the associated WSRP consumer or producer.

WebLogic Portal can act as either a WSRP remote producer or as a consumer. When acting as a consumer, WebLogic Portal's remote--or proxy--portlets are WSRP-compliant. These portlets present content that is collected from WSRP-compliant producers, allowing you to use external sources for portlet content, rather than having to create its content or its structure yourself.

The following types of portlets can be exposed with WSRP inside a WebLogic portal:

- Page flow portlets
- JavaServer Pages (JSP) portlets
- Struts portlets
- Java portlets (JSR168; supported only for complex producers)
- JavaServer Faces (JSF) portlets

The minimum and maximum response time measurements are captured in addition to the average response time measurements. These metrics, if present, display visually in the window in the right pane.

### 16.2.10.1 WSRP Summary

To view the WSRP Producers Summary:

1. Select the WSRP Producers node to show the WSRP Producers Summary tab.

The WSRP Producers summary includes the following table ([Table 16–51, "WSRP Producers Summary"](#)):

**Table 16–51 WSRP Producers Summary**

Column	Description
WSRP Producer	Name of the producer portlet
WSDL URL	URL of the WSD.

2. To view the portlet details, click the Consumer Portlets node under the WSRP Producers.

The following tables are in this view:

- WSRP Producer Information
- WSRP Consumer Portlet Performance

Select the portlet name and right click. Select **Open** to drill down to more detailed view.

- **WSRP Producer Portlets**

Select the portlet name and right click. Select **Open** to drill down to more detailed view.

**Table 16–52 WSRP Producers Information**

Column	Description
TestPortlets	Defined by the user for the Producer, for example description, handle, and more
URL	Lists the details of each item under the TestPortlet column

**Table 16–53 WSRP Consumer Portlet Performance**

Column	Description
Portal	The Portal is the logical containment unit for a Portal application. A typical Portal can contain a few desktops, several of books, tens of pages, and hundreds of portlets.
Desktop	The desktop is the top-level container for the portal components included in that specific view of the portal.  Portal administrators can create new desktops beyond what portal developers create in WebLogic Workshop.
Book	The top-level book contains all sub-books, pages, and portlets. The top-level book defines the initial menu navigation style used for the desktop. For each sub-book, you add to a desktop you can select a different navigation style.
Page	Pages and sub-books are the navigable containers used for organizing portlets.
Portlet	Portlets are the containers that surface Web content and applications in your desktops.
Response Time (ms)	Average response time in milliseconds.
Completions	Number of Completed Instances for a specific node.
Response Time Min (ms)	Minimum response time in milliseconds.
Response Time Max (ms)	Maximum response time in milliseconds.

**Table 16–54 WSRP Producer Portlets**

Column	Description
Producer Portlet	Name of the producer portlet
Producer	Name of the producer

3. Click a portlet name in the tree view to see the performances associated with the consumer and producer portlets.

### 16.2.10.2 WSRP Topology

Use this option to visually explore WSRP consumer - producer relationships and the WSRP deployment topology.

To view the WSRP Topology:

1. Select the WSRP Producers node to show the WSRP Topology tab.

2. Click the WSRP Topology tab to view the details.

### 16.2.10.3 Display Portal Desktop

The portal desktop is described in [Section 16.2.1.1.1, "Display Portal Desktop - Desktop Structure Viewer"](#).

Access the Architecture View:

1. To view the portal desktop for a specific portlet, right-click the portlet name under the Consumer Portlet node.
2. Select Display Portal Desktop.
3. You can drill down to view the Architecture View from this view. See the instructions in [Section 16.2.1.2, "Portlet Drill Down"](#).

## 16.2.11 Integration

The Integration node under Oracle Enterprise Manager contains information about the WebLogic Integration resources in the managed domain. By selecting the Integration node under Oracle Enterprise Manager, ADP displays the Integration Summary.

The Integration Summary includes the following ([Table 16–55, "Integration Summary"](#)):

**Table 16–55 Integration Summary**

Metric	Description
Process	Name of process
Running	Total number of currently running instances for a specific process
Suspended	Total number of suspended instances for a specific process
Frozen	Total number of frozen instances for a specific process
Completed	Total number of completed instances for a specific process
Aborted	Total number of aborted instances for a specific process
Terminated	Total number of terminated instances for a specific process
Average Execution Time	Average execution completion time for a specific process

**Tip:** Statistics on the number of process instances with Terminated, Abort, and Frozen states can indicate abnormal operation of WebLogic Integration application or container. It is possible to unfreeze Frozen process instances from WLI Console.

ADP presents these metrics in a table format in the Main Display Window when you select the Integration node. Graphical representations of these metrics, Running Instances, Completed Instances, and Average Execution Time, are displayed below the table.

Expand the Integration tree by clicking on the plus (+) icon next to Integration node.

The expanded Integration tree allows you to look at various components of WebLogic Integration and help identify performance bottlenecks. This section explains the nodes under the Integration Tree.

### 16.2.11.1 Health

In the expanded Integration tree, the first node you see is the Health node. Under the Health node, ADP lists various subsystems in WebLogic Integration. By expanding the Health node, you can see the following:

- Execute Queues
- Async Dispatchers
- Sync Dispatchers
- JMS Destinations
- Stateless Containers
- Persistent Containers

You can get to the health information specific to each of these subsystems by clicking the appropriate node. Also, you can get to a particular instance of a subsystem.

**16.2.11.1.1 Execute Queues** In the Execute Queues node, ADP provides operational statistics of each execute queues configured for WebLogic Integration. Select the Execute Queues node in the Monitor Workspace to display the Execute Queues Summary in the Main Display Window.

The Execute Queues Summary provides the following information ([Table 16–56, "Execute Queues Summary"](#)):

**Table 16–56** *Execute Queues Summary*

Metric	Description
Execute Queue	Execute Queue ID
Aggregated Execute Queue	Aggregated execute queue statistics per resource
Idle Threads	Current number of idle threads in a specific Execute Queue
Pending Threads	Current number of pending threads in a specific Execute Queue
Requests	Total number of requests serviced for a specific Execute Queue
Total Threads	Total number of threads configured in a specific Execute Queue

**Tip:** Pay attention to Idle Threads and Pending Threads counts. Rapidly decreasing Idle Threads count combined with rapidly increasing Pending Threads count can indicate a backup in the Execute Queue.

Use the following guidelines to adjust the Execute Queue Thread Count ([Table 16–57, "Guidelines to Adjust the Execute Queue Thread Count"](#)):

**Table 16–57** *Guidelines to Adjust the Execute Queue Thread Count*

Execute Queue Is Backed Up?	Application Is CPU Bound?	Adjustment Guideline
Yes	No	Increase execute queue thread count.
Yes	Yes	Decrease thread count and explore JVM or application issues that may be causing high CPU utilization.

ADP presents these metrics in a table format in the Main Display Window when you select the Health node. Graphical representations of these metrics, Idle Treads, Pending Threads, and Requests, are displayed below the table.

Expand the Health tree by clicking on the plus (+) icon next to Health node. You can get the same summary as previously described for a specific execute queue.

**16.2.11.1.2 Async Dispatchers** In the Async Dispatcher node, ADP provides operational statistics of each of the Async Dispatchers configured in WebLogic Integration. Select the Async Dispatchers node in the Monitor Workspace to show the Async Dispatchers Summary in the Main Display Window.

The Async Dispatcher Summary includes the following information ([Table 16-58](#), "Async Dispatcher Summary"):

**Table 16-58 Async Dispatcher Summary**

Metric	Description
EJB	Name of the Message Driven EJB
In Use	Number of instances for a specific Message Driven EJB currently in use
Idle	Number of instances for a specific Message Driven EJB currently in the idle state
Waits	Number of instances for a specific Message Driven EJB currently in the wait state
Timeouts	Number of instances for a specific Message Driven EJB currently in the timeout state
Commits (Transaction)	Total number of commits performed for a specific Message Driven EJB
Rollbacks (Transaction)	Total number of transaction rollbacks performed for a specific Message Driven EJB
Timeouts (Transaction)	Total number of transaction timeouts performed for a specific Message Driven EJB

**Tip:** Rapidly increasing counts in MDB Waits and Timeouts metrics may indicate a tuning opportunity for the MBD container.

Furthermore, increasing numbers in the Transaction Rollbacks and Timeouts metrics may indicate issues interacting with the database.

Ideally, these metrics should not increase rapidly.

ADP presents these metrics in a table format in the Main Display Window when you select the Async Dispatchers node. Graphical representation of one metrics, Message Driven EJB in use, is displayed below the table.

Expand the Async Dispatchers tree by clicking the plus (+) icon next to Async Dispatchers node. You can get the same summary as previously described for a specific async dispatcher.

**16.2.11.1.3 Sync Dispatchers** In the Sync Dispatchers node, ADP provides operational statistics of each of the Sync Dispatchers used by WebLogic Integration. Select the Sync Dispatchers node in the Monitor Workspace to show the Sync Dispatchers Summary in the Main Display Window.

The Sync Dispatcher Summary includes the following information ([Table 16-59](#), "Sync Dispatcher Summary"):

**Table 16–59 Sync Dispatcher Summary**

Metric	Description
EJB	Name of the Stateless EJB
In Use	Number of instances for a specific Stateless EJB currently in use
Idle	Number of instances for a specific Stateless EJB currently in the idle state
Waits	Number of instances for a specific Stateless EJB currently in the waits state
Timeouts	Number of instances for a specific Stateless EJB currently in the timeouts state

**Tip:** Rapidly increasing counts in Stateless EJB Waits and Timeouts metrics may indicate performance issues and a tuning opportunity for the EJB container. Ideally, these metrics should not increase at a rapid pace.

ADP presents these metrics in a table format in the Main Display Window when you select the Sync Dispatchers node. Graphical representation of one metrics, Stateless EJB in use, is displayed below the table.

Expand the Sync Dispatchers tree by clicking on the plus (+) icon next to Sync Dispatchers node. You can get the same summary as previously described for a specific sync dispatcher.

**16.2.11.1.4 JMS Destinations** In the JMS Destination node, ADP provides operational statistics of each of the JMS Destinations used by WebLogic Integration. Select the JMS Destinations node in the Monitor Workspace to show the JMS Destinations Summary in the Main Display Window.

JMS Destination Summary includes the following tables: JMS destination message statistics and JMS destination byte statistics. The JMS destination message statistics table includes the following information ([Table 16–60, "JMS Destination Message Statistics"](#)).

**Table 16–60 JMS Destination Message Statistics**

Column/Metric	Description
JMS Destination	Name of the JMS destination
Message Current	Number of JMS messages currently at a specific JMS destination
Message High	Maximum number of JMS messages at a specific JMS destination
Message Pending	Number of JMS messages pending to be delivered to a specific JMS destination
Message Received	Total number of JMS messages at a specific JSM destination

**Tip:** Pay attention to Message Pending metric. Too many pending messages in a specific JMS destination could result in a performance slowdown. Rapidly increasing count for the Message Pending metric may indicate a performance problem and a JMS destination tuning opportunity.

The JMS destination byte statistics table includes the following information ([Table 16–61, "JMS Destination Byte Statistics"](#)).



**Table 16–61 JMS Destination Byte Statistics**

Column/Metric	Description
JMS Destination	Name of the JMS destination
Byte Current	Byte count of JMS messages currently at a specific JMS destination
Byte High	Maximum byte count of JMS messages at a specific JMS destination
Byte Pending	Byte count of JMS messages pending to be delivered to a specific JMS destination
Byte Received	Total Byte count of JMS messages at a specific JMS destination

ADP presents these metrics in table format in the Main Display Window when you select the JMS Destinations node. Graphical representations of the following metrics, Message pending and Byte pending, are displayed below the table.

Expand the JMS Destinations tree by clicking on the plus (+) icon next to JMS Destinations node. You can get the same summary as described above for a specific JMS destination.

**16.2.11.1.5 Stateless Containers** In the Stateless Containers node, ADP provides operational statistics of each of the Stateless Containers used by WebLogic Integration. Select the Stateless Containers node in the Monitor Workspace to show the Stateless Containers Summary in the Main Display Window.

The Stateless Containers Summary includes the following information ([Table 16–62, "Stateless Containers Summary"](#)):

**Table 16–62 Stateless Containers Summary**

Metric	Description
EJB	Name of the Stateless EJB
Stateless EJB Transactions	Runtime statistics. You can monitor stateless session EJBs using the metrics in this table.
In Use	Number of instances for a specific Stateless EJB currently being used from the free pool [Snapshot Count]
Idle	Number of instances for a specific Stateless EJB currently in the idle state in the free pool. These bean instances are available for use. [Snapshot Count]
Waits	Number of Threads currently waiting for a specific Stateless EJB instance from the free pool [Snapshot Count]
Timeouts	Total number of Threads that have timed out waiting for an available bean instance from the free pool [Aggregated Count]

ADP presents these metrics in a table format in the Main Display Window when you select the Stateless Containers node. Graphical representation of one metrics, Stateless EJB in use, is displayed below the table.

Expand the Stateless Containers tree by clicking on the plus (+) icon next to Stateless Containers node. You can get the same summary as previously described for a specific stateless container.

**16.2.11.1.6 Persistent Containers** In the Persistent Containers node, ADP provides operational statistics of each of the Persistent Containers used by WebLogic Integration. Select the Persistent Containers node in the Monitor Workspace to show the Persistent Containers Summary in the Main Display Window.

The Persistent Containers Summary includes the following tables:

- Entity EJB Activity
- Entity EJB Cache
- Entity EJB Transactions
- Entity EJB Locking

Entity EJB Activity table includes the following information ([Table 16–63, "Entity EJB Activity"](#)):

**Table 16–63 Entity EJB Activity**

Metrics	Description
EJB	Name of the Entity EJB
In Use	Number of instances for a specific Entity EJB currently being used from the free pool [Snapshot Count]
Idle	Number of instances for a specific Entity EJB currently in the idle state in the free pool. These bean instances are available for use. [Snapshot Count]
Waits	Number of Threads currently waiting for a specific Entity EJB bean instance from the free pool [Snapshot Count]
Timeouts	Total number of Threads that have timed out waiting for an available bean instance from the free pool [Aggregated Count]

**Tip:** Pay attention to Waits and Timeouts metrics. Activities in the Waits metric and increasing count in the Timeouts metric are signs that requests waiting to be serviced by the EJB container. Ideally, 0 should be indicated for these metrics.

Entity EJB Cache table includes the following information ([Table 16–64, "Entity EJB Cache"](#)):

**Table 16–64 Entity EJB Cache**

Metrics	Description
EJB	Name of the Entity EJB
Hits	Total number of times an attempt to access the Entity EJB instance from the cache succeeded [Aggregated Count]
Accesses	Total number of attempts to access the Entity EJB instance from the cache [Aggregated Count]
Size	Number of beans instances from this EJB Home currently in the EJB cache [Snapshot Count]
Activations	Total number of beans from this EJB Home that have been activated [Aggregated Count]
Passivations	Total number of beans from this EJB Home that have been passivated [Aggregated Count]

**Tip:** Passivation (serializing EJB state information to disk) and activation (reconstitute EJB state information from disk) are resource intensive operations. Ideally, Oracle recommends low level of activity in these metrics.

Entity EJB Transactions table includes the following information ([Table 16–65, "Entity EJB Transactions"](#)):

**Table 16–65 Entity EJB Transactions**

Metric	Description
EJB	Name of the Entity EJB
Commits	Total number of transactions that have been committed for this EJB [Aggregated Count]
Rollbacks	Total number of transactions that have been rolled back for this EJB [Aggregated Count]
Timeouts	Total number of transactions that have timed out for this EJB [Aggregated Count]

**Tip:** High number of EJB Transaction Rollbacks may indicate problems with the data used - for some reason the target database is unable to commit the change. High number of EJB Transaction Timeouts may indicate problems accessing the database including network outage, database lock contention, database outage, and more.

Entity EJB Locking table includes the following information ([Table 16–66, "Entity EJB Locking"](#)):

**Table 16–66 Entity EJB Locking**

Metric	Description
EJB	Name of the Entity EJB
Entries	Number of Entity EJB instances currently locked [Snapshot Count]
Lock Accesses	Total number of attempts to obtain a lock on an Entity EJB instance [Aggregated Count]
Current Waiters	Number of Threads that currently waiting for a lock on an Entity EJB instance [Snapshot Count]
Total Waiters	Total number Threads that have waited for a lock on an Entity EJB instance [Aggregated Count]
Timeouts	Total number Threads that have timed out waiting for a lock on an Entity EJB instance [Aggregated Count]

**Tip:** Pay attention to Current Waiters and Timeouts. These metrics can indicate possible performance problems caused by EJB Locking. Ideally, 0s should be displayed for these metrics.

ADP presents these metrics in a table format in the Main Display Window when you select the Persistent Containers node. Graphical representations of three metrics, Entity EJB in use, Entity EJB cache access, and Entity EJB lock access, are displayed below the table.

Expand the Persistent Containers tree by clicking on the plus (+) icon next to Persistent Containers node. You can get the same summary as previously described for a specific persistent container.

### 16.2.11.2 Performance

In the expanded Integration tree, the second node you see is the Performance node. ADP provides the Performance Summary for WebLogic Integration in the Main Display Window when the Performance node is selected.

The Performance Summary includes the following tables: Process Node and Events. The Process Node table provides performance information for various process nodes running in WebLogic Integration. It includes the following information ([Table 16–67, "Performance - Process Node Summary"](#)):

**Table 16–67 Performance - Process Node Summary**

Column/Metric	Description
Node	Name of a specific node
ID	Process Node ID for a specific node
Type	Control Type for a specific node
Method	Node Method Name for a specific node
Arrival	Number of Requests Arrived for a specific node
Active	Number of Active Instances for a specific node
Elapsed Time	Average Time Elapsed to Complete an Instance for a specific node
Completions	Number of Completed Instances for a specific node
Aborts	Number of Aborted Instances for a specific nod.
Exceptions	Number of Exception Encountered for a specific node

**Tip:** You can use Arrivals and Elapsed Time data collected by ADP to characterize the performance of your installation. Since ADP measures performance at cluster level, you are capturing the actual performance of your configuration. You can also perform simple capacity planning analysis by plotting Arrivals versus Elapsed Time (arrival rate versus response time). Ask your Oracle consultant for more information.

The Events table provides a list of SLO violations triggered relevant to WebLogic Integration. It includes the following information ([Table 16–68, "Performance - Events Node Summary"](#)):

**Table 16–68 Performance - Events Node Summary**

Column/Metric	Description
Start Time	Start time for the process instance that violated a SLO
Entity Name	Name of the process node that violated a SLO
SLO Name	Name of the violated SLO
Service URI	URI of the process that violated a SLO
Application	Name of the application that violated a SLO
Event Type	Violation type (violation or cautionary)
Entity Type	Violation Metric type
SLO Threshold	Type of threshold (high or low)
SLO Trigger Value	Value that triggered a SLO violation

### 16.2.11.3 Channels

In the expanded Integration tree, the third node you see is the Channels node. ADP shows the Channels Summary for various channels configured for WebLogic Integration.

The Channels Summary includes the following information ([Table 16–69, "Channels Summary"](#)):

**Table 16–69 Channels Summary**

Column/Metric	Description
Channel	Name of channel
Type	Channel type
Message Count	Total number of messages processed for a specific channel
Dead Message Count	Total number of dead messages for a specific channel

**Tip:** Increasing count in the Dead Message Count metric may indicate a configuration issue. When the Message Broker is unable to determine the URI to send a message to, the message is sent to the appropriate deadletter channel. Ensure the URI configured for the channel is reachable.

Expand the Channels tree by clicking the plus (+) icon next to Channels node. You can get the same health summary as previously described for a specific channel.

#### 16.2.11.4 Subscribers

In the expanded Integration tree, the fourth node you see is the Subscribers node. ADP shows the Subscribers Summary for various subscribers configured for WebLogic Integration.

Expand the Subscribers tree by clicking the plus (+) icon next to Subscribers node. You can get specific information about an individual subscriber.

## 16.2.12 Applications

The Applications node in the navigation tree contains information about all deployed applications in the managed domain. By selecting the Applications node, ADP displays the Applications Summary.

The Applications Summary includes the following information ([Table 16–70, "Applications Summary"](#)):

**Table 16–70 Applications Summary**

Column/Metric	Description
Application	Name of application
Status	Operations status for a specific application
Response Time (ms)	Average response time in milliseconds for a specific application. This is the average of response times of all JSPs and servlets contained in the deployment archive.
Invocation Count	Total number of invocations for a specific application. This is the total invocation count of all JSPs and servlets contained in the deployment archive.

**Tip:** Application is a packaging unit in Java EE. Each EAR, WAR, and JAR files deployed to the application server is considered an individual application. These metrics track performance and arrival rate of these entities.

ADP presents these metrics in a table format in the Main Display Window when you select the Applications node. Graphical representations of the following metrics, Response Time, Invocation Count, and Active Sessions, are displayed below the table.

Expand the Applications tree by clicking the plus (+) icon next to Applications node. You can get more information about a specific application.

ADP displays performance summary for the selected application in the Main Display Window. You can obtain additional performance data by clicking different tabs in the Main Display Window.

The Applications Summary includes the following tabs ([Table 16-71, "Applications Summary Tabs"](#)):

**Table 16-71 Applications Summary Tabs**

Tab Name	Description
Summary	Includes performance data at the application level including time-based trend graphs of Application Response Time, Application Invocation Count, and Application Active Sessions. The invocation count and response time for the top 10 slowest servlets, the usual application entry points, are also included.
Response Times	Includes time-based trend graphs of component response times. Graphs include Servlet Response Time, EJB Response Time, and JDBC Response Time.
Invocations	Includes time-based trend graphs of component invocation counts. Graphs include Servlet Invocation Count, EJB Invocation Count, and JDBC Invocation Count.
Errors/Exceptions	Errors metrics associated with the selected portal.
Transactions	Transaction events associated with the selected portal and children below. By default, the Transactions tab is not enabled.
Modeled Entities	Includes a catalog of entities modeled by ADP. Only the modeled entities associated with the selected application are included.
Instrumentation	Includes performance data by different types of instrumentation probe points. There are different tabs available: Class, Method, and SQL. Each tab includes basic information such as Probe Point Name, Invocation Count, and Response Time. This detailed performance data can help you identify low-level bottlenecks.
Topology	Includes the topology view associated with the selected application.

Under each named application node, ADP displays performance and other relevant information specific to that application. For example, by clicking the children nodes, the relevant data is displayed in the Main Display Window. Application response time and invocations measurements can be reached by clicking the panes in the Main Display Window.

In this section, we will further expand on the following nodes:

- Services
- Dependencies
- Deployments
- Workshop Projects
- Web Applications
- Stateless Beans
- Stateful Beans
- Entity Beans
- Message Driven Beans

---

**Note:** The number of children nodes available under each application node depends solely on the complexity of the selected application. Simple Java EE web applications will not have nodes like Workshop Projects, Stateless Beans, Stateful Beans, Entity Beans, and Message Driven Beans.

---

### 16.2.12.1 Services

The Services node includes all the external entry points associated with the selected application. When this node is selected, ADP displays a summary view in the Main Display Window. ADP displays the performance data associated with various entry points associated with the selected application.

**Tip:** The children nodes under the Services node include entry point specific performance data.

### 16.2.12.2 Dependencies

The Dependencies node shows a list of internal and external components and share resources that a specific application depends on for its normal operation. When the Dependencies node is selected, ADP displays all external references made by the application in the Main Display Window. The following is a list of columns and their descriptions ([Table 16–72, "Dependencies Column Descriptions"](#)):

**Table 16–72 Dependencies Column Descriptions**

Column/Metric	Description
Name	Display name of the component or resource used by the application. If this is undefined in the Deployment Descriptor, the reference name for the component is used.
Reference	Reference name of the component or resource used by the application.
Reference Type	Component or resource type.
Referer Component	Name of the component that is part of the application which obtained the reference to external component or resource.
Referer Module	Name of the module that is part of the application which obtained the reference to external component or resource.

ADP displays all the references associated with components in the selected application.

The Dependencies node can be further expanded by clicking the plus (+) icon. The children nodes of the Dependencies node are organized by type. Here are the list of dependency types and their descriptions ([Table 16–73, "Dependency Types"](#)):

**Table 16–73 Dependency Types**

Dependency Type	Description
Data Sources	All shared data sources used by the application
Entity Beans	All entity beans used by the application
Session Beans	All session beans used by the application
JMS Queues	All JMS queues used by the application for publishing JMS messages
JMS Topics	All JMS topics subscribed by the application
Web Services	All web services used by the application

When a specific node is selected, ADP displays relevant performance summary. These nodes can also be expanded by clicking the plus (+) icons. The expanded tree includes specific components and share resources used by the application.

The Performance summary view associated with the Data Sources node under Dependencies provides information on both connection pools and SQL statements.

For more information on the metric description, refer to [Section 16.1.18, "Metric Types"](#).

### 16.2.12.3 Deployments

The Deployments node shows the architecture of the deployed application. When this node is selected, ADP shows all the modules deployed as part of this application. The default view in the Main Display Window shows the active module-level call path. [Table 16–74, "Deployment Tabs"](#) lists the tabs available as part of this summary view and their descriptions.

**Table 16–74 Deployment Tabs**

Tab Name	Description
Module Level Execution	Shows the active calling relationships among various Java EE modules (EAR, WAR, JAR, and more). Shared resources are also included. This is the default Architecture View at the module level.
Module Level	Shows the potential calling relationships among various Java EE modules. Shared resources are also included. By default, the Module Level tab is not enabled.
Instrumentation	Includes detailed performance data at the method level. The table includes caller components, caller method, callee (target) component, callee module, invocation count, and response time.
SQL Statement	Includes all SQL statements executed as part of this application. It also includes performance information such as invocation count and response time.

Active module-level call path is displayed as the default view for the Deployments node of a selected application.

Double-click a specific module to trigger ADP to display the architecture of the selected module.

Expand the Deployments node by clicking the plus (+) icon to reveal all the deployed modules in this application. Further expanding the nodes at the *module* level reveals components associated with the selected module. Further expanding the nodes at the *component* level reveals methods associated with the selected component.

When you select one of these children nodes (module, component, and method levels), ADP displays associated tabs for active call path diagram, static call path diagram, instrumentation and SQL statements.

**Tip:** Use the active call path diagram as a guide to identify entities with performance data. If an entity does not have performance data, ADP displays *No data available for the selected time frame* in the Main Display Window.

### 16.2.12.4 Workshop Projects

The Workshop Projects node includes performance information about modules and components created using the Oracle WebLogic Workshop. These modules and components include WebLogic Integration processes, WebLogic Integration web services, and WebLogic Portal pageflows.



Workshop Project node and its children nodes provide performance data associated with WLI processes, web services, and WLP pageflows.

When you select a specific children node, ADP displays detailed performance information.

### 16.2.12.5 Web Applications

The Web Applications node includes performance information related to the Web Applications modules and components associated with the selected application. Click the Web Applications node to reveal a performance summary in the Main Display Window. Click the plus (+) icon to expand the Web Applications node to reveal various web modules deployed as part of this application.

Click the plus (+) icon to expand on a specific web module and reveal different groupings for web components, for example, Pageflows, Struts Modules and Servlets. Clicking one of these nodes triggers ADP to display rolled up performance summary for the entire grouping. You can further expand these nodes by clicking the plus (+) icon to reveal more detailed information. Fully expanded Web Applications node contains all web modules organized by type.

Detailed performance information at the individual pageflow, struts action, and servlet levels will be displayed when you click the lowest level nodes.

### 16.2.12.6 Stateless Beans

The Stateless Beans node includes activity information related to the stateless EJB components associated with the selected application. Click the Stateless Beans node to reveal an activity summary in the Main Display Window. Click the plus (+) icon to expand the Stateless Beans node to reveal various stateless EJBs deployed as part of this application.

You can further select individual nodes to obtain detailed activity information. Selecting a specific Stateless Bean node triggers ADP to display detailed activity metrics.

The detailed view contains the following activity metrics ([Table 16-75, "Stateless Beans Detail View"](#)):

**Table 16-75 Stateless Beans Detail View**

Column/Metric	Description
EJB	Name of the stateless EJB.
In Use	Number of instances for a specific stateless EJB currently being used from the free pool. [Snapshot Count]
Idle	Number of instances for a specific stateless EJB currently in the idle state in the free pool. These bean instances are available for use. [Snapshot Count]
Waits	Number of threads currently waiting for a specific stateless EJB bean instance from the free pool. [Snapshot Count]
Timeouts	Total number of threads that have timed out waiting for an available bean instance from the free pool. [Aggregated Count]

**Note:** The metrics reported in the Stateless Beans node are reported by the MBean (Management Bean) of the EJB container. These activity metrics can be used for checking the overall health of the EJB container. When the EJB container is restarted, these metrics are reset.

### 16.2.12.7 Stateful Beans

The Stateful Beans node includes activity information related to the stateful EJB components associated with the selected application. Click the Stateful Beans node to reveal an activity summary in the Main Display Window. Click the plus (+) icon to expand the Stateful Beans node to reveal various stateful EJBs deployed as part of this application.

You can further select individual nodes to obtain detailed activity information.

The Stateful EJB Summary includes the following tables:

- Stateful EJB Cache
- Stateful EJB Transactions
- Stateful EJB Locking

**16.2.12.7.1 Stateful EJB Cache** Stateful EJB Cache table includes the following information (Table 16–76, "Stateful EJB Cache"):

**Table 16–76 Stateful EJB Cache**

Metrics	Description
EJB	Name of the Stateful EJB
Hits	Total number of times an attempt to access the Stateful EJB instance from the cache succeeded [Aggregated Count]
Accesses	Total number of attempts to access the Stateful EJB instance from the cache [Aggregated Count]
Size	Number of beans instances from this Stateful Home currently in the EJB cache [Snapshot Count]
Activations	Total number of beans from this Stateful Home that have been activated [Aggregated Count]
Passivations	Total number of beans from this Stateful Home that have been passivated [Aggregated Count]

**Tip:** Passivation (serializing EJB state information to disk) and activation (reconstitute EJB state information from disk) are resource intensive operations. Ideally, Oracle recommends low level of activity in these metrics.

**16.2.12.7.2 Stateful EJB Transactions** Stateful EJB Transactions table includes the following information (Table 16–77, "Stateful EJB Transactions"):

**Table 16–77 Stateful EJB Transactions**

Metrics	Description
EJB	Name of the Stateful EJB
Commits	Total number of transactions that have been committed for this Stateful [Aggregated Count]
Rollbacks	Total number of transactions that have been rolled back for this Stateful [Aggregated Count]
Timeouts	Total number of transactions that have timed out for this EJB [Aggregated Count]

**Tip:** High number of EJB Transaction Rollbacks may indicate problems with the data used; for some reason the target database is unable to commit the change. High number of EJB Transaction Time-outs may indicate problems accessing the database including network outage, database lock contention, and database outage.

**16.2.12.7.3 Stateful EJB Locking** Stateful EJB Locking table includes the following information ([Table 16–78, "Stateful EJB Locking"](#)):

**Table 16–78 Stateful EJB Locking**

Metric	Description
EJB	Name of the Stateful EJB
Entries	Number of Stateful EJB instances currently locked [Snapshot Count]
Lock Accesses	Total number of attempts to obtain a lock on an Stateful EJB instance [Aggregated Count]
Current Waiters	Number of Threads that currently waiting for a lock on an Stateful EJB instance [Snapshot Count]
Total Waiters	Total number Threads that have waited for a lock on an Stateful EJB instance [Aggregated Count]
Timeouts	Total number Threads that have timed out waiting for a lock on an Stateful EJB instance [Aggregated Count]

**Tip:** Pay attention to Current Waiters and Time-outs. These metrics can indicate possible performance problems caused by EJB Locking. Ideally, 0s should be displayed for these metrics.

ADP presents these metrics in a table format in the Main Display Window when you select the Stateful Beans node. Graphical representations of two metrics, Stateful EJB cache access, and Stateful EJB lock access, are displayed below the table.

By looking at the activities related to Stateful EJBs, you can determine if there any abnormal activities associated with Stateful EJBs.

---

**Note:** The metrics reported in the Stateful Beans node are reported by the MBean (Management Bean) of the EJB container. These activity metrics can be used for checking the overall health of the EJB container. When the EJB container is restarted, these metrics are reset.

---

### 16.2.12.8 Entity Beans

The Entity Beans node includes activity information related to the Entity EJB components associated with the selected application. Click the Entity Beans node to reveal an activity summary in the Main Display Window. Click the plus (+) icon to expand the Entity Beans node to reveal various Entity EJBs deployed as part of this application.

You can further select individual nodes to obtained detailed activity information. Selecting a specific Entity Bean node triggers ADP to display detailed activity metrics.

The Entity EJB Summary includes the following tables:

- Entity EJB Activity
- Entity EJB Cache
- Entity EJB Transactions
- Entity EJB Locking

**16.2.12.8.1 Entity EJB Activity** Entity EJB Activity table includes the following information ([Table 16–79, "Entity EJB Activity"](#)):

**Table 16–79 Entity EJB Activity**

Metrics	Description
EJB	Name of the Entity EJB.
In Use	Number of instances for a specific Entity EJB currently being used from the free pool. [Snapshot Count]
Idle	Number of instances for a specific Entity EJB currently in the idle state in the free pool. These bean instances are available for use. [Snapshot Count]
Waits	Number of Threads currently waiting for a specific Entity EJB instance from the free pool. [Snapshot Count]
Timeouts	Total number of Threads that have timed out waiting for an available bean instance from the free pool. [Aggregated Count]

**Tip:** Pay attention to Waits and Timeouts metrics. Activities in the Waits metric and increasing count in the Timeouts metric are signs that requests are waiting to be serviced by the EJB container. Ideally, 0 should be indicated for these metrics.

**16.2.12.8.2 Entity EJB Cache** Entity EJB Cache table includes the following information (Table 16–80, "Entity EJB Cache"):

**Table 16–80 Entity EJB Cache**

Metrics	Description
EJB	Name of the Entity EJB
Hits	Total number of times an attempt to access the Entity EJB instance from the cache succeeded [Aggregated Count]
Accesses	Total number of attempts to access the Entity EJB instance from the cache [Aggregated Count]
Size	Number of beans instances from this EJB Home currently in the EJB cache [Snapshot Count]
Activations	Total number of beans from this EJB Home that have been activated [Aggregated Count]
Passivations	Total number of beans from this EJB Home that have been passivated [Aggregated Count]

**Tip:** Passivation (serializing EJB state information to disk) and activation (reconstituting EJB state information from disk) are resource intensive operations. Ideally, Oracle recommends a low level of activity in these metrics.

**16.2.12.8.3 Entity EJB Transactions** Entity EJB Transactions table includes the following information (Table 16–81, "Entity EJB Transactions"):

**Table 16–81 Entity EJB Transactions**

Metric	Description
EJB	Name of the Entity EJB
Commits	Total number of transactions that have been committed for this EJB [Aggregated Count]
Rollbacks	Total number of transactions that have been rolled back for this EJB [Aggregated Count]
Timeouts	Total number of transactions that have timed out for this EJB [Aggregated Count]

**Tip:** High numbers of EJB Transaction Rollbacks may indicate problems with the data used; for some reason the target database is unable to commit the change. High numbers of EJB Transaction Timeouts may indicate problems accessing the database including network outage, database lock contention, database outage, and more.

**16.2.12.8.4 Entity EJB Locking** Entity EJB Locking table includes the following information (Table 16–82, "Entity EJB Locking"):

**Table 16–82 Entity EJB Locking**

Metric	Description
EJB	Name of the Entity EJB
Entries	Number of Entity EJB instances currently locked [Snapshot Count]
Lock Accesses	Total number of attempts to obtain a lock on an Entity EJB instance [Aggregated Count]
Current Waiters	Number of Threads that currently waiting for a lock on an Entity EJB instance [Snapshot Count]
Total Waiters	Total number Threads that have waited for a lock on an Entity EJB instance [Aggregated Count]
Timeouts	Total number Threads that have timed out waiting for a lock on an Entity EJB instance [Aggregated Count]

**Tip:** Pay attention to Current Waiters and Timeouts. These metrics can indicate possible performance problems caused by EJB Locking. Ideally, 0s should be displayed for these metrics.

When you select the Entity Beans node, ADP presents these metrics in a table format in the Main Display Window. Graphical representations of the following metrics, Entity EJB in use, Entity EJB cache access, and Entity EJB lock access, are displayed below the table.

Expand the Entity Beans tree by clicking the plus (+) icon next to Entity Beans node. You can get the same summary as previously described for a specific Entity EJB.

By looking at the activities related to Entity EJBs, you can determine if there any abnormal activities associated with Entity EJBs.

---

**Note:** The metrics reported in the Entity Beans node are reported by the MBean (Management Bean) of the EJB container. These activity metrics can be used for checking the overall health of the EJB container. When the EJB container is restarted, these metrics are reset.

---

### 16.2.12.9 Message Driven Beans

The Message Driven Beans node includes activity information related to the message driven EJB components associated with the selected application. Click the Message Driven Beans node reveals an activity summary in the Main Display Window. Click the plus (+) icon to expand the Message Driven Beans node to reveal various message driven EJBs deployed as part of this application.

You can further select individual nodes to obtained detailed activity information.

The Message Driven EJB Summary includes the following tables:

- Message Driven EJB Activity

- Message Driven EJB Transactions

**16.2.12.9.1 Message Driven EJB Activity** Message Driven EJB Activity table includes the following information ([Table 16–83, "Message Driven EJB Activity"](#)):

**Table 16–83 Message Driven EJB Activity**

Metric	Description
EJB	Name of the Message Driven EJB.
In Use	Number of instances for a specific Message Driven EJB currently being used from the free pool. [Snapshot Count]
Idle	Number of instances for a specific Message Driven EJB currently in the idle state in the free pool. These bean instances are available for use. [Snapshot Count]
Waits	Number of Threads currently waiting for a specific Message Driven EJB instance from the free pool. [Snapshot Count]
Timeouts	Total number of Threads that have timed out waiting for an available bean instance from the free pool. [Aggregated Count]

**Tip:** Pay attention to Waits and Timeouts metrics. Activities in the Waits metric and increasing count in the Timeouts metric are signs that requests are waiting to be serviced by the EJB container. Ideally, 0 should be indicated for these metrics.

**16.2.12.9.2 Message Driven EJB Transactions** Message Driven EJB Transactions table includes the following information ([Table 16–84, "Message Driven EJB Transactions"](#)):

**Table 16–84 Message Driven EJB Transactions**

Metric	Description
EJB	Name of the Message Driven EJB
Commits	Total number of transactions that have been committed for this EJB [Aggregated Count]
Rollbacks	Total number of transactions that have been rolled back for this EJB [Aggregated Count]
Timeouts	Total number of transactions that have timed out for this EJB [Aggregated Count]

**Tip:** High numbers of EJB Transaction Rollbacks may indicate problems with the data used; for some reason the target database is unable to commit the change. High numbers of EJB Transaction Timeouts may indicate problems accessing the database including network outage, database lock contention, database outage, and more.

ADP presents these metrics in a table format in the Main Display Window when you select the Message Driven Beans node. Graphical representation of the Message Driven EJB in use metric is displayed below the table.

By looking at the activities related to Message Driven EJBs, you can determine if there are any abnormal activities associated with Message Driven EJBs.

---



---

**Note:** The metrics reported in the Message Driven Beans node are reported by the MBean (Management Bean) of the EJB container. These activity metrics can be used for checking the overall health of the EJB container. When the EJB container is restarted, these metrics are reset.

---



---

### 16.2.13 Oracle WebLogic Resources

The Resources node under Oracle Enterprise Manager contains information for the managed domain organized by logical clusters, machines, servers, and more. You can look for low-level technology metrics organized by technology subsystems for a specific WebLogic Server.

The Resources tree includes the following nodes ([Table 16–85](#), "WebLogic Resources Tree"):

**Table 16–85** WebLogic Resources Tree

Example Node	Description
CSS Domain	Name of the WebLogic Domain configured
b-15/192.168.128.15	ID of the physical machine
cgServer	Name of the WebLogic Server configured
Applications	Performance measurements of all deployed applications running on this server
JDBC	Information of all configured JDBC resources for this server
JMS Servers	Information of all JMS destinations configuration for this server
Execute Queues	Information of all Execute Queues configured for this server
JVM	JVM information including Heap Size for this server
JRockit	JRockit information including Heap Size for this server
Modeling Status	Entities modeled by ADP for this server
ADP Modules	Status of the ADP Java Agent Module for this server

Expand these nodes by clicking the plus (+) icon next to the node name to get more information.

If the ADP OS Agent is deployed on the machine, clicking on the physical machine ID would show OS metrics collected by the OS Agent. These OS metrics include CPU Usage, Disk Usage, and Physical Memory Usage.

### 16.2.14 WebSphere Resources

The Resources node under Oracle Enterprise Manager contains information for the managed domain organized by logical clusters, machines, servers, and more. You can look for low-level technology metrics organized by technology subsystems for a specific WebSphere Server.

The Resources tree includes the following nodes ([Table 16–86](#), "WebSphere Resources Tree"):

**Table 16–86 WebSphere Resources Tree**

Example Node	Description
WPSSUn	Resource name, for example, WPSUn
WebSphere Portal	Machine name, for example, WebSphere_Portal
WebSphere Portal	Server name, for example, WebSphere_Portal
Applications	Performance measurements of all deployed applications running on this server
JDBC	Information of all configured JDBC resources for this server
JMS Servers	Information of all JMS destinations configuration for this server
Thread Pools	Performance information about all threads used by the container to process request
JVM	JVM information including Heap Size for this server
WebServices	Performance measurements about web services deployed in the container
Sessions	Information about active HTTP sessions
Transactions	Information about transactions performance
Cache	Information about cache performance
ORB	Information about ORB performance
Modeling Status	Modeled entities for the container
ADP Modules	Status of the ADP Java Agent Module for this server
Applications	Performance information about the applications deployed in the container

Expand these nodes by clicking the plus (+) icon next to the node name to get more information.

Clicking on the physical machine ID would show OS metrics. These OS metrics include CPU Usage, Disk Usage, and Physical Memory Usage.

## 16.2.15 Oracle Resources

The Resources node under Oracle Enterprise Manager contains information for the managed domain organized by logical clusters, machines, servers, and more. You can look for low-level technology metrics organized by technology subsystems for a specific Oracle AS Server.

The Resources tree includes the following nodes ([Table 16–87, "Oracle Resources Tree"](#)):

**Table 16–87 Oracle Resources Tree**

Example Node	Description
Managed System Resource Name	Top-level Resource name, for example, oc4j_soa
Oracle AS Server	Machine name which can be navigated to both within or outside a cluster, for example, oc4j_soa@192.168.1.119 which includes both the server name and the host server IP address
Applications	Performance measurements of all deployed applications running on this server
JDBC	Information of all configured JDBC resources for this server
JMS Servers	Information of all JMS destinations configuration for this server
Thread Pools	Performance information about all threads used by the container to process requests



**Table 16–87 (Cont.) Oracle Resources Tree**

Example Node	Description
JVM	JVM information including Heap Size for this server
BPEL Processes	Performance measurements about BPEL Processes deployed in the container
ESB	Performance measurements about ESB services deployed in the container
Modeling Status	Modeled entities for the container
ADP Modules	Status of the ADP Java Agent Module for this server
Applications	Performance information about the applications deployed in the container

Clicking the physical machine ID would show OS metrics. These OS metrics include CPU Usage, Disk Usage, and Physical Memory Usage.

### 16.2.16 Custom Metrics

The Custom Metrics node under Oracle Enterprise Manager contains all the custom metrics you defined. Currently ADP supports custom metrics for Java classes. When Custom Metrics node is selected, ADP displays various summaries. You can select individual entities to get more detailed performance information.

Expanding the Custom Metrics node reveals a list of Java classes with custom metrics configured.

The following is a list of columns in the Custom Class Performance table and their descriptions ([Table 16–88, "Custom Class Performance"](#)):

**Table 16–88 Custom Class Performance**

Column/Metric	Description
Caller Class	Fully qualified name of the class that is making the inbound call
Caller Method	Method name in the class that is making the inbound call
Class	Fully qualified name of the class that is the destination of the inbound call
Invocation Count	Total number of times the inbound call is made
Response Time (ms)	Average response time of the inbound call in milliseconds

### 16.2.17 CAMM Node

The CAMM node in the navigation tree contains information for the CAMM environment for the monitored WebLogic domain, WebSphere cell, or Oracle AS cluster. Select the CAMM node to see the CAMM Java Agent status for the WebLogic domain.

The CAMM Java Agent status includes the following ([Table 16–89, "CAMM Java Agent Status"](#)):

**Table 16–89 CAMM Java Agent Status**

Column/Metric	Description
Server	Name of the WebLogic server, WebSphere cell, or Oracle AS cluster
Container Status	Operational status of the WebLogic, WebSphere, or Oracle AS server (running or not)
Agent In Sync	Version synchronization between CAMM and CAMM Agent status (true or false)
EJB Installed	CAMM EJB installation status (true or false)

**Table 16–89 (Cont.) CAMM Java Agent Status**

Column/Metric	Description
Agent Installed	CAMM Java Agent installation status
Agent Activated	CAMM Java Agent activation status
Agent Status	CAMM Java Agent operational status
Server Type	Identifies server as administration, individual, or clustered server
Admin URI	Location of the domain admin server
Manager RMI Registry Host	Host name of the CAMM RMI registry
Manager RMI Registry Port	Port number of the CAMM RMI registry
EJB Major Version	CAMM EJB major version
EJB Minor Version	CAMM EJB minor version
EJB Build ID	CAMM EJB build number - for version synchronization check
Agent Major Version	CAMM Java Agent major version
Agent Minor Version	CAMM Java Agent minor version
Agent Build ID	CAMM Java Agent build number - for version synchronization check

Click the Modeling Status node under CAMM node to see a table of all modeled entities in the managed domain. This table shows all the managed clusters, servers, and applications in the CAMM environment. Mismatches between the Modeling Status table and your environment are indications of configuration problems.

You can use this information to debug and resolve CAMM configuration issues.

## 16.2.18 Service Component Architecture (SCA)

Service Component Architecture (SCA) provides a set of features and services that simplify the process of detecting the presence of Service-Oriented Architecture (SOA) components.

**Table 16–90 SCA Composites**

Composite	Description
Services	Metrics related to Services defined on the SOA composite.
Wires	Metadata related to Wires defined in the SOA composite
References	Metrics related to References defined in the SOA composite
Components	Metrics related to Components within the SOA composite

### 16.2.18.1 Components

The following components make up the Service Component Architecture:

**Table 16–91 Components in SCA**

Component	Description
Decision Services	Metrics related to components in the Decision Services engine
Mediators	Metrics related to components in the Mediator engine

**Table 16–91 (Cont.) Components in SCA**

Component	Description
Human Workflows	Metrics related to components in the Human Workflow engine
BPEL	Metrics related to components in the BPEL engine

## 16.3 Exploring the Configuration Tab

Using the Configuration tab you can set up the resources you want to monitor using ADP.

The configurations explained in this section are:

- [Section 16.3.1, "Database Configuration"](#)
- [Section 16.3.2, "Resource Configuration"](#)
- [Section 16.3.3, "Service Level Objective Configuration"](#)
- [Section 16.3.4, "SLO Blackout Configuration"](#)
- [Section 16.3.5, "Action Configuration"](#)
- [Section 16.3.6, "Custom Metric Configuration"](#)

A running ADP manager must be registered in Enterprise Manager. Refer to [Chapter 10.1, "Quick Steps for Installing and Configuring the ADP Manager"](#) for steps of how to perform the registration. After the registration, Enterprise Manager continues to keep the manager as a valid manager even if it is down. When this occurs, the Enterprise Manager UI displays the ADP manager as Unreachable.

### 16.3.1 Database Configuration

The Database Configuration page lists the databases accessible to ADP which you want to monitor. You can configure a database to be used by ADP, edit an existing database configuration, delete a database configuration, and enable a configuration.

Refer to [Section 10.3.3, "Setting Up ADP Data Repository"](#) for details.

### 16.3.2 Resource Configuration

The Resource Configuration node in the Configuration tree enables you to create resources (for example, target application server domains) that can be monitored by ADP.

### 16.3.3 Service Level Objective Configuration

The Service Level Objective Configuration node in the Configuration tree enables you to create service level objectives (SLOs) that can be monitored by ADP. A service level objective is a measurable attribute, for example, availability. Service Level Agreements (SLA) are made up of SLOs.

SLOs are hierarchical which allows you to set service levels at any level within the modeled hierarchy of an application.

#### 16.3.3.1 Service Level Objectives Configuration

Service Level Objectives Configuration node in the Configuration tree allows you to manage SLOs.

In this window you can:

- Create and edit a specific SLO to open the Service Level Objective Editor.
- View or edit the selected SLO.

### 16.3.4 SLO Blackout Configuration

The SLO Blackout Configuration node in the Configuration tree enables you to create time periods when information will not be monitored for a specific SLO. You can define blackouts by a SLO file, an individual SLO, or by entity.

### 16.3.5 Action Configuration

Action Configuration node in the Configuration tree allows you to manage actions for ADP. Actions are triggered by a SLO violation event. ADP supports the following actions:

- Issue a SNMP trap
- Send an e-mail
- Execute a script
- Log to a file

In the Action Configurations window you can:

- Click **Create Action** to create a specific type of action.

In the Action Configuration window, double-click on a specific action to see its configuration information. You can edit and copy edit selected action. ADP automatically enforces referential integrity during the deletion process.

---



---

**Note:** You can include a set of SLO variables into e-mail, script, and log actions. This feature significantly increases the value of these actions by using real-time performance data. See [Table 16–92, "List of SLO Variables"](#) for a list of SLO variables.

---



---

**Table 16–92 List of SLO Variables**

SLO Variable	Description	Example Value
\$EventType	SLO event type (Violation or Cautionary)	Event.SLO.Cautionary
\$EventAttributes.SLOName	Name of the SLO fired	CSR Portal Desktop Response Time Violation
\$Event.Attributes.SLOType	Metric where SLO violation was observed	Metrics.J2EE.JVM.HeapFree
\$EventAttributes.TriggerValue	Value of metric when SLO threshold was exceeded	35001
\$EventAttributes.TriggerThreshold	Threshold type (High or Low)	High
\$Entity.InfrastructureID	Name of the platform	WebLogic
\$Entity.NodeID	Server node where SLO violation was observed	B93/192.168.3.93
\$Entity.DomainID	Domain in which SLO violation was observed	mydomain

**Table 16–92 (Cont.) List of SLO Variables**

SLO Variable	Description	Example Value
\$Entity.ResourceID	Cluster in which SLO violation was observed	my_cluster
\$Entity.EntityTypeID	Type of the entity in which the SLO violation was observed	J2EE.JVM
\$StartTime	Start time of the SLO violation	1112322030000
\$EndTime	Stop time of the SLO violation	1112322045000

**Tip:** Customize your alert using SLO variables. The following is an example of a customized message for a Mail Action:

```
SLO Event:
SLO Name = $EventAttributes.SLOName;
Event Type = $EventType;
Trigger Domain = $Entity.DomainID;
Trigger Application = $Entity.ApplicationID;
Trigger SLO Type = $EventAttributes.SLOType;
Trigger Value = $EventAttributes.TriggerValue;
Trigger Threshold = $EventAttributes.TriggerThreshold;
Trigger Element = $Entity.ElementID;
Event ID = $EventID;
```

### 16.3.6 Custom Metric Configuration

There are cases where additional instrumentation is needed based on your specialized requirements. Custom metrics allow you to instrument a class or method of your choice and receive performance metrics collected by the ADP agent.

To create a metric configuration, do the following:

1. From the Grid Control Home page, click **Targets**, then click **Middleware**. In the Related Links sections, select **Application Dependency and Performance**.
2. Click the **Configuration** tab, choose the configuration in which you are interested. Click **Custom Metric Configuration**.
3. In the right pane, click the **Create Custom Metric** button.
4. On the Custom Metric File page, choose whether to use an existing .xml file or a new file. If you choose a new file, the ADP Manager will create the new .xml file. Click **Continue**.
5. On the Custom Metric Configuration page, provide the following information:
  - Resource name is a monitored Weblogic domain or Oracle Application Server or WebSphere cell.  
You created a name when you configured ADP to monitor. The same name is used here during custom metric configuration.
  - Class name is the name of the implementation class in the code. You are required to enter a fully qualified class name.
  - Method name is the name of the implementation method in the code.

Once the custom metric configuration is complete, restart the monitored application server.

## 16.4 Exploring the Registration Tab

The managers perform complex mathematical modeling and statistical calculations with summarized data from all Java Agents.

Using the Registration tab, you can add, edit, and remove Managers configured to Enterprise Manager. By accessing ADP through Remote Method Invocation (RMI), you can manipulate all the managers configured to Enterprise Manager through a secured protocol.

### 16.4.1 Adding a New Manager (RMI Configuration)

The first time the Registration tab displays there are no managers in the Managers tree. To add a new manager, perform the following steps:

1. In the Registration tab, click the Managers node in the tree.
2. Type the new manager information in the Main Display window.
3. Decide whether to use Request Monitoring.

Request Monitoring provides end-to-end visibility into requests, localizes end-user performance problems to specific application deployments, and provides a platform for context-based drill down diagnostics.

When you select **Enable Request Monitoring**, ADP creates and sets up targets for collecting request performance data in Request Monitoring. If you do not select Enable Request Monitoring, the ADP manager is only registered in Enterprise Manager.

---

---

**Note:** The grayed out information represents configuration data for connecting to the ADP manager by way of a secure protocol, for example Key Store, Trust Store, and passwords. This information is extracted from the ADP manager by way of the RMI call.

---

---

4. If you enable request monitoring on an existing manager, click **Upload** to populate the manager configuration properties to the ADP target in Request Monitoring.
5. Click **Test Connect** to test the connection to the new manager. Should the test connection fail, this may be because the manager is not running or the manager is not yet installed.
6. Click **Add**.

Once the manager is added, the name of the manager will display in the Configuration tab under the Managers node in the tree.

### 16.4.2 Editing a Previously Configured Manager (RMI Configuration)

To add a previously configured manager, perform the following steps:

1. Click + (plus sign) next to the Managers node in the tree, then select the subnode for the manager you want to edit.
2. After you make changes to the manager information, click **Update**. This results in the manager entries in the Enterprise Manager repository to be updated with the new values.

If a manager is configured before using this Enterprise Manager configuration page, Enterprise Manager continues to keep the manager as a valid manager even though the manager may be down or permanently removed.

The list of managers is not refreshed.

### 16.4.3 Removing or Disabling a Previously Configured Manager

To remove a configured manager:

1. Click + (plus sign) next to the Managers node in the tree, then select the subnode for the manager you want to remove.
2. Click **Remove** in the main pane.

Deleting a manager from Enterprise Manager does not uninstall and remove the manager from the remote host where the manager is located and may be running. Remove only deletes the manager entry from the Enterprise Manager repository.

To disable a configured manager:

1. Click + (plus sign) next to the Managers node in the tree, then select the subnode for the manager you want to disable.
2. Deselect **Enable Request Monitoring**.
3. Click **Update**.

When you deselect the Enable Request Monitoring option, the manager settings are preserved. The UI displays these managers as disabled. There will not be any further information under the disabled manager in the tree.





---

---

## Performance Analytics

As ADP collects measurements, it processes incoming data and stores resulting information in the ADP embedded database. The ADP performance analytics feature then queries against this database to create statistical models and perform mathematical calculations.

This chapter includes the following performance analytics features:

- [Section 17.1, "Entity Performance Ranking"](#)
- [Section 17.2, "Performance Characterization"](#)
- [Section 17.3, "Memory Leak Detection"](#)
- [Section 17.4, "Drill Down - Bottleneck Analysis"](#)
- [Section 17.5, "Drill Out - Impact Analysis"](#)

### 17.1 Entity Performance Ranking

For Portal metrics, you can use ADP Entity Performance Ranking to accelerate bottleneck isolation.

Entity Performance Ranking becomes available when a node contains multiple children entities. For example, if the Labels node has multiple children entities, clicking on the Analysis tab of the node displays the Entity Performance Ranking.

ADP provides Entity Performance Ranking on Nodes with Children Entities.

### 17.2 Performance Characterization

Performance Characterization is a set of performance analytics provided by ADP. Use the Performance Characterization analytics to visualize actual performance of a monitored entity in a given time frame. The following is a list of Performance Characterization analytics provided by ADP.

A statistically significant amount of data is required before these graphs will appear. You will need at least 1 hour of continuous load for any graphs to appear and at least 5 hours of continuous load for all of the graphs to appear. Analysis requires that load metrics actually be present and zero points do not count for the purposes of this analysis. It may be helpful to increase your time frame manually until you encompass enough data points.

### 17.2.1 Multi-Point Performance/Load Regression

ADP automatically fits a multi-point line through the data set of performance and load measurements collected during a specific time frame. This feature allows you to quickly characterize the performance of an entity for a certain load level in real time.

ADP performs multi-point regression and fits a 3-point line through the data set.

As a performance analytical tool, the Multi-Point Performance/Load Regression is extremely useful. It enables you to characterize the performance of a system or a component for a certain load range. You can use this analysis to capture the performance baseline of a system with a specific infrastructure configuration such as deployed applications, application server configuration, network topology and hardware capacity and load their characteristics. This analysis can also be used for capacity planning. Since the regression illustrates the actual application performance for a specific configuration, this information is useful to determine how to expand computing capacity to best meet estimated demand.

### 17.2.2 Performance/Load Scattergram

This feature automatically plots the interceptions between arrival rate and response time for a specific monitored entity into a scattergram. This feature gives you a visual interpretation of actual performance (response time) under a range of loads (arrival rate) in real time.

ADP plots intersections of arrivals and response time metrics in a scattergram.

### 17.2.3 Time-Based Performance Distribution

This feature automatically plots the performance (response time) of a specific monitored entity during a time frame specified by you. The X axis is response time and Y axis is time on the clock. This plot gives you a visual representation of a monitored entity's performance over time.

Time-Based Performance Distribution can help you identify abnormal performance patterns.

### 17.2.4 Performance Histogram

This feature automatically creates a histogram with ten bins (performance ranges) and plots the occurrences of response time that falls in each bin. The X axis is the number of occurrences (frequency) and Y axis is the response time ranges. You can use this histogram to see the performance distribution for a specific monitored entity.

Performance Histogram can help you visualize performance distribution.

### 17.2.5 Time-Based Performance Trend

This feature automatically plots a trend line associated with the performance (response time) of a specific monitored entity during a time frame specified by you. The X axis is response time and Y axis is time on the clock. This trend line gives you a visual representation of a monitored entity's performance over time.

Time-based Performance Trend allows you to quickly identify abnormal performance patterns during a certain time frame.

## 17.3 Memory Leak Detection

ADP has a built-in analytic to detect performance memory leak.

To access Memory Leak Detection, select the JVM node under Resources. When the JVM node is selected, ADP analyzes memory usage patterns during the time frame specified. Two graphs, Memory Leak Rate and Memory Leak Status, display in the Main Display Window.

You can access Memory Leak Detection by clicking on the JVM node.

The Memory Leak Rate graph enables you to perform time-based trend analysis. The graph depicts the JVM heap memory growth rate in KB/minute for a given period of time. The Memory Leak Status graph enables you to quickly identify the presence of persistent memory leak.

**Tip:** Persistent JVM heap growth can result in a memory leak. ADP evaluates the memory growth rate continuously to determine leak status. The Memory Leak Status graph has two values (0 and 1). The value 0 means there is no leak. The value 1 means there may be a leak. When there is a severe leak, the value of the Memory Leak Status graph remains at 1.

## 17.4 Drill Down - Bottleneck Analysis

Finding performance bottlenecks using ADP is easy. Often, ADP guides you to the exact location of the bottleneck. There are some basic ways for performing drill down operations in ADP.

**Table 17-1** *Drill-Down Methods*

Operation	Description
Double-Clicking	Available on many views in ADP. Double-clicking triggers ADP to bring up information one logical level lower than what is currently displayed.
Right-Click Menu	Available on most views in ADP. From most views, selecting the Display Architecture View option initiates a diagnostic session. From the Architecture View, use the drill down option to go one level deeper.

Whenever possible, use the built-in Delay Analysis to isolate performance bottlenecks. Delay Analysis is available under the Process node as well as in the Architecture View. ADP Delay Analysis shows the delay contribution associated with a specific component in a process or an active call path to the overall delay.

You can also use Entity Performance Ranking to identify abnormally behaving components and performance bottlenecks.

Use the detailed performance data to further isolate the performance problem. This pop-up window contains the following information:

- Servlet Performance [Table]
- Servlet Response Time (ms) [Graph]
- Servlet Invocation Count [Graph]
- EJB Performance [Table]
- Top five EJB Response Time (ms) [Graph]
- Top five EJB Invocation Count [Graph]

Double-clicking on a specific row in this pop-up window brings up the Architecture View associated with the selected row.

Double-clicking on the slow performing component brings up the Architecture View with delay analysis.

**Tip:** Right-click and check the **Hide other arrows** option to show only arrows on the component which you have clicked on. Follow the slowest Inbound calls to the slowest next level component. Repeat this method to find the bottleneck.

Look at the Fan Out table first to see if any outbound calls are performing poorly. If a specific call is performing poorly, follow the call to the destination component and evaluate further.

Evaluate all the Fan Out calls and follow the slowest Fan Out call.

If Fan Out calls are not performing poorly, it is very likely the performance bottleneck exists within the currently selected component. Use the Fan In table to isolate the poorly performing calls. Double-clicking on a specific inbound call in the Fan In table brings up a pop-up window with detailed performance data associated with this inbound call.

Double-click on the slowest Fan In call to get detailed performance data.

**Tip:** Compare current delay contribution breakdown to baseline delay contribution breakdown to identify any abnormal behavior. This can be done using Comparative Views.

## 17.5 Drill Out - Impact Analysis

After performance bottleneck is identified, you can use the ADP unique drill out ability to determine the impact of the bottleneck. The impact analysis can be done in the Architecture View.

To start the drill out process:

1. Select the entity that has been identified as the bottleneck in the Architecture View
2. Right-click and select **Drill Out**. This brings the Architecture View one logical level higher.
3. Use a combination of Drill Out and Show in Context navigation techniques to determine the scope of impact.

To determine the impact of the performance bottleneck, refocus the Architecture View and the module that contains the class in context (white background color).

On this window you can:

- Check the **Hide other arrows** option in the right-click menu to show only arrows on the component you have clicked on and improve visual navigation.
- Use the Architecture View or the Fan In table to identify the upstream components that make inbound calls into the class. The impact analysis would follow these inbound calls to their origins to further determine the impact of the performance bottleneck.

For example, the blue background color of a class indicates this entity belongs to a module that is different from current view context.

This chapter includes the following export features:

- [Section 18.1, "Data Export Modes"](#)
- [Section 18.2, "ADP Export Configuration"](#)
- [Section 18.3, "Example of Exported Data for WebLogic"](#)

### 18.1 Data Export Modes

There are three different modes to export performance data collected by ADP to external databases and other persistence formats. These modes give you flexibility to choose the best way to extract performance data from ADP.

- [Section 18.1.1, "Export to File"](#)
- [Section 18.1.2, "Export to Database"](#)
- [Section 18.1.3, "Aggregation Export to File"](#)

#### 18.1.1 Export to File

In this mode, ADP exports its raw performance data as several CSV (comma separated value) files.

#### 18.1.2 Export to Database

In this mode, ADP exports its raw performance data as several ANSI SQL statements. These SQL statements allow you to create tables and insert data.

#### 18.1.3 Aggregation Export to File

In this mode, ADP exports its aggregated performance data after it's daily aggregation operation as several CSV files.

### 18.2 ADP Export Configuration

The following sections describe ADP export configuration:

- [Section 18.1, "Data Export Modes"](#)
- [Section 18.2, "ADP Export Configuration"](#)
- [Section 18.3, "Example of Exported Data for WebLogic"](#)

## 18.2.1 ADP Periodic Export Configuration

ADP stores real-time performance metrics in its internal data repository (Oracle database). If you want to store this data in your historical data repository, ADP provides automatic means for performance data export. You can control the frequency of export runs, the time when the export should run, and the time range of export data within a day.

### **Example 18–1 ADP Export Configuration**

```
# Setting for integrated export
AggregationManager.IntegratedExport = false
AggregationManager.ExportDataStartHour = 0
AggregationManager.ExportDataEndHour = 0
AggregationManager.ExportDataSetRangeInHour = 4
AggregationManager.ExportDataSetIntervalInHour = 1
AggregationManager.ExportDataSetDelay = 10000
AggregationManager.ExportStartTime = 0
AggregationManager.ExportEndTime = 0
AggregationManager.ExportFilePurgeTime = 10d
```

By default, automatic data export feature is disabled. To enable it, set `AggregationManager.IntegratedExport` parameter to true. A pair of parameters, `AggregationManager.ExportDataStartHour` and `AggregationManager.ExportDataEndHour`, indicates the time range within a day of the export performance data in which you are interested. By default it is 24 hours.

Parameter `AggregationManager.ExportDataSetRangeInHour` shows how much data is stored in each export file. By default it is 4 hours worth of data. This means that ADP will create multiple export data files with 4 hours worth of data.

To minimize export query impact on normal ADP performance data collection functionality, spread out lengthy data exporting queries. This is achieved by setting `AggregationManager.ExportDataSetIntervalInHour` parameter. By default it is 1 hour. This means that export query thread will be running every hour.

`AggregationManager.ExportDataSetDelay` defines cool down time for consecutive export queries. By default it is 10 seconds. This means that the next export query will happen not earlier than 10 seconds after the previous one.

`AggregationManager.ExportFilePurgeTime` indicates how many days the export data file will be available before ADP deletes it. By default it is 10 days. Current default values are optimal and this section should not be changed except from enabling the automatic export feature.

Automatic export function relies on the definition of the data to be exported by looking into `$CAMM_HOME/config/export.xml` file and exports performance data based on the rules defined in this file.

The output directory is specified in `export.xml` and will be overwritten on each export interval.

## 18.2.2 Manual Execution of Metric Export

The bin directory on the ADP manager contains scripts called `runExportMetric.sh/bat` and `runExportEvent.sh/bat` that export metrics and events (such as alerts) to CSV files, respectively.

Run `runExportMetric.sh` like:

```
./runExportMetric.sh <path to export.xml configuration> <start time> <end time>
```

For example:

```
C:\oracle\em11g\bin\runExportMetric.bat c:\oracle\em11g\config\export.xml
"4/1/09 16:06:00" "4/1/09 16:36:00"
```

The start time and end time are in the machine's local time zone. The output exported csv files' timestamps will be in UTC/GMT.

## 18.2.3 export.xml File

The export.xml file contains all the directives and filters required to export performance metrics and events. This file is used by the ADP Integrated Automatic export feature as well as manual export scripts.

### Example 18–2 Contents of export.xml File

```
<?xml version="1.0" encoding="UTF-8"?>
<export xmlns="http://www.acsera.com/ns/export" verbose="true" exportMetric="true"
exportEvent="true" metricDataGrain="180s" exportFullMetric="true">
  <!--
  <output type="jdbc" convertTimeFormat="true"
  arguments="access,metric,sun.jdbc.odbc.JdbcOdbcDriver,jdbc:odbc:acsera"/>
  -->
  <output type="file" convertTimeFormat="false"
  arguments="/home/acsera/acsera/export,metric"/>
  <entityTypes exportAllTypes="false">
    <entityType name="BEA.ProcessNode"/>
    <entityType name="J2EE.Dispatcher"/>
    <entityType name="J2EE.JDBC.ConnectionPool"/>
    <entityType name="J2EE.JVM"/>
  </entityTypes>
  <!--extra filter -->
  <!--
  <filters>
  <filter key="containerID" values="cgServer"/>
  </filters>
  -->
  <!-- don't modify this -->
  <columns>
    <column header="Timestamp" type="Timestamp"/>
    <column header="EntityID" type="EntityID"/>
    <column header="Application" type="Entity" key="applicationID" default="" />
  </columns>
</export>
```

The following tables explain the various attributes.

**Table 18–1 Attributes on <export>**

Attribute	Description
exportMetric	true/false, whether to export performance metrics
exportEvent	true/false, whether to export SLO events
metricDataGrain	60s, 180s, or 1800s, the metric aggregation tier to export. Note that the population of the 180s tier will be delayed by 1.25 hours and the 1800s tier will be delayed by 7 hours compared to the 60s tier.
exportFullMetric	true/false, whether to include sum, count, min, and max metrics or not.

**Table 18–2 Attributes on <output>**

Attribute	Description
type	file/jdbc, whether to output to csv files or to write data to another database using JDBC. In the case of JDBC output, the necessary tables will be created automatically by the export mechanism.
convertTimeFormat	true/false, whether to convert the metric timestamp to human readable format (in UTC/GMT). If false, the timestamp will be a long integer. Provide the arguments (jdbc) in a comma separated list.
arguments	See <a href="#">Table 18–3</a> and <a href="#">Table 18–4</a> for details.

**Table 18–3 Value of the "arguments" Attribute of the <output> Element When the "type" Attribute Is "JDBC"**

Attribute	Description
First Parameter	Database type, an arbitrary string
Second Parameter	Table prefix name, should always be metric
Third Parameter	Fully qualified JDBC driver class
Fourth Parameter	JDBC URL

**Table 18–4 Value of the "arguments" Attribute of the <output> Element When the "type" Is "file"**

Argument	Description
First Parameter	CSV file output directory (this directory will be created, existing files will be overwritten)
Second Parameter	Table prefix name, should always be <b>metric</b>

**Table 18–5 Attribute of the <entityTypes> Element**

Attribute	Description
exportAllTypes	true/false, whether to output all entity types or only the ones specified in <entityType> elements.

**Table 18–6 Attribute of the <entityType> Element**

Attribute	Description
name	Name of the entityType to include in the export if exportAllTypes is set to false

## 18.3 Example of Exported Data for WebLogic

The tables in this section describe the fields in the various export files.

- [Table 18–7, "Export File Name: metricBEA\\_ChannelInstance.csv"](#)
- [Table 18–8, "Export File Name: metricBEA\\_ProcessType.csv"](#)
- [Table 18–9, "Export File Name: metricBEA\\_TimerEventGenerator.csv"](#)
- [Table 18–10, "Export File Name: metricJ2EE\\_Dispatcher.csv"](#)
- [Table 18–11, "Export File Name: metricJ2EE\\_EJB\\_Entity.csv"](#)
- [Table 18–12, "Export File Name: metricJ2EE\\_EJB\\_Stateless.csv"](#)
- [Table 18–13, "Export File Name: metricJ2EE\\_JDBC\\_ConnectionPool.csv"](#)



- Table 18–14, " Export File Name: metricJ2EE\_JMS\_Destination.csv"
- Table 18–15, " Export File Name: metricJ2EE\_JMS\_Service.csv"
- Table 18–16, " Export File Name: metricJ2EE\_JVM.csv"
- Table 18–17, " Export File Name: metricJ2EE\_Server.csv"
- Table 18–18, " Export File Name: metricJ2EE\_Servlet.csv"

**Table 18–7** Export File Name: *metricBEA\_ChannelInstance.csv*

Field	Description
StartTime	Clock time (long) at data insertion
EntityID	ADP unique identifier for the monitored entity
elementID	Fully qualified name of the channel
channelID	Fully qualified name of the channel
serviceID	URL of the service / JPD
infrastructureID	ID of the monitoring infrastructure. Oracle is the only value at this time.
containerID	Name of the Oracle WebLogic Server instance
nodeID	Name of the physical machine
domainID	Name of the Oracle WebLogic domain
displayNameID	Display name
resourceID	Name of the monitored resource as configured by the user
entityTypeID	Type of the monitored entity
Metric.J2EE.ChannelInstance.MessageCount	JMX metric
Metric.J2EE.ChannelInstance.DeadMessageCount	JMX metric

**Table 18–8** Export File Name: *metricBEA\_ProcessType.csv*

Field	Description
StartTime	Clock time (long) at data insertion
EntityID	ADP unique identifier for the monitored entity
elementID	Implementation class name
processID	Display name of the process
serviceID	URL of the service / JPD
projectID	Name of Workshop project / web application module
containerID	Name of the Oracle WebLogic Server instance
nodeID	Name of the physical machine
domainID	Name of the Oracle WebLogic domain
entityTypeID	Type of the monitored entity
applicationID	Name of the Application
infrastructureID	ID of the monitoring infrastructure. Oracle is the only value at this time.
deploymentID	Unique ID used by Oracle WebLogic to track application deployments
resourceID	Name of the monitored resource as configured by the user

**Table 18–8 (Cont.) Export File Name: metricBEA\_ProcessType.csv**

Field	Description
displayNameID	Display name
controlContainerID	Implementation class name of the process
Metric.J2EE.ProcessType.Arrivals	Instrumentation metric -- number of arrivals
Metric.J2EE.ProcessType.Aborts	Instrumentation metric -- number of aborts
Metric.J2EE.ProcessType.ElapsedTime	Instrumentation metric -- average elapsed time
Metric.J2EE.ProcessType.Active	Instrumentation metric -- number of active requests
Metric.J2EE.ProcessType.VisitCount	Instrumentation metric -- number of completed requests
Metric.J2EE.ProcessType.Exceptions	Instrumentation metric -- number of exceptions

**Table 18–9 Export File Name: metricBEA\_TimerEventGenerator.csv**

Field	Description
StartTime	Clock time (long) at data insertion
EntityID	ADP unique identifier for the monitored entity
elementID	Fully qualified name of the channel
channelID	Fully qualified name of the channel
infrastructureID	ID of the monitoring infrastructure. Oracle is the only value at this time.
channelTxID	Fully qualified name of the channel
domainID	Name of the Oracle WebLogic domain
displayNameID	Display name
resourceID	Name of the monitored resource as configured by the user
entityTypeID	Type of the monitored entity
Metric.J2EE.TimerEventGenerator.MessageCount	JMX metric
Metric.J2EE.TimerEventGenerator.ErrorCount	JMX metric

**Table 18–10 Export File Name: metricJ2EE\_Dispatcher.csv**

Field	Description
StartTime	Clock time (long) at data insertion
EntityID	ADP unique identifier for the monitored entity
elementID	Fully qualified name of the Execute Queue
infrastructureID	ID of the monitoring infrastructure. Oracle is the only value at this time.
containerID	Name of the Oracle WebLogic Server instance
nodeID	Name of the physical machine
domainID	Name of the Oracle WebLogic domain
executeQueueID	Name of the Execute Queue as configured by user
displayNameID	Display name
resourceID	Name of the monitored resource as configured by the user

**Table 18–10 (Cont.) Export File Name: metricJ2EE\_Dispatcher.csv**

Field	Description
entityTypeID	Type of the monitored entity
Metric.J2EE.Dispatcher.ServicedRequestsTotalCount	JMX metric
Metric.J2EE.Dispatcher.IdleThreads	JMX metric
Metric.J2EE.Dispatcher.PendingRequests	JMX metric

**Table 18–11 Export File Name: metricJ2EE\_EJB\_Entity.csv**

Field	Description
StartTime	Clock time (long) at data insertion
EntityID	ADP unique identifier for the monitored entity
methodID	Name of the EJB method executed
domainID	Name of the Oracle WebLogic domain
entityTypeID	Type of the monitored entity
infrastructureID	ID of the monitoring infrastructure. Oracle is the only value at this time.
ejbID	Name of the EJB
webApplicationID	Name of the web module
displayNameID	Display name
controlContainerTypeID	Identifies type of control
elementID	Implementation class name
processID	Display name of the process
serviceID	URL of the service / JPD
projectID	Name of Workshop project / web application module
containerID	Name of the Oracle WebLogic Server instance
nodeID	Name of the physical machine
ejbComponentID	Name of Java EE component that contains this EJB
applicationID	Name of the Application
resourceID	Name of the monitored resource as configured by the user
controlContainerID	Implementation class name of the process
Metric.J2EE.EJB.Entity.Locking.LockManagerAccessCount	JMX metric
Metric.J2EE.EJB.Entity.ResponseTime	Instrumentation metric -- response time
Metric.J2EE.EJB.Entity.Cache.BeansCurrentCount	JMX metric
Metric.J2EE.EJB.Entity.Cache.AccessCount	JMX metric
Metric.J2EE.EJB.Entity.Pool.WaiterCurrentCount	JMX metric
Metric.J2EE.EJB.Entity.Transaction.CommittedTotalCount	JMX metric
Metric.J2EE.EJB.Entity.Locking.WaiterTotalCount	JMX metric
Metric.J2EE.EJB.Entity.Transaction.TimedOutTotalCount	JMX metric

**Table 18–11 (Cont.) Export File Name: metricJ2EE\_EJB\_Entity.csv**

Field	Description
Metric.J2EE.EJB.Entity.Cache.HitCount	JMX metric
Metric.J2EE.EJB.Entity.Locking.WaiterCurrentCount	JMX metric
Metric.J2EE.EJB.Entity.Pool.IdleCount	JMX metric
Metric.J2EE.EJB.Entity.Locking.EntriesCurrentCount	JMX metric
Metric.J2EE.EJB.Entity.VisitCount	Instrumentation metric -- invocation count
Metric.J2EE.EJB.Entity.Locking.TimeoutTotalCount	JMX metric
Metric.J2EE.EJB.Entity.Pool.InUseCount	JMX metric
Metric.J2EE.EJB.Entity.Pool.WaiterTotalCount	JMX metric
Metric.J2EE.EJB.Entity.Pool.TimeoutTotalCount	JMX metric
Metric.J2EE.EJB.Entity.Transaction.RolledBackTotalCount	JMX metric
Metric.J2EE.EJB.Entity.Cache.ActivationCount	JMX metric
Metric.J2EE.EJB.Entity.Cache.PassivationCount	JMX metric

**Table 18–12 Export File Name: metricJ2EE\_EJB\_Stateless.csv**

Field	Description
StartTime	Clock time (long) at data insertion
EntityID	ADP unique identifier for the monitored entity
elementID	Implementation class name
projectID	Name of Workshop project / web application module
nodeID	Name of the physical machine
containerID	Name of the Oracle WebLogic Server instance
domainID	Name of the Oracle WebLogic domain
ejbComponentID	Name of Java EE component that contains this EJB
entityTypeID	Type of the monitored entity
applicationID	Name of the Application
infrastructureID	ID of the monitoring infrastructure. Oracle is the only value at this time.
ejbID	Name of the EJB
resourceID	Name of the monitored resource as configured by the user
displayNameID	Display name
Metric.J2EE.EJB.Stateless.Transaction.TimedOutTotalCount	JMX metric
Metric.J2EE.EJB.Stateless.Pool.WaiterTotalCount	JMX metric
Metric.J2EE.EJB.Stateless.Pool.InUseCount	JMX metric
Metric.J2EE.EJB.Stateless.Transaction.CommittedTotalCount	JMX metric

**Table 18–12 (Cont.) Export File Name: metricJ2EE\_EJB\_Stateless.csv**

Field	Description
Metric.J2EE.EJB.Stateless.Transaction.RolledBackTotalCount	JMX metric
Metric.J2EE.EJB.Stateless.Pool.IdleCount	JMX metric
Metric.J2EE.EJB.Stateless.Pool.TimeoutTotalCount	JMX metric

**Table 18–13 Export File Name: metricJ2EE\_JDBC\_ConnectionPool.csv**

Field	Description
StartTime	Clock time (long) at data insertion
EntityID	ADP unique identifier for the monitored entity
elementID	Name of JDBC connection pool
infrastructureID	ID of the monitoring infrastructure. Oracle is the only value at this time.
containerID	Name of the Oracle WebLogic Server instance
nodeID	Name of the physical machine
domainID	Name of the Oracle WebLogic domain
displayNameID	Display name
resourceID	Name of the monitored resource as configured by the user
entityTypeID	Type of the monitored entity
Metric.J2EE.JDBC.ConnectionPool.WaitingForConnectionCurrentCount	JMX metric
Metric.J2EE.JDBC.ConnectionPool.WaitingForConnectionHighCount	JMX metric
Metric.J2EE.JDBC.ConnectionPool.ActiveConnectionsHighCount	JMX metric
Metric.J2EE.JDBC.ConnectionPool.ActiveConnectionsCurrentCount	JMX metric
Metric.J2EE.JDBC.ConnectionPool.FailuresToReconnectCount	JMX metric
Metric.J2EE.JDBC.ConnectionPool.WaitSecondsHighCount	JMX metric
Metric.J2EE.JDBC.ConnectionPool.ConnectionDelayTime	JMX metric

**Table 18–14 Export File Name: metricJ2EE\_JMS\_Destination.csv**

Field	Description
StartTime	Clock time (long) at data insertion
EntityID	ADP unique identifier for the monitored entity
elementID	Name of JMS destination
nodeID	Name of the physical machine
containerID	Name of the Oracle WebLogic Server instance
jmsServerRuntimeID	Name of the JMS server
domainID	Name of the Oracle WebLogic domain
jmsDistributedQueueMemberID	Name of the JMS distributed queue member
entityTypeID	Type of the monitored entity

**Table 18–14 (Cont.) Export File Name: metricJ2EE\_JMS\_Destination.csv**

Field	Description
jmsQueueID	Name of the JMS queue
jmsRuntimeID	Name of the JMS service
infrastructureID	ID of the monitoring infrastructure. Oracle is the only value at this time.
jmsDistributedQueueID	Name of the JMS distributed queue
resourceID	Name of the monitored resource as configured by the user.
displayNameID	Display name
Metric.J2EE.JMS.Destination.ConsumersCurrentCount	JMX metric
Metric.J2EE.JMS.Destination.BytesCurrentCount	JMX metric
Metric.J2EE.JMS.Destination.MessagesPendingCount	JMX metric
Metric.J2EE.JMS.Destination.BytesThresholdTime	JMX metric
Metric.J2EE.JMS.Destination.MessagesHighCount	JMX metric
Metric.J2EE.JMS.Destination.BytesReceivedCount	JMX metric
Metric.J2EE.JMS.Destination.MessagesReceivedCount	JMX metric
Metric.J2EE.JMS.Destination.BytesHighCount	JMX metric
Metric.J2EE.JMS.Destination.MessagesCurrentCount	JMX metric
Metric.J2EE.JMS.Destination.ConsumersTotalCount	JMX metric
Metric.J2EE.JMS.Destination.ConsumersHighCount	JMX metric
Metric.J2EE.JMS.Destination.BytesPendingCount	JMX metric

**Table 18–15 Export File Name: metricJ2EE\_JMS\_Service.csv**

Field	Description
StartTime	Clock time (long) at data insertion
EntityID	ADP unique identifier for the monitored entity
elementID	Name of the JMS service
infrastructureID	ID of the monitoring infrastructure. Oracle is the only value at this time.
containerID	Name of the Oracle WebLogic Server instance
nodeID	Name of the physical machine
domainID	Name of the Oracle WebLogic domain
displayNameID	Display name
resourceID	Name of the monitored resource as configured by the user
entityTypeID	Type of the monitored entity
jmsRuntimeID	Name of the JMS service
Metric.J2EE.JMS.Service.ConnectionsHighCount	JMX metric
Metric.J2EE.JMS.Service.ConnectionsCurrentCount	JMX metric
Metric.J2EE.JMS.Service.JMSServersCurrentCount	JMX metric

**Table 18–15 (Cont.) Export File Name: metricJ2EE\_JMS\_Service.csv**

Field	Description
Metric.J2EE.JMS.Service.JMSServersHighCount	JMX metric
Metric.J2EE.JMS.Service.ConnectionsTotalCount	JMX metric
Metric.J2EE.JMS.Service.JMSServersTotalCount	JMX metric

**Table 18–16 Export File Name: metricJ2EE\_JVM.csv**

Field	Description
StartTime	Clock time (long) at data insertion
EntityID	ADP unique identifier for the monitored entity
elementID	Name of the JVM
infrastructureID	ID of the monitoring infrastructure. Oracle is the only value at this time.
containerID	Name of the Oracle WebLogic Server instance
nodeID	Name of the physical machine
domainID	Name of the Oracle WebLogic domain
displayNameID	Display name
resourceID	Name of the monitored resource as configured by the user
entityTypeID	Type of the monitored entity
Metric.J2EE.JVM.JRockit.HeapSizeCurrent	JMX metric
Metric.J2EE.JVM.JRockit.HeapFreeCurrent	JMX metric
Metric.J2EE.JVM.JRockit.PhysMemTotal	JMX metric
Metric.J2EE.JVM.JRockit.PhysMemUsed	JMX metric
Metric.J2EE.JVM.JRockit.GarbageCollectionCountTotal	JMX metric
Metric.J2EE.JVM.JRockit.GarbageCollectionTimeTotal	JMX metric
Metric.J2EE.JVM.HeapFreeCurrent	JMX metric
Metric.J2EE.JVM.JRockit.PhysMemFree	JMX metric
Metric.J2EE.JVM.JRockit.NursurySizeTotal	JMX metric
Metric.J2EE.JVM.JRockit.ActiveDaemonThreads	JMX metric
Metric.J2EE.JVM.JRockit.ActiveThreads	JMX metric
Metric.J2EE.JVM.HeapSizeCurrent	JMX metric
Metric.J2EE.JVM.JRockit.HeapUsedCurrent	JMX metric

**Table 18–17 Export File Name: metricJ2EE\_Server.csv**

Field	Description
StartTime	Clock time (long) at data insertion
EntityID	ADP unique identifier for the monitored entity
elementID	Name of the Java EE server instance
infrastructureID	ID of the monitoring infrastructure. Oracle is the only value at this time.

**Table 18–17 (Cont.) Export File Name: metricJ2EE\_Server.csv**

Field	Description
containerID	Name of the Java EE server instance
nodeID	Name of the physical machine
domainID	Name of the Oracle WebLogic domain
displayNameID	Display name
resourceID	Name of the monitored resource as configured by the user
entityTypeID	Type of the monitored entity
Metric.J2EE.Server.RestartsTotalCount	JMX metric

**Table 18–18 Export File Name: metricJ2EE\_Servlet.csv**

Field	Description
StartTime	Clock time (long) at data insertion
EntityID	ADP unique identifier for the monitored entity
elementID	Name of the servlet implementation class
applicationID	Name of the Application
infrastructureID	ID of the monitoring infrastructure. Oracle is the only value at this time.
containerID	Name of the Oracle WebLogic Server instance
nodeID	Name of the physical machine
domainID	Name of the Oracle WebLogic domain
servletID	Name of the servlet
webApplicationID	Name of the web module
displayNameID	Display name
resourceID	Name of the monitored resource as configured by the user
entityTypeID	Type of the monitored entity
Metric.J2EE.Servlet.InvocationTotalCount	JMX metric
Metric.J2EE.Servlet.ExecutionTimeAverage	JMX metric

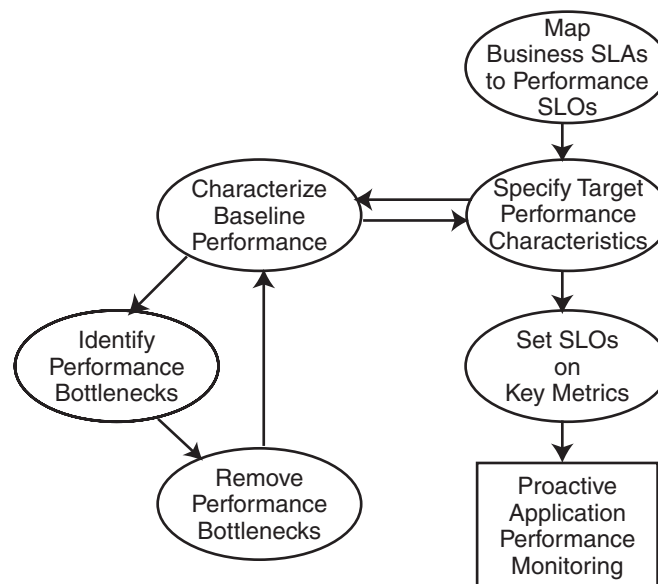


## ADP Methodology

Enterprise Manager's Application Dependency and Performance (ADP) capabilities automatically select performance metrics and track contextual relationships for various applications. The methodology focuses on other important activities to allow you to setup and maintain an effective application performance monitoring environment.

These activities include the following:

**Figure 19–1 Steps of ADP Methodology**



This methodology describes a series of steps for users to establish and maintain a proactive application performance monitoring environment leveraging Enterprise Manager's Application Dependency and Performance capabilities. [Figure 19–1](#) illustrates these steps in a sequential order.

Methodology steps:

1. Map business SLAs to performance SLOs.

The process of using agreed business SLAs to determine the value of performance SLOs.

2. Specify target performance characteristics.

Specify the ideal application performance characteristics using performance SLOs identified in step 1.

3. Characterize baseline performance.
4. Identify performance bottlenecks.
5. Remove performance bottlenecks.

Steps 3, 4, and 5 should be grouped together to form a process of incremental performance improvement. Iterations of this process may be required to improve the application performance to meet the performance target as specified in step 2.

6. Set SLOs on key metrics.

Once application performance reaches the targeted goal, you need to set performance SLOs on key metrics to establish a proactive monitoring environment. This environment provides you with warnings when key performance metrics start to report abnormalities. These warnings enable you to proactively solve potential problems before they begin to impact business.

This chapter explains the following activities in more detail:

- [Section 19.1, "ADP Methodology Activities"](#)
- [Section 19.2, "Map Business SLAs to Performance SLOs"](#)
- [Section 19.3, "Characterize Baseline Performance"](#)
- [Section 19.4, "Identify Performance Bottlenecks"](#)
- [Section 19.5, "Set SLOs on Key Metrics"](#)

## 19.1 ADP Methodology Activities

The ADP methodology activities include the following:

- [Section 19.1.1, "Map Business SLAs to Performance SLOs,"](#)
- [Section 19.1.2, "Specify Target Performance Characteristics"](#)
- [Section 19.1.3, "Performance Improvement Process"](#)

### 19.1.1 Map Business SLAs to Performance SLOs

To successfully setup a proactive application performance monitoring environment, the first step is to map a set of business objectives to a set of performance thresholds for you to monitor. These business objectives are often referred to as business service level agreements (SLAs). These business SLAs provide the basic application performance requirements at a high level. As such, mapping these high level SLAs to low level performance thresholds is often a very difficult activity to do well.

Using tools that only measure performance at technology levels (EJB, JSP, servlet, portlet, SQL calls, and so on) to perform this type of activity continues to be very difficult as the correlations between low-level metrics and high-level objectives are often fuzzy at best. Consequently, the mapping activity is considered by many as an art rather than a science.

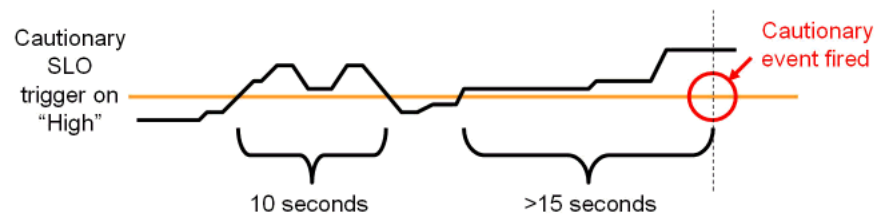
By measuring performance at both technology and functional levels, Enterprise Manager makes this mapping activity significantly less complicated. Since functional metrics measure performance for high level constructs such as business processes or portal desktops, mapping business SLAs directly to performance SLOs (Service Level Objectives) is straightforward.

## 19.1.2 Specify Target Performance Characteristics

Defining the target performance characteristics for the monitored applications is the next step after mapping business SLAs to performance SLOs. Since these SLOs represent absolute minimal performance requirements for these applications, using these *violation* thresholds as target performance characteristics makes little sense. Instead, you need to define what performance range is acceptable for normal operation and when to send out cautionary alerts for abnormal activities.

For some applications, it may be sufficient to just specify a set of *cautionary* performance thresholds. Application performance monitoring tools, such as ADP, will send out cautionary alerts if these thresholds have been breached. Since these thresholds are cautionary, it may be acceptable to have a few violations before an alert is sent out. By defining the minimal violation duration, you can minimize the number of duplicate alerts generated. [Figure 19–2](#) illustrates this concept.

**Figure 19–2 Control Number of Alerts**



In [Figure 19–2](#), the minimal violation duration is defined to be 15 seconds. So if the cautionary state does not persist for more than 15 seconds, no cautionary alert would be fired.

For other applications, it is necessary to define both a high and low performance thresholds. Having both thresholds would effectively define a normal range of operation for these applications. With ADP, both high and low triggers can be set for any SLO.

With a set of clearly defined target performance characteristics, you are able to determine how much performance tuning is needed to achieve the ideal performance range. You will also have a set of cautionary performance thresholds to enable a proactive application performance monitoring environment to be established.

## 19.1.3 Performance Improvement Process

The activities explained in this section should be grouped together as a single performance improvement process. This process would start with characterizing the baseline performance of our application, move on to identifying performance bottlenecks, and finish with removing performance bottlenecks. We would continue to perform these activities in iterations until the performance of our application meets the target characteristics.

### 19.1.3.1 Characterize Baseline Performance

Once the specification of the target performance characteristics is completed, the next activity is to capture the performance baseline for our application. The performance baseline will be compared with the set of target performance characteristics to determine if further performance improvement is needed. If so, you will improve application performance iteratively through the next two steps until the performance meets the target characteristics.

### 19.1.3.2 Identify Performance Bottlenecks

To identify performance bottlenecks, you must first isolate performance abnormalities in your performance baseline. Once you isolate a performance abnormality, you need to determine if this issue is a localized occurrence or a systematic problem. By using monitoring and diagnostic tools available to you, you can perform the analysis needed to identify the cause of the performance bottleneck.

### 19.1.3.3 Remove Performance Bottlenecks

Once these performance bottlenecks are identified, you need to determine how to remove them. The strategies for bottleneck removal vary by cause. A few examples follow:

- If the cause of the bottleneck is an application defect, the strategy would involve the application development team.
- If the bottleneck is caused by a configuration problem, you would request assistance from system administrators.
- If the bottleneck is in the application server or framework, you would seek help from those vendors.

The following is a list of possible bottleneck removal activities:

- Change application code to fix defects
- Modify environment setting to fix configuration problem
- Install patches to fix software defects
- Replace defective hardware
- Upgrade network infrastructure
- Add computing resources
- Remove resource hogging programs
- Tune back-end connectivity and response time

As you can see from the list, the Remove Performance Bottlenecks activity varies widely by cause. Correctly and quickly finding the appropriate groups to help resolve performance bottlenecks is the key for success for this activity. Once the performance bottleneck removal is completed, you must redo the Characterize Baseline Performance activity to confirm the fix implemented indeed improved performance.

### 19.1.3.4 Set SLOs on Key Metrics

In addition to setting application specific performance thresholds, it is also important to set performance thresholds on key system metrics and on some selective component metrics. Setting these thresholds will help you establish an early warning system and alert you to smaller issues before they manifest into big production problems.

Setting SLOs on key system metrics involves some basic understanding of how the system behaves under load. If the system becomes unstable or performs poorly when it runs out of free JDBC connections or idle ExecuteQueue threads, these system metrics should be monitored.

To determine which system metrics to monitor, it is critical to figure out the correlations between overall system performance and specific system metrics. You would use this information to decide which of the system metrics to monitor. Once appropriate system metrics are identified, you will then determine the performance

range for normal operation and figure out the cautionary as well as violation thresholds.

While it is fairly straightforward to determine which system metrics to monitor and what system metric performance thresholds to set, setting SLOs on key component metrics is significantly more difficult. In theory, you can assume performance degradation at the component level would negatively impact application level performance. However, this assumption may not accurately reflect reality.

To predict application performance by monitoring component level performance metrics, there must be a very strong correlation between the performance of a specific component and that of the application. Sometimes, a drop in performance in one component is compensated by a jump in performance in another. These performance changes in opposite directions at the component level would essentially result in little change at the application level. Therefore, you must be careful not to draw conclusions by monitoring the performance of a few components unless there are strong correlations.

The last task to perform is to associate various actions and responses for various threshold violations. Once these associations are completed, you can begin to use your proactive application performance monitoring environment.

## 19.2 Map Business SLAs to Performance SLOs

One of the primary reasons companies purchase solutions to establish proactive application performance monitoring is the demand to meet business SLAs. Business SLAs for enterprise applications are a set of service level expectations defined by internal or external customers. In most cases, these business SLAs are defined at such a high level, they are not useful for setting thresholds in application performance monitoring tools.

As a result, the process of mapping business SLAs to performance SLOs is extremely important for companies to meet the service requirements set forth by their customers. Since Enterprise Manager monitors performance at both functional and technology levels, it is easy to perform this mapping exercise.

In this section, our example explains how to use Enterprise Manager's Application Dependency and Performance features to determine the proper performance threshold values for a set of business SLAs.

In the example, we were given the following high-level business SLAs:

**Table 19–1 Example - Guidelines for Business SLAs**

Business SLA	SLA Requirement
Fast customer self-service portal.	On average, pages in customer self-service portal should load within 2 seconds. This SLA must be fulfilled 99% of the time.
Customer service representative portal must be as fast as mainframe system.	All pages in customer service rep. portal must load within 6 seconds. This SLA must be fulfilled 99.9% of the time.
Fast to schedule a service call.	On average, scheduling a service call should take less than 30 seconds. This SLA must be fulfilled 99.99% of the time.

Let's map the first business SLA to a performance SLO. This SLA requirement states that the average response time for customer self-service portal (desktop) should be less than 2 seconds. In Enterprise Manager, we would set a high-level performance SLO at the desktop. Using the hierarchy in the ADP UI, you would select the *customer* desktop and right-click to set the SLO.

Because ADP monitors performance at both functional and technology levels, you can directly translate business SLAs to SLOs on functional metrics. In our example, it is the response time for portal desktop *customer*. For our example, we would proceed to set a violation SLO and a warning SLO.

We can calculate how often violations occur to figure out whether or not our current system is able to meet the SLA requirement 99% of the time. With Enterprise Manager, we can see whether there are any obvious violations. If there are any violations or close calls, we should confirm by examining actual data. If we have data for at least 24 hours, we would use the export capability within Enterprise Manager Application Dependency and Performance to prepare raw data for this calculation.

For the other two business SLAs, we would set performance SLOs on the appropriate metrics.

## 19.3 Characterize Baseline Performance

Also known as performance base-lining, characterize baseline performance involves a set of activities to capture the baseline performance of a system under specific level of load. For example, you can measure the baseline performance of a portal application deployed to a WebLogic cluster.

For example, you can display the performance data during the first four hours of a load test. The number of active sessions grows at a steady pace for the first ninety minutes. Eventually, the number of active sessions stays at approximately seven hundred as the number of new and expiring sessions reach an equilibrium.

Visualize the portal performance following a typical pattern of slow performance initially and gradually reaching a steady state. The initial slow performance is expected as the application server load components into memory and populates data into its caching mechanism. The performance improves gradually and reaches a steady state after approximately thirty minutes. The performance pattern during this initial thirty-minute period can be characterized as *startup performance during increasing load*. After thirty minutes, performance of the portal application stabilizes.

Enterprise Manager's ability to quickly establish an application performance monitoring environment allows you to carry out *characterize baseline performance* painlessly. Because Enterprise Manager is able to monitor at cluster level as well as at individual server level, it can characterize performance for the entire cluster or individual servers.

By verifying that a less loaded server has lower resource utilization and faster performance, you can draw the following observations about the performance characteristics of a portal application running on this environment:

- Since Server A's resource utilization is near maximum, we can use the load on that server as the maximum limit for individual servers. We can calculate individual server maximum load limit by using the load metric provided by Enterprise Manager's Application Dependency and Performance features.
- We should examine the load balancing algorithm and the configuration of the load balancer.

Comparing load and resource usage of two servers in a cluster confirms resource usage is inversely correlated to the load.

---

---

**Note:** This is a very basic performance characterization of an individual server. Performance of a multi-server cluster cannot be calculated by multiplying performance characteristics of individual servers because of the overhead involved with a clustered configuration. True cluster level performance must be measured with application performance monitoring tools like Enterprise Manager.

---

---

## 19.4 Identify Performance Bottlenecks

Enterprise Manager can also be used to quickly identify performance bottlenecks in QA, staging, and production settings. You can use the hierarchical model within Enterprise Manager's Application Dependency and Performance feature to identify an application performance bottleneck. Furthermore, you can use Enterprise Manager to track down an application performance problem caused by resource starvation.

### 19.4.1 System Level Performance

Since the performance problem seems to affect all components in the same way, we should suspect there is some type of performance degradation at the system level. To view system level performance data, we would look under the *Resources* hierarchy. Performance metrics under the Resources hierarchy provides the raw data for us to perform correlation analysis. This type of analysis is needed to determine whether resource starvation is the cause of the performance slowdown.

Graphs can reveal some interesting patterns.

- OS Agent Abnormalities

We noticed a sudden drop in CPU utilization and a sudden increase in disk utilization during the time period in question. This pattern indicates a large amount of virtual memory paging activities on this machine. Memory paging to disk is extremely expensive and slows down request processing as indicated by lowered CPU utilization. To understand why page is occurring, we will take a look at performance metrics on the JVM.

- JVM Heap Size

We noticed that for the initial twelve hours of this example, both total JVM heap size and free JVM heap size grew at a steady pace. The growth of the total JVM heap size stopped at 512 MB - an expected behavior since we configured WebLogic to have a maximum heap size of 512 MB. While we expect the free JVM heap size to stop growing after total heap size reached 512 MB, the free JVM heap size actually starts to drop. Combining this information with some previously obtained information such as no sudden increase in load, we can conclude that there is a high likelihood of a memory leak.

This abnormal consumption of memory caused the total JVM heap to reach its pre-defined maximum. It is also very likely this memory leak caused the increase in virtual memory paging activities and corresponding reduction in CPU utilization. This reduction in CPU utilization impacts the response time for all components running in this machine including JDBC Connections.

- Memory Leak

We were able to identify a memory leak in the WebLogic JVM gradually caused resource starvation and eventually impacts application performance. In order to further diagnose this problem, a deep-level memory profiling tool is required to understand memory usage of the JVM.

## 19.5 Set SLOs on Key Metrics

This step in the Oracle Methodology allows you to proactively monitor key system metrics to avoid catastrophic failures such as server hangs (non-responsive), server crashes, cluster hangs, and more. The ability to recognize signs leading up to these catastrophic failures is a must to maintain quality of service for your WebLogic infrastructure.

You can proactively set thresholds and actions for key WebLogic system metrics. [Table 19–2](#) lists the key system metrics for the WebLogic Platform:

**Table 19–2 List of Key System Metrics for WebLogic**

Key System Metric	Reason to Monitor
ExecuteQueue Idle Thread Count	Running out of ExecuteQueue threads is often a precursor to application server hangs (non-responsive). In some severe cases, when the application server runs out of ExecuteQueue threads, all of its operations would stop working.
ExecuteQueue Pending Request Count	A steady increase in the number of ExecuteQueue pending requests is also a precursor to server hangs. This metric is inversely correlated with the ExecuteQueue Idle Thread Count metric.
Total JVM Heap Size	There are two reasons to monitor this metric: <ol style="list-style-type: none"> <li>1. If total JVM heap size grows to predefined maximum, a cautionary event should be fired notifying the administrator.</li> <li>2. If total JVM heap size suddenly drops to 0, this may be an indication of a JVM crash or a non-operational application server.</li> </ol>
Free JVM Heap Size	A steady decrease in the free JVM heap size is an indicator of either a memory leak or misconfigured application server. A JVM running out of heap will experience instability and performance degradation as garbage collector and JVM competes for resources to perform cleanup and object creation respectively.
Open Sessions Count	If open session count drops to 0 and remains at 0 for a period of time, some investigation is warranted. Often this pattern indicates a network or load balancing problem.
Application Invocation Count	If application invocation count drops to 0 and remains at 0 for a period of time, some investigation is warranted. While this pattern often indicates a network or load balancing problem, it could also be a symptom of a hanged server.

Understanding these key WebLogic system metrics, setting the SLO thresholds and assigning appropriate responses are critical to establishing a proactive monitoring. In this example, we will configure SLOs and actions with ADP.

The first task is to set a cautionary and a violation SLO for ExecuteQueue Idle Thread Count metric so the appropriate person can be alerted when available ExecuteQueue is running low. To configure SLOs, right-click on the Execute Queues metric and select **Configure service level objects**. In this example, we will create the following SLOs for ExecuteQueue Idle Threads:

**Table 19–3 SLOs for ExecuteQueue Idle Threads**

SLO Name	Metric	Threshold Type	Threshold Value	Trigger On
Low ExecuteQueue Idle Threads	Metric.J2EE.Dispatcher.IdleThreads	Cautionary	3	Low
ExecuteQueue Idle Threads Exhaustion	Metric.J2EE.Dispatcher.IdleThreads	Violation	0	Low



When SLO trigger is set to Low, ADP will fire an alert when current measurement reaches the threshold value AND the previous measurement has a higher value than the threshold.

For example, we would create the following actions for the SLOs previously configured:

**Table 19–4 Action for SLO**

SLO Name	Action Name	Action Type
Low ExecuteQueue Idle Threads	Enter Low ExecuteQueue Idle Threads event into server log	Log
ExecuteQueue Idle Threads Exhaustion	Email ExecuteQueue Idle Threads Exhaustion alert	Email
ExecuteQueue Idle Threads Exhaustion	Send ExecuteQueue Idle Threads Exhaustion SNMP trap to HP Overview	SNMP
ExecuteQueue Idle Threads Exhaustion	Enter ExecuteQueue Idle Threads Exhaustion event into server log	Log

After configuring these SLOs and actions, we now have a proactive monitoring environment to detect ExecuteQueue resource starvation related problems before a catastrophic event occurs. We would use this approach to establish proactive monitoring for other key WebLogic system metrics.

The following is the Oracle recommendation:

**Table 19–5 ExecuteQueue Pending Requests**

SLO Name	Metric	Threshold Type	Threshold Value	Trigger On
ExecuteQueue Pending Request Warning	Metric.J2EE.Dispatcher.PendingRequests	Cautionary	5 ~ 10 <sup>1</sup>	High
ExecuteQueue Pending Request Violation	Metric.J2EE.Dispatcher.PendingRequests	Violation	10 ~ 20	High

<sup>1</sup> **Threshold values for these SLOs vary by environment.** Figuring out what threshold values to use is an iterative process. Users should gather information about the performance characteristic of their WebLogic environment as the first step. Based on this information, users can set SLOs accordingly. As users continue to improve the performance of their WebLogic environment, they should re-evaluate these threshold values and change them as needed.

When SLO trigger is set to High, ADP will trigger an alert when current measurement hits the threshold value.

**Table 19–6 Total JVM Heap Size**

SLO Name	Metric	Threshold Type	Threshold Value	Trigger On
JVM Heap Reached Max	Metric.J2EE.JVM.HeapSizeCurrent	Cautionary	512 MB <sup>1</sup>	High
JVM Heap Reached 0	Metric.J2EE.JVM.HeapSizeCurrent	Violation	0 MB	Low

<sup>1</sup> **Threshold value for this SLO varies by environment.** Users would set this value to the maximum heap size specified in the WebLogic configuration file.

**Table 19–7 Free JVM Heap Size**

SLO Name	Metric	Threshold Type	Threshold Value	Trigger On
Low JVM Free Heap Warning	Metric.J2EE.JVM.HeapFreeCurrent	Cautionary	72 MB	Low
Low JVM Free Heap Violation	Metric.J2EE.JVM.HeapFreeCurrent	Violation	24 MB	Low

**Table 19–8 Open Session Count**

SLO Name	Metric	Threshold Type	Threshold Value	Trigger On
No user session in system for 5 minutes	Metric.J2EE.WebApplication.OpenSessionCurrentCount	Cautionary	0 <sup>1</sup>	Low

<sup>1</sup> In the example, this SLO would have a measurement window of 5 minutes. By setting the measurement window to 5 minutes, ADP will fire an alert only if this condition persists for at least 5 minutes.

**Table 19–9 Application Invocation Count**

SLO Name	Metric	Threshold Type	Threshold Value	Trigger On
No application invocation in system for 5 minutes	Metric.J2EE.Servlet.InvocationTotalCount	Cautionary	0	Low

Setting these SLOs and corresponding actions establishes a proactive monitoring environment for your WebLogic deployment. This proactive monitoring approach allows you to identify problems leading up to catastrophic problems before they impact your system's performance and availability.

## 19.6 Conclusion

The ADP Methodology is a critical aspect of your application performance management strategy. By following this methodology carefully, you will be able to use ADP to improve your ability to proactively monitor the performance and availability of your deployed applications and WebLogic infrastructure. ADP's automation reduces time, effort, and errors associated with manual processes. This allows ADP users to focus on other crucial activities such as the ones listed in the Oracle Methodology.

---

---

## Frequently Asked Questions About ADP

This chapter answers frequently asked questions regarding Application Dependency and Performance.

This chapter includes the following sections:

- [Section 20.1, "How Many Resources Can Be Added to the ADP Manager?"](#)
- [Section 20.2, "Can I Erase the darchive Directory?"](#)
- [Section 20.3, "How Do I Undeploy the Agent?"](#)

### 20.1 How Many Resources Can Be Added to the ADP Manager?

There is a default resource limit set in the `Acsera.properties` file to 2 resources. This is done to prevent overwhelming the default memory configuration of the CAMM manager. Edit `Acsera.properties` and search for "ResourceLimit". Change the value from 2 to your desired limit, or 0 for unlimited. After doing this, you will need to restart the manager. You should check the manager log prior to this to ensure that you have adequate free memory to accommodate additional resources.

### 20.2 Can I Erase the darchive Directory?

In general, you should not erase the darchive directory while the CAMM manager is running. If you erase the darchive directory after shutting down the CAMM manager, the darchive directory will be re-populated when the CAMM manager next restarts.

If the size of the darchive directory is of concern, determine which subdirectories are occupying the most space, in particular the `server_archive` directory. If there are large directories that do not contain Java class files (for example `.`, `..`, or `wlserver10.3`), modify the `Acsera.properties` file to exclude them. Search for `Model.StaticAnalysis.ExcludeClassPaths` and add the path to the list, separated by comma.

### 20.3 How Do I Undeploy the Agent?

Undeploy the agent using the same interface you used to deploy the agent. There is a drop-down menu in the last step in the deployment process that, by default, is set to *deploy*. Change it to *disable* to remove the agent startup arguments from the application servers, and to *remove* to erase the agent files from the application servers.



---



---

## Configuration Directories and Files

This appendix lists and defines the files and directories available in ADP. Topics include:

- [Section A.1, "Configuration Directories"](#)
- [Section A.2, "Acsera.properties File"](#)
- [Section A.3, "UrlMap.properties"](#)

### A.1 Configuration Directories

After ADP is installed all the components of the application package are located in \$CAMM\_HOME directory.

#### A.1.1 Directory Structure

The following is the \$CAMM\_HOME directory structure once ADP is installed:

```
-apache-tomcat-5.5.20
-bin
-config
-darchive (created when manager first discovers monitored application servers)
-deploy
-j2sdk
-lib
-log
-mcconfig
-tmp
-uninstall Oracle Composite Application Monitor and Modeler
-userdata
```

**Table A-1** Directories in \$CAMM\_HOME Directory

Directory	Description
apache-tomcat-5.5.20	Tomcat server where ADP Web Application (GUI) is located. By default, Apache Tomcat is disabled.
bin	Contains all the executable files to start and stop ADP, run deployer for Agent and ADP EJB, run export utility.
config	Contains all the ADP runtime configuration parameters that control execution logic, ADP schemas enablement, ADP GUI functionality, Service Level Objectives definition, export logic and many more.

**Table A-1 (Cont.) Directories in \$CAMM\_HOME Directory**

Directory	Description
darchive	Stores temporary images for Java EE applications that need to be monitored as well as the results of analysis and modeling of the Java EE application. This directory is created once ADP is up and running.
deploy	Contains agent libraries and configuration files, as well as <i>ADP EJB</i> and <i>ADP Admin Web Application</i> . These components are deployed on the remote host (Web or Application servers) using <i>deployer</i> utility found in bin directory of ADP package.
j2sdk	Contains minimum Java SDK sufficient to run ADP Server.
lib	Has all the libraries required for ADP's proper functionality
log	Has all the diagnostics records of ADP performance metrics collection activities. Also logs indicating successful deployment of ADP Agents can be found here. This directory is created once ADP is up and running.
mcconfig	Contains internal base instrumentation configuration. Do not modify these files.
tmp	Contains all the metadata definitions used for ADP's needs.
Uninstall CAMM	Contains utility used to uninstall the ADP.
userdata	Contains saved custom views per user.

### A.1.2 Config Directory

Config directory has many files that potentially can be configured and make ADP to run in a particular way. Any changes applied to files in this directory require restarting ADP server.

Most of the files never get touched directly by user. The following are the main three files which can be configured manually to achieve desired effect:

File	Description
Acsera.properties	This file is the main ADP configuration file customization of which helps to tune up ADP.
configuration.xml	In this file you define location of Administration Server and credentials to access it. Usually you do not touch this file. The entire configuration is done through ADP GUI.
dbconfig.xml	This file contains database configuration information for the ADP metric repository.
export.xml	This file contains information that drives proper data export logic. It is used for manual and automatic export of performance metric and events data from the ADP Data Repository.
UrlMap.properties	This file is used to map server addresses to load balancer addresses. By default, this file is not available; it must be created by the user.

It is worth mentioning that Service Level Objective definitions and Actions associated with the SLOs are described in *slo.xml* and *event.xml* respectively. The content of these files is completely controlled by definitions applied from ADP GUI (configuration tab).

### A.1.3 BIN Directory

The */bin* directory has all the executable files to start and stop ADP, run deployer for Agent and ADP EJB, run export utility.

There are two main reasons why one would need to customize the content of the files in this directory:

- To change the pointers to the location of Java Runtime Environment and ADP installation directory (ACSERA\_HOME)
- To configure ADP Server JVM parameters, e.g. memory.

Only the files that are frequently customized will be described in this chapter.

All the files in this directory should have these lines, pointing to the ADP Installation location:

For UNIX: `ACSERA_HOME=/home/CAMM; export ACSERA_HOME`

For Windows: `set ACSERA_HOME=C:\oracle\em11g\`

In addition, `acseraenv.sh(.bat)` should have a pointer to the JDK used by ADP:

- For UNIX: `JAVA_HOME=/home/oracle/em11g/j2sdk;export JAVA_HOME`
- For Windows: `set JAVA_HOME=C:\oracle\em11g\j2sdk`

ADP is a Java application and runs in its own JVM. Default size of JVM memory is 1024 MB. Should you want to change this value you can do so by modifying `-Xms` and `-Xmx` parameters in `acsera.sh (.bat)` file.

Bear in mind that if you have installed ADP as a Windows service and need to change the JVM Memory size, you need to change the size either by updating the Windows Registry or rerunning the `createmanagerservice.bat` utility with the new JVM parameters.

### A.1.4 Deploy Directory

The `/deploy` directory contains the ADP Java Agent distributable, including configuration files as well as corresponding libraries. These files are copied to the target systems hosting the Managed Servers when running the deployer utility. Rarely one needs to modify configuration files in this directory. Remember though if you modify the files they will be distributed to ALL targets within single server/cluster.

### A.1.5 apache-tomcat-5.5.20 Directory

The `/apache-tomcat-5.5.20` directory has the same structure and content as the standard Tomcat distributable. One interesting point about it is that ADP GUI application `acseraadimn.war` is residing in `em11g/apache-tomcat-5.5.20/webapps` directory.

## A.2 Acsera.properties File

The `acsera.properties` file contains global configuration parameters that define the operation of the ADP Manager.

### A.2.1 Log Files Management

This section of `Acsera.properties` file defines log rotation policies. `Log.MaxFiles` indicates max number of log files available at any given moment, whereas the `Log.MaxFileSizeMB` indicates maximum size of the log file.

#### **Example A-1 Log Files Management Section**

```
Log.CopyOut = false
```

```
Log.MaxFiles = 10
Log.MaxFileSizeMB = 30
Log.MergeLogs = true

Debug.CopyOut = false
Debug.LogLevel = all
Debug.MaxFiles = 10
Debug.MaxFileSizeMB = 30
```

Log files are stored in the log directory.

## A.2.2 Multi-Domain Monitoring Configuration

One can limit number of domains to be monitored by setting resource limit parameter: `ConfigurationManager.ResourceLimit=4`

### *Example A–2 Multi-Domain Monitoring Configuration*

```
ConfigurationManager.ResourceLimit=4
```

The default is 2, this means that by default ADP is set to monitor no more than two Application Server domains.

## A.2.3 ADP RMI Port Assignment

ADP uses RMI ports for communication with the agents and collects incoming performance metrics from a particular RMI port. By default, the RMI port is set on the same machine that hosts ADP. `RMI.Registry.Host` needs to be un-commented and have a value other than localhost if the host is multi-homed (such as, many network interfaces or has any ipv6 addresses) and you need to make sure that ADP listens to the incoming traffic on the particular interface.

You may need to change `RMI.Registry.Port` value in case the default 51099 port number has been allocated to an other application. Also if ADP is running in multi-instance mode, the port number will be different from instance to instance.

### *Example A–3 ADP RMI Port Assignment*

```
#RMI.Registry.Host = localhost
RMI.Registry.Port = 51099
```

## A.2.4 ADP Aggregation and Data Life Time Configuration

ADP has sophisticated multi-tiered logic for aggregation (or compression) of performance data. This helps to optimize performance of interaction with the internal data repository both when querying data for presentation or inserting new performance metrics.

Users who want to store longer term data should look for this section in `Acsera.properties`:

```
#####
# Production setting
# NOTE: use Model.GlobalSamplingRateSecs to configure Metric.Grain.0
#####
Metric.Grain.0 0s
Metric.TableInterval.0 = 4h
Metric.DataLife.0 = 8d

Metric.Grain.1 = 3m
```



```

Metric.TableInterval.1 =1d
Metric.DataLife.1 = 8d

#Metric.Grain.2 = 30m
#Metric.TableInterval.2 = 7d
#Metric.DataLife.2 = 420d

```

and uncomment the last 3 lines for the Metric.\*.2 properties

## A.2.5 Aggregating Incoming Metrics On the Fly

ADP by default aggregates data coming from multiple cluster members by application thus minimizing rate of insertion in to the data repository. This greatly improves performance of ADP in heavily loaded environments.

As a side effect of this approach though, the user is unable to see metrics from instrumentation (processes and portals) on per server level. If you need to enable this then set the JavaMIP.AggregateInserts to *false*.

## A.2.6 Configuring List of Applications To Be Monitored or Excluded From Monitoring

To avoid overhead of unnecessary monitoring of certain applications, you can explicitly state which applications to monitor, or which applications to exclude from monitoring.

Users should append the name of their application to the property ComponentProvider.Application.Exclude.

### **Example A-4 Specifying Which Applications to Monitor**

```

# Control which applications to analyze
#
ComponentProvider.Application.Exclude=WLI System EJBs,WLI-AI
Design-time,B2BDefaultWebAppApplication,WLI
Worklist,JWSQueueTransport,Deployer,BEA_WLS_DBMS_ADK,
Acsera,ClearApp,HttpDeployer,ServiceBus_Console,em

```

## A.2.7 Firewall Mitigation (for Internal RMI Ports)

If there is a firewall between the ADP Manager and the monitored application servers, ports need to be opened between them especially in the case where multiple resources are configured. For example, if two resources are configured and the first one uses 55006 as the port, then the next resource must use 55007 as the port. Each additional resource increments the port by 1.

In addition to the application server's JMX access ports, the following two properties in Acsera.properties indicate the ports used specifically by ADP:

- RMI.Registry.Port (51099 by default)
- RMI.JavaProvider.ServerPort (55003 by default)

## A.2.8 SLO Dampening

There are times when you deliberately want to cut down on the number of repeated notifications should SLO violation persist for a given period of time. To suppress notifications of the same violation in a short period of time, ADP provides the SLO Dampening feature. Once enabled, should a SLO violation occur and be repeated several times in a short period, ADP will not fire the SLO violation notification for the

time period defined in SLO.RearmDelay. To disable this feature, set the value of this parameter to 0.

SLO.SuppressDelayedAsserts indicates that if the violation still persists upon time period expiry ADP, should fire the SLO notification. By default it is *false*, for example, fire the notification.

#### **Example A-5 SLO Dampening**

```
# The following property is specified in units of
# minutes (m), hours (h) or days (d)
SLO.RearmDelay = 15m
SLO.SuppressDelayedAsserts = false
```

### **A.3 UrlMap.properties**

The UrlMap.properties file should be created in the ADP Manager's config directory and used to provide address mappings between load balancers and application servers. The format of this file is:

```
# Format:
#   $app_server_ip = $load_balancer_id
# E.g:
#   http://localhost:7001 = http://localhost:7005
#
# Note: ":" character need to be escaped with "\"
#
http://192.168.128.53:7002 = http://192.168.3.187:80
http://192.168.128.53:7003 = http://192.168.3.187:80
http://192.168.128.54:7005 = http://192.168.128.54:7011
http://192.168.128.54:7006 = http://192.168.128.54:7011
```

---

---

## Support Matrix for Application Dependency and Performance

This appendix provides the support matrix information for Oracle Enterprise Manager Grid Control for Application Dependency and Performance (ADP) release 11.1.0.

### B.1 Standalone ADP Manager System Requirements

The system requirements are:

**Table B-1** Minimal Hardware Configuration

Minimal Hardware	Configuration
# of CPUs	2
Physical Memory	2 GB
Minimum Disk Space	10 GB SCSI

**Table B-2** Reference Hardware Configuration

Reference Hardware	Configuration
CPU Specification	Intel X86 3.6GHz 1MB 380 G4 Processor
# of CPUs	2
Total Physical Memory	4 GB
# of Memory Modules	2
Hard Disk Specification	72 GB SCSI Pluggable
# of Hard Disk Modules	2

---

---

**Note:** Hardware configuration varies based on the complexity of the monitored applications.

---

---

### B.2 ADP Manager Platform Support

This section lists the platforms supported by ADP.

**Table B-3 Operation System Information**

Operating System	Version	Chip Architecture
Windows Server 2003	Data Center Edition Enterprise Edition	IA32
Windows XP	Professional SP1+	
Windows 2000	Professional SP2+ Server SP2+ Advanced Server SP2+ Datacenter Server SP2+	
Oracle and Red Hat Enterprise Linux	2.x 3.x 4.x 5.x	IA32 IA32 IA32 IA32/IA64
Sun Solaris	8 9 10	SPARC 64
IBM AIX	5.2+	PowerPC_Power 5 64-bit
HP UX	11i	IA64/PA-RISC <b>Note:</b> For IA64 use JDK 1.5.x.

**Table B-4 Repository Information**

Repository Database	Version
Oracle Database	10g, 11g

**Table B-5 Managed Server Platforms**

Managed Server Platforms	Version
Oracle WebLogic Server	8.1 SP3 8.1 SP4 8.1 SP5 8.1 SP6 9.1.x 9.2.x 10.0.x 10.1.x 10.2.x 10.3.x 11g, 11g PS1 (10.3.3)
Oracle WebLogic Integration	8.1 SP2 8.1 SP3 8.1 SP4

**Table B-5 (Cont.) Managed Server Platforms**

<b>Managed Server Platforms</b>	<b>Version</b>
Oracle WebLogic Portal	8.1 SP2 8.1 SP3 8.1 SP4 8.1 SP5 8.1 SP6 9.2.x 10.0.x 10.1.x 10.2.x 10.3.x
WebCenter	11gR1, 11gR1 PS1
Oracle Application Server	10.1.3.x
Oracle SOA Suite	10.1.3.x 11gR1, 11gR1 PS1
Oracle Service Bus (formerly Aqualogic Service Bus)	2.6 2.6.1 3 10gR3
Oracle Application Integration Architecture (AIA)	2.2.1 2.3
IBM WebSphere Portal	5.1 6
IBM WebSphere AppServer	5.0.2 (JMX Only) 5.1 6.0.2 6.1



## A

---

access  
  application platform, 10-6  
  connectivity, 10-6  
accessing architecture view, 16-12  
acsera.properties file, A-3  
action configuration, 16-62  
activity  
  entity EJB, 16-44  
ADF task flows  
  Oracle WebCenter, 16-28  
administrators  
  privileges, 10-7  
  statistics, 10-7  
ADP  
  architecture, 15-4  
  backing up database, 10-14  
  configuring, 10-10  
  database, 10-6, 15-6  
  how to use, 19-1  
  installing and configuring data repository, 10-13  
  Java agents, 15-5  
  manager, 15-6  
  node, 16-59  
  running multiple instances, 10-12  
  starting, 16-2  
  user interface, 15-6  
ADP agents  
  deploying Java agent for Oracle platform, 10-12  
  deploying on WebLogic, 10-11  
  deploying on WebSphere, 10-11  
aggregated count  
  metric type, 16-13  
aggregating  
  incoming metrics, A-5  
  performance data, A-4  
aggregation data  
  export to file, 18-1  
AIX  
  installing ADP on, 10-9  
analysis  
  bottleneck, 17-3  
  delay, 17-3  
  impact, 17-4  
apache-tomcat-5.5.20 directory, A-3

application  
  domains, 10-7  
Application Schema model  
  functional view, 16-10  
Application Server  
  analyzing historical performance, 3-7  
  extensible monitoring, 3-9  
  managing configurations, 3-14  
applications, 16-47 to 16-56  
  Web, 16-51  
architecture  
  ADP, 15-4  
architecture view, 16-11  
  accessing, 16-12  
  summary, 16-12  
  types of, 16-11  
arguments  
  export.xml file, 18-4  
async dispatchers, 16-41  
attributes  
  export.xml file, 18-3, 18-4  
automatic domain member discovery, 2-4  
average timing  
  metric type, 16-13

## B

---

backing up ADP database, 10-14  
baseline performance, 19-3, 19-6  
beans  
  entity, 16-53  
  message driven, 16-55  
  stateful, 16-52  
  stateless, 16-51  
bin directory, A-2  
blackouts  
  configuring for Service Level Objectives, 16-4  
  creating for Service Level Objectives, 16-5  
  deleting Service Level Objectives, 16-5  
  Service Level Objectives summary list, 16-5  
books  
  Oracle WebLogic portal, 16-17  
bottlenecks  
  analysis, 17-3  
  identifying performance, 19-4, 19-7  
  removing performance, 19-4

business SLAs guidelines, 19-5  
byte statistics, JMS destinations, 16-43

## C

---

cache  
  entity EJB, 16-44  
  stateful EJB, 16-52  
certificate  
  importing into manager's keystore, 10-17  
channels node  
characterization, performance, 17-1  
class node  
  in processes, 16-34  
clustering, 10-7  
  WebLogic, 10-7  
clusters  
  WebLogic, 10-7  
comparative view  
  performance analysis, 16-8  
component level  
  architecture view, 16-11  
component level execution  
  architecture view, 16-11  
components  
  deploying, 10-10  
config directory, A-2  
configuration  
  action, 16-62  
  data life time, A-4  
  directories and files, A-1  
  export, 18-1  
  multi-domain monitoring, A-4  
  periodic export, 18-2  
configurations  
  managing, 3-14  
configuring  
  ADP, 10-10  
  ADP agent when WebLogic installed as Windows Service, 10-19  
  ADP data repository, 10-13  
  list of applications excluded from monitoring, A-5  
  list of applications to be monitored, A-5  
  MySQL DBMS for ADP, 10-14  
  Oracle DBMS for ADP, 10-13  
connectivity  
  configuring Oracle SOA Suite for, 10-17  
  configuring Oracle WebLogic Portal for security, 10-17  
  configuring Oracle WebLogic Server for security, 10-17  
  configuring WebSphere 5.1 for security, 10-15  
DHCP, 10-6  
  network, 10-6  
containers  
  persistent, 16-43  
  persistent in processes, 16-32  
  stateless, 16-43  
creating

  Service Level Objectives, 16-3  
  Service Level Objectives blackouts, 16-5  
custom metrics, 16-59  
  monitoring environment, 16-9

## D

---

dashboards  
  health indicators, 16-2  
data  
  example of exported for WebLogic, 18-4  
  export modes, 18-1  
  performance graphs and data items, 16-7  
data displayed  
  display interval, 16-6  
  time frame, 16-6  
data items  
  performance data, 16-7  
data life time configuration, A-4  
data repository  
  installing and configuring for ADP, 10-13  
  setting up, 10-10  
database  
  ADP, 10-6, 15-6  
  export data to, 18-1  
database administrator  
  contact for installation statistics, 10-7  
date, exporting, 18-1  
delay analysis, 17-3  
  view  
    in processes, 16-31  
    Oracle BPEL processes, 16-23  
dependencies, environment, 10-5  
dependency  
  node, 16-49  
  types, 16-49  
deploy directory, A-3  
deploying  
  ADP components, 10-10  
deployments nodes, 16-50  
desktops  
  displaying portal, 16-39  
  Oracle WebLogic portal, 16-16  
destinations  
  JMS, 16-42  
DHCP connectivity, 10-6  
directory  
  apache-tomcat-5.5.20, A-3  
  bin, A-2  
  config, A-2  
  configuration, A-1  
  deploy, A-3  
  structure, A-1  
dispatchers  
  async, 16-41  
  sync, 16-41  
display interval  
  context, 16-6  
  data displayed, 16-6  
  time frame, 16-6



do-it-yourself manual processes, avoiding, 15-3  
domains  
    application, 10-7

## E

---

easy scroller, 16-9

### EJB

activity, 16-44  
activity table in processes, 16-32  
cache table in processes, 16-33  
entity cache, 16-44, 16-54  
entity locking, 16-45, 16-55  
entity transactions, 16-45, 16-54  
entity transactions table in processes, 16-33  
locking table in processes, 16-33  
message driven activity, 16-56  
message driven transactions, 16-56  
services, 16-36  
stateful cache, 16-52  
stateful locking, 16-53  
stateful transactions, 16-52

### entity

beans, 16-53  
EJB activity, 16-44  
EJB activity table in processes, 16-32  
EJB cache, 16-44, 16-54  
EJB cache table in processes, 16-33  
EJB locking, 16-45, 16-55  
EJB locking table in processes, 16-33  
EJB transactions, 16-45, 16-54  
EJB transactions table in processes, 16-33  
performance ranking, 17-1

environment dependencies, 10-5

### environments

service-oriented views, 15-2

### events

node performance, 16-46  
view in processes, 16-31  
viewer in Service Level Objectives, 16-4

### execute queues

health, 16-40

### execute thread

thread count, 16-40

### export

configuration, 18-1  
manual execution of metric, 18-2

### export modes

data, 18-1

exporting data, 18-1

export.xml file, 18-3

extensible monitoring, 3-9

## F

---

### file

export data to, 18-1

firewall mitigation, A-5

functional view of Application Schema model, 16-10

## G

---

### graphs

performance data, 16-7  
right-click operations, 16-8

### guidelines

business SLAs, 19-5

## H

---

### health

integration, 16-40

### health indicators

dashboard, 16-2

### histogram

performance, 17-2

how to use ADP, 19-1

HTTP services, 16-36

HttpDeployer.ear procedure, 10-11

## I

---

impact analysis, 17-4

improving performance, 19-3

### installing

ADP data repository, 10-13

on AIX, 10-9

on Linux, 10-9

on Solaris, 10-9

on Windows, 10-8

instrumentation in processes, 16-34

integration, 16-39 to 16-47

### interval context

display interval, 16-6

## J

---

### Java agents

ADP, 15-5

### Java EE application

managing, 15-1

Java EE application responsiveness, monitoring, 3-6

JDBC services, 16-36

JMS destinations, 16-42

### JSF pages

Oracle WebCenter, 16-29

## K

---

### keystore

importing certificate, 10-17

## L

---

### Linux

installing ADP on, 10-9

load regression, multi-point, 17-2

load scattergram, 17-2

### locking

entity EJB, 16-45

stateful EJB, 16-53

log files management  
acsera.properties file, A-3

## M

---

managing  
  configurations, 3-14  
mapping  
  SLAs to SLOs, 19-2, 19-5  
maximum response time measurement  
  metric type, 16-13  
memory leak detection, 17-3  
message driven  
  beans, 16-55  
  EJB activity, 16-56  
  EJB transactions, 16-56  
message statistics  
  JMS destinations, 16-42  
metadata view  
  Oracle BPEL processes, 16-23  
method node  
  in processes, 16-34  
methodology, ADP, 19-1 to 19-10  
metric export  
  manual execution, 18-2  
metrics  
  aggregating incoming, A-5  
  custom, 16-59  
  customizing for monitoring, 16-9  
  setting Service Level Objectives on, 19-4  
  setting SLOs on, 19-8  
  types, 16-13  
  WebLogic, 19-8  
Middleware  
  management, 1-1, 3-1  
minimum response time measurement  
  metric type, 16-13  
modeled entities view  
  Oracle BPEL processes, 16-25  
modes  
  data export, 18-1  
module level  
  architecture view, 16-11  
module level execution  
  architecture view, 16-11  
monitoring  
  configuring custom metrics, 16-9  
  extensible, Application Server, 3-9  
  multi-domain configuration, A-4  
MySQL DBMS  
  configuring for ADP, 10-14

## N

---

name  
  Service Level Objectives, 16-61  
navigation  
  easy scroller, 16-9  
node  
  hierarchy in processes, 16-31

  method in processes, 16-34  
node manager  
  used by WebLogic, 10-7  
nodes  
  ADP, 16-59  
  class in processes, 16-34  
  dependencies, 16-49  
  deployments, 16-50  
  hierarchy in Oracle BPEL processes, 16-25  
  pageflows, 16-35  
  services, 16-49

## O

---

OC4J  
  administrator  
    contact for installation statistics, 10-8  
  operation routing rules view  
    Oracle ESB, 16-28  
Oracle  
  Application Server  
    historical performance, 3-7  
    Web Cache, 3-6  
  deploying ADP Java agent for, 10-12  
  Oracle BPEL Processes, 16-22 to 16-25  
  Oracle DBMS  
    configuring for ADP, 10-13  
  Oracle ESB, 16-25 to 16-28  
  Oracle HTTP Server session volumes, 3-6  
  Oracle resources, 16-58  
  Oracle SOA Suite  
    configuring for secure connectivity, 10-17  
  Oracle WebCenter, 16-28 to 16-29  
  Oracle WebLogic  
    portals, 16-14 to 16-19  
    resources, 16-57  
  Oracle WebLogic Portal  
    configuring for secure connectivity, 10-17  
  Oracle WebLogic Server  
    configuring for secure connectivity, 10-17

## P

---

pageflow viewer  
  Oracle WebLogic portal, 16-17  
pageflows node, 16-35  
pages  
  JSF in Oracle WebCenter, 16-29  
  Oracle WebLogic portal, 16-18  
  WebSphere portals, 16-21  
parameters  
  defining Service Level Objectives, 16-3  
partner link bindings view  
  Oracle BPEL processes, 16-24  
partner link type role view  
  Oracle BPEL processes, 16-24  
partner links view  
  Oracle BPEL processes, 16-24  
PDF  
  saving view as, 16-9

- performance
  - analytics, 17-1 to 17-4
  - analyzing historical, 3-7
  - baseline, 19-3, 19-6
  - characteristics of target, 19-3
  - characterization, 17-1
  - comparative view analysis, 16-8
  - graphs and data items, 16-7
  - histogram, 17-2
  - identifying bottlenecks, 19-4, 19-7
  - improvement process, 19-3
  - node, 16-45
  - Oracle Application Server, historical, 3-7
  - removing bottlenecks, 19-4
  - system level, 19-7
- periodic export configuration, 18-2
- persistent containers, 16-43
  - in processes, 16-32
- platform
  - intelligent, 15-3
- port assignment
  - RMI, A-4
- portal desktop
  - displaying, 16-39
- portal hierarchy
  - Oracle WebLogic portal, 16-14
  - WebSphere portals, 16-19
- portals
  - Oracle WebLogic, 16-14
  - WebSphere virtual, 16-20
- portlet drill down
  - Oracle WebLogic portal, 16-17
- portlet performance
  - Web Services Remote Portlet producers, 16-38
- portlets
  - Oracle WebCenter, 16-29
  - Oracle WebLogic portal, 16-18
  - Web Services Remote Portlet producer, 16-38
  - WebSphere portals, 16-21
- privileges
  - administrators, 10-7
- process nodes
  - performance, 16-46
- processes, 16-30 to 16-35
  - deployed WebLogic, 16-30
  - node hierarchy, 16-31
- properties file
  - acsera.properties, A-3
  - UrlMap.properties, A-6

## R

---

- resources
  - Oracle, 16-58
  - Oracle WebLogic, 16-57
  - WebSphere, 16-57
- right-click operations
  - tables and graphs, 16-8
- RMI port assignment, A-4
- RMI ports

- firewall mitigation, A-5

## S

---

- saving
  - view as PDF, 16-9
- SCA *See* Service Component Architecture
- scattergram
  - load, 17-2
- scroller
  - easy navigation, 16-9
- secure connectivity
  - configuring WebSphere 5.1 for, 10-15
  - Oracle SOA Suite, 10-17
- secured connections
  - WebSphere, 10-15
- Service Component Architecture, 16-60
- service definition view
  - Oracle ESB, 16-27
- service details view
  - Oracle ESB, 16-26
- Service Level Objectives
  - blackouts summary list, 16-5
  - configuring, 16-3
  - configuring blackouts, 16-4
  - creating, 16-3
  - creating blackouts, 16-5
  - defining parameters for, 16-3
  - deleting blackouts, 16-5
  - events viewer, 16-4
  - name, 16-61
  - setting on key metrics, 19-4
  - types of, 16-4
  - variables, 16-62
- service mode
  - ADP topology, 15-5
- service operations view
  - Oracle ESB, 16-27
- service parent details view
  - Oracle ESB, 16-27
- service-oriented views, 15-2
- services, 16-35 to 16-36
- services node
  - applications, 16-49
- setting
  - Service Level Objectives on key metrics, 19-4
  - SLOs on key metrics, 19-8
- SLO dampening, A-5
- SLOs
  - See* Service Level Objectives
- snapshot count
  - metric type, 16-13
- SOA application
  - monitoring, 15-1
- Solaris
  - installing ADP on, 10-9
- starting ADP, 16-2
- stateful
  - beans, 16-52
  - EJB cache, 16-52

- EJB locking, 16-53
- EJB transactions, 16-52
- stateless
  - beans, 16-51
  - containers, 16-43
- statistics
  - administrators, 10-7
- steps in using ADP, 19-1
- subscribers node, 16-47
- sync dispatchers, 16-41
- system level performance, 19-7

## T

---

- tab descriptions
  - Oracle WebLogic portal level, 16-15
  - WebSphere portal level, 16-20
- tables
  - right-click operations, 16-8
- target
  - performance characteristics, 19-3
- task flows
  - ADF in Oracle WebCenter, 16-28
  - user-defined in Oracle WebCenter, 16-28
- terminology, xvi
- time frame
  - data displayed, 16-6
  - display interval, 16-6
  - specifying, 16-7
- time-based
  - performance distribution, 17-2
  - performance trend, 17-2
- topology
  - service mode, 15-5
  - Web Services Remote Portlet, 16-38
- topology view, 16-10
  - Oracle BPEL processes, 16-25
- transactions
  - entity EJB, 16-45
  - stateful EJB, 16-52
- tree summary
  - Oracle WebLogic portal, 16-15
  - WebSphere portals, 16-19
- trend
  - time-based performance, 17-2
- types
  - dependency, 16-49
  - metric, 16-13

## U

---

- UrlMap.properties file, A-6
- user
  - interface in ADP, 15-6
- user-defined task flows
  - Oracle WebCenter, 16-28

## V

---

- view
  - architecture, 16-11

- comparative, 16-8
- delay analysis in Oracle BPEL processes, 16-23
- delay analysis in processes, 16-31
- events in processes, 16-31
- functional of Application Schema model, 16-10
- metadata in Oracle BPEL processes, 16-23
- operation routing rules in Oracle ESB, 16-28
- partner link bindings in Oracle BPEL
  - processes, 16-24
- partner link type role in Oracle BPEL
  - processes, 16-24
- partner links in Oracle BPEL processes, 16-24
  - saving as PDF, 16-9
- service definition view in Oracle ESB, 16-27
- service details in Oracle ESB, 16-26
- service operations in Oracle ESB, 16-27
- service parent details in Oracle ESB, 16-27
- topology, 16-10
  - topology of Oracle BPEL processes, 16-25
- view modeled entities in Oracle BPEL
  - processes, 16-25
- viewers
  - pageflow for Oracle WebLogic portal, 16-17
- views
  - zooming in and out, 16-9

## W

---

- Web 2.0 service
  - Oracle WebCenter, 16-29
- Web applications, 16-51
- Web Services, 16-35
- WebLogic
  - administrator
    - contact for installation statistics, 10-8
  - clustering, 10-7
  - clusters, 10-7
  - deploying ADP agents on, 10-11
  - example of exported data, 18-4
  - installed as Windows Service, 10-19
  - key metrics, 19-8
  - node manager, 10-7
- WebLogic Domain Refresh job, 2-4
- WebSphere
  - administrator
    - contact for installation statistics, 10-8
  - deploying ADP agents on, 10-11
  - deploying ADP Java Agent for, 10-12
  - file transfer application, 10-12
  - libraries required, 10-11
  - portals, 16-19 to 16-22
  - post-deployment requirements, 10-15
  - resources, 16-57
  - secured connections, configuring for, 10-15
- WebSphere 5.1
  - secure connectivity, 10-15
- Windows
  - installing ADP on, 10-8
- Windows Service
  - configuring ADP agent, 10-19

workshop projects, 16-50  
WSRP producers, 16-37 to 16-39

## **Z**

---

zooming in and out for views, 16-9

