Oracle® Fusion Middleware

Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite

11*g* Release 1 (11.1.1.6.2)

E10226-14

March 2012

Documentation for system administrators that describes how to administer Oracle Service-Oriented Architecture (SOA) composite applications consisting of binding components and Oracle BPEL process, human task, business rule, Oracle Mediator, and Oracle Business Process Management Notation (BPMN) service components in Oracle Enterprise Manager Fusion Middleware Control. Includes additional information on administering Oracle B2B, Oracle Business Activity Monitoring (BAM), Oracle JCA Adapters, Oracle User Messaging Service, and business events.

ORACLE®

Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite, 11*g* Release 1 (11.1.1.6.2)

E10226-14

Copyright © 2006, 2012, Oracle and/or its affiliates. All rights reserved.

Primary Author: Anirban Ghosh, Solveig Haugland, Mark Kennedy, Robert May, Richard Smith, Carol Thom, and Savija Vijayaraghavan

Contributor: Oracle SOA Suite and Oracle Fusion Middleware development, product management, and quality assurance teams

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 is software or related documentation that 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 America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware 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 that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware 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

Preface xvii

Audience xvii
Documentation Accessibility xvii
Related Documents xvii
Conventions xviii

What's New in This Guide for Release 11.1.1.6.*x* xix

Part I Introduction to Oracle SOA Suite and Oracle Business Process Management Suite

1 Introduction and Concepts

What Is Oracle Fusion Middleware? 1-1
What Is Oracle SOA Suite? 1-1
Introduction to the SOA Infrastructure Application 1-2
Introduction to SOA Composite Applications 1-4
Introduction to SOA Composite Application Instances 1-6
Introduction to Service Components and Service Component Instances 1-7
Introduction to Binding Components 1-9
Introduction to Service Engines 1-9
Introduction to the Service Infrastructure 1-11
Introduction to the Contents of SOA Composite Applications 1-11
What Is Oracle Business Process Management Suite? 1-11
Administration of Oracle SOA Suite and Oracle BPM Suite 1-12
Configuration of Oracle SOA Suite and Oracle BPM Suite 1-13
Monitoring of Oracle SOA Suite and Oracle BPM Suite 1-15
Management of Oracle SOA Suite and Oracle BPM Suite 1-15
Administration for Application Developers 1-22

Part II Getting Started with Administration

2 Getting Started with Administering Oracle SOA Suite and Oracle BPM Suite

Logging In to Oracle Enterprise Manager Fusion Middleware Control 2-1
Navigating to Oracle SOA Suite and Oracle BPM Suite Administration Tasks 2-2
Navigating Through the SOA Infrastructure Home Page and Menu 2-2

Navigating Through the SOA Composite Application Home Page and Menu 2-8
Navigating Through the Partition Home Page and Menu 2-11
Navigating to Deployed Java EE Applications 2-12
Navigating to the Oracle WebLogic Server Administration Console and Other Pages 2-14
Navigating to the SOA Infrastructure or SOA Composite Application Home Page from the Farm Home Page 2-16
Logging Out of Oracle Enterprise Manager Fusion Middleware Control 2-17

Part III Administering the SOA Infrastructure

3 Configuring the SOA Infrastructure

Configuring SOA Infrastructure Properties 3-1
Disabling Instance and Fault Count Metrics Retrieval with the System MBean Browser 3-9
Stopping and Starting the Managed Server and SOA Infrastructure 3-9
Waiting for SOA Infrastructure Startup Initialization to Complete 3-10
SOA Composite Application States and SOA Infrastructure Shutdown 3-10
Restarting the SOA Infrastructure Does Not Activate Endpoints When a Retired Composite is Activated 3-11
SOA Infrastructure Startup Failure When cwallet.sso Includes the SOA Map 3-11
Changing the SOA Infrastructure Server URL Property Port 3-12
Configuring Log Files 3-13
Configuring the Logging File Encoding Property 3-15
Changing the Driver Name to Support Custom XA Drivers 3-16
Specifying a Nondefault XA Transaction Timeout Value for XA Data Sources 3-16
Configuring Local Optimization 3-17
Condition Checks for Using Local Optimization 3-17
Overriding or Forcing Local Optimization 3-18
Local Optimization Logging 3-19
Local Optimization Calls Use Case 3-20

4 Monitoring the SOA Infrastructure

Monitoring SOA Infrastructure Recent Instances and Faults and Deployed Composites 4-1
Monitoring Message Delivery Processing Requests 4-5
Monitoring Service and Reference Binding Components in the SOA Infrastructure 4-6

Part IV Administering SOA Composite Applications and Instances

5 Securing SOA Composite Applications

Introduction to Securing SOA Composite Applications 5-1
Configuring Oracle HTTP Server with Oracle BPM Worklist 5-2
Setting up SAML Message-Protected Policy Configuration for the SOA Infrastructure 5-2
Automatically Authenticating Oracle BPM Worklist and Oracle Business Process Management Users 5-4
Automatically Authenticating Oracle BPM Worklist Users in SAML SSO Environments 5-4
Automatically Authenticating Oracle BPM Worklist Users in Windows Native Authentication Environments 5-5

Automatically Authenticating Oracle Business Process Management Process Composer Users in Windows Native Authentication Environments 5-5
Listing the Authentication Provider 5-5
Listing Oracle Internet Directory as the First Authentication Provider 5-5
Accessing Web-based Applications with the Default Authentication Provider 5-6
Configuring SSL 5-6
Recommendation to Configure Either All or No Managed Servers with SSL 5-6
Switching from Non-SSL to SSL Configurations with Oracle BPM Worklist 5-6
Configuring SOA Composite Applications for Two-Way SSL Communication 5-6
Invoking References in One-Way SSL Environments in Oracle JDeveloper 5-9
Configuring Oracle SOA Suite and Oracle HTTP Server for SSL Communication 5-9
Configuring SSL Between SOA Composite Application Instances and Oracle WebCache ... 5-11
Using a Custom Trust Store for One-Way SSL During Design Time 5-11
Enabling an Asynchronous Process Deployed to an SSL-Enabled, Managed Server to Invoke Another Asynchronous Process Over HTTP 5-11
Configuring Security for Human Workflow WSDL Files 5-12

6 Monitoring SOA Composite Applications

Monitoring SOA Composite Application Recent Instances and Faults and Rejected Messages..... 6-1

7 Deploying and Managing SOA Composite Applications

Deploying SOA Composite Applications 7-2
Deploying SOA Composite Applications with Task Flows 7-4
Deploying SOA Composite Applications with ant Scripts and the WLST Command Line Tool 7-5
Redeploying SOA Composite Applications 7-5
Undeploying SOA Composite Applications 7-8
Managing the State of Deployed SOA Composite Applications 7-10
Managing the State of All Applications at the SOA Infrastructure Level 7-11
Managing the State of an Application from the SOA Composite Application Home Page .. 7-15
Starting and Stopping a Managed Oracle WebLogic Server on Which the SOA Infrastructure is Deployed in the Middle of BPEL Processing 7-19
Setting the Composite Instance Name 7-20
Automating the Testing of SOA Composite Applications 7-20
Managing SOA Composite Application Policies 7-23
WS-RM Sessions 7-26
Policy Attachments and Local Optimization in Composite-to-Composite Invocations 7-26
Exporting a Deployed SOA Composite Application 7-27
Grouping SOA Composite Applications into Partitions 7-29
Creating and Deleting Partitions 7-31
Performing Bulk Lifecycle Management Tasks on Composites in Partitions 7-34
Disabling and Enabling BPEL and BPMN Business Monitors 7-36

8 Managing SOA Composite Application Instances

Initiating a SOA Composite Application Test Instance 8-1

Specifying RPC/Literal-Style WSDL Files on the Test Web Service Page 8-7
Monitoring and Deleting SOA Composite Application Instances from the Application Home Page 8-7
Mismatch Between the Number of SOA Composite Application Instances and Service Component Instances 8-11
Instance States of Service Components and SOA Composite Applications 8-12
Monitoring and Deleting SOA Composite Application Instances at the SOA Infrastructure Level 8-12
Recovering from SOA Composite Application Faults at the SOA Infrastructure Level 8-15
Examples of Fault Recovery for BPEL Processes 8-19
Examples of Fault Recovery for BPMN Processes 8-21
Examples of Fault Recovery for Oracle Mediator 8-24
Recovering from SOA Composite Application Faults in the Application Home Page 8-27
Deleting Rejected Messages at the SOA Infrastructure Level 8-30
Deleting Rejected Messages from the Application Home Page 8-32

9 Managing Database Growth

Introduction to Managing Database Growth 9-1
Deleting Large Numbers of Instances with the Purge Script 9-1
Partitioning the Component Database Tables 9-2
Developing a Purging and Partitioning Methodology 9-3
Deleting Large Numbers of Instances with the Purge Scripts 9-4
Looped Purge Script 9-5
Looped Purge in Parallel Script with dbms_scheduler 9-7
Purge States 9-8
Resequenced Message Purge States for Oracle Mediator 9-8
Purging the Instances of a Specific SOA Composite Application 9-9
Executing the Purge Scripts 9-10
Partitioning Component Tables 9-11
Configuring Partitions 9-12
Introduction to the Verification Script 9-12
Component Tables 9-12
Executing the Verification Script 9-16
Verifying and Dropping Partitions 9-16
Moving Active, Long Running Instances to a Different Partition 9-17
Partial Partitioning of Components 9-19

10 Programmatically Managing SOA Composite Applications

Introduction to Programmatically Managing SOA Composite Applications 10-1
Facade API Interfaces 10-2
Facade API Examples 10-4
Retrieving the State of a Composite 10-4
Finding Composite and Component Instances 10-4

11 Diagnosing Problems with SOA Composite Applications

Introduction to the Diagnostic Frameworks 11-1
Introduction to WLDF 11-2

Introduction to the Diagnostic Framework 11-3
Predefined Incident Processing Rules 11-4
Supported Oracle SOA Suite Diagnostic Dumps 11-5
Supported DMS Metrics 11-19
Creating Watches and Notifications 11-23
Manually Triggering and Executing Dumps 11-27
Viewing Incident Packages with ADR Tools 11-29
Querying Problems and Incidents 11-29

Part V Administering BPEL Process Service Components and Engines

12 Configuring BPEL Process Service Components and Engines

Configuring BPEL Process Service Engine Properties 12-1
Configuring Automatic Recovery for Oracle BPEL Process Manager 12-4
Configuring Master Node Recovery Scheduling 12-6
Configuring Automatic Recovery Attempts for Invoke and Callback Messages 12-8
Setting the Audit Level at the BPEL Process Service Component Level 12-8

13 Monitoring BPEL Process Service Components and Engines

Monitoring the Audit Trail and Process Flow of a BPEL Process Service Component 13-1
Flow Traces for Composite-to-Composite Invocations on Multiple Servers 13-10
Monitoring BPEL 2.0 Activities in the Audit Trail and Process Flow 13-10
Monitoring BPEL Process Service Component Recent Instances and Faults 13-13
Monitoring BPEL Process Service Component Instances 13-15
Monitoring Sensor Data and Values in BPEL Process Service Components 13-16
Behavior of Activity Sensors in Compensate and CompensateScope Activities in BPEL 2.0 13-17
Monitoring BPEL Process Service Engine Instances and Faults 13-18
Monitoring BPEL Process Service Engine Request and Thread Statistics 13-21
Monitoring BPEL Process Service Engine Instances 13-22
Monitoring Deployed BPEL Process Service Components in the Service Engine 13-23

14 Managing BPEL Process Service Components and Engines

Recovering from BPEL Process Service Component Faults 14-1
Managing BPEL Process Service Component Policies 14-4
Recovering from BPEL Process Service Engine Faults 14-5
Performing BPEL Process Service Engine Message Recovery 14-7
Storing Instance and Callback Message Data in Oracle Coherence Distributed Cache on Oracle Exalogic Platforms 14-11
Introduction to the Oracle Coherence Caching Architecture 14-12
Running with Default SOA Cluster Nodes and Coherence Cache Grid Nodes 14-13
Configuring Oracle Coherence Caching 14-14
Configuring the Storage of Multiple Audit Trail Messages in One Transaction 14-14
Starting the BPEL Process Cache Servers 14-15

Part VI Administering Oracle Mediator Service Components and Engines

15 Configuring Oracle Mediator Service Components and Engines

Configuring Oracle Mediator Service Engine Properties 15-1
Configuring Resequenced Messages 15-5

16 Monitoring Oracle Mediator Service Components and Engines

Introduction to the Oracle Mediator Dashboard Pages 16-1
Recent Instances Section 16-1
Components Section 16-2
Recent Faults Section 16-3
Routing Statistics Section 16-3
Instance Rate Per Min Section 16-4
Monitoring Oracle Mediator Service Component Instances and Faults 16-4
Monitoring Oracle Mediator Service Component Recent Instances and Faults 16-4
Monitoring Oracle Mediator Service Component Instances 16-6
Monitoring Oracle Mediator Service Component Faults 16-7
Monitoring Oracle Mediator Routing Statistics 16-8
Monitoring Audit Trail and Fault Details for an Oracle Mediator Component Instance 16-9
Monitoring Oracle Mediator Service Engine Instances and Faults 16-10
Monitoring Oracle Mediator Service Engine Recent Instances and Faults 16-11
Monitoring Oracle Mediator Service Engine Instances 16-12
Monitoring Request Breakdown Statistics 16-14
Monitoring Deployed Oracle Mediator Service Components in the Service Engine 16-15
Monitoring Resequenced Messages 16-16
Monitoring Resequenced Messages from the Mediator Service Component Home Page 16-16
Monitoring Resequenced Messages from the Mediator Instance Dialog 16-20

17 Managing Oracle Mediator Service Components and Engines

Recovering From Oracle Mediator Service Component Faults 17-1
Managing Oracle Mediator Policies 17-4
Recovering From Oracle Mediator Service Engine Faults 17-5

18 Managing Cross-References

Deleting Cross-Reference Values 18-1

Part VII Administering Decision Service Components and Business Rules Service Engines

19 Monitoring Decision Service Components and Engines

Monitoring Business Rules Service Engine Recent Instances and Faults 19-1
Monitoring Business Rules Service Engine Statistics 19-4
Monitoring Business Rules Service Engine Instances 19-5
Monitoring Business Rules Service Engine Faults 19-7
Monitoring Business Rules Service Engine Deployed Components 19-9
Monitoring Decision Service Component Instances of a Composite Application 19-10
Monitoring Business Rule Tracing 19-12

Tracing Rule Execution at the Development Audit Level 19-13
Tracing Rule Execution at the Production Audit Level 19-19
Monitoring Decision Service Component Logs 19-20
Viewing Decision Service Component Logs 19-20
Setting the Diagnostic Logging Level with a Log Configuration 19-21

Part VIII Administering Human Task Service Components and Human Workflow Service Engines

20 Configuring Human Workflow Service Components and Engines

Configuring Human Workflow Notification Properties 20-1
Configuring Human Workflow Task Service Properties 20-5
Configuring Oracle HTTP Server for Task Form Attachments 20-9
Configuring the Pluggable Notification Service 20-9
Pluggable Notification Service Implementation 20-10
Pluggable Notification Service Registration 20-10
Globally Disabling the Automatic Release Timers for Oracle BPM Worklist Tasks 20-11
Configuring the Number of Email Notification Messages 20-12
Configuring Multiple Send Addresses 20-12
Configuring Notification Retries 20-13
Configuring the Identity Service 20-13
Adding an Authentication Provider 20-14
Creating Users and Groups in the Authentication Provider 20-18
Configuring the Directory Service 20-25
Customizing the Identity Provider 20-26
Seeding Users, Groups, and Application Roles using LDAP Tools 20-26
Changing the Default Password in the Embedded LDAP Server 20-27
Seeding Users or Groups through the LDAP Browser 20-27
Seeding Application Roles using WLST Scripts 20-28
Managing Application Roles in Oracle Enterprise Manager Fusion Middleware Control 20-28
Enabling Case Agnostic Group Names in Human Tasks 20-29
Configuring Security Policies for Human Workflow Web Services 20-29

21 Monitoring Human Workflow Service Components and Engines

Monitoring Recent Human Task Service Component Instances and Faults 21-1
Viewing the Status of Human Workflow Tasks 21-3
Monitoring Human Task Service Component Instances 21-5
Monitoring Human Workflow Service Engine Recent Instances and Faults 21-6
Monitoring Human Workflow Service Engine Active Requests and Operation Statistics 21-8
Monitoring Human Workflow Service Engine Instances 21-10
Monitoring Deployed Human Workflows in the Service Engine 21-11

22 Managing Human Workflow Service Components and Engines

Managing Human Task Service Component Policies 22-1
Recovering from Human Workflow Service Engine Faults 22-3
Managing the URI of the Human Task Service Component Task Details Application 22-5

Recovering from Human Task Service Component Faults ... 22-6
Managing Outgoing Notifications and Incoming Email Notifications ... 22-7
Moving Human Workflow Data from a Test to a Production Environment ... 22-9
Moving Human Workflow Data from Test to Production Environments ... 22-10
migration.properties File Syntax ... 22-12
ant Script Data Migration Syntax ... 22-19

Part IX Administering Oracle Business Activity Monitoring

23 Configuring Oracle Business Activity Monitoring

Introduction to Configuring Oracle BAM ... 23-1
Configuring Oracle BAM Web Basic Properties ... 23-2
Configuring Oracle BAM Web Applications Properties ... 23-3
Configuring the Application URL ... 23-4
Configuring the Report Loading Indicator ... 23-4
Configuring the Server Name ... 23-4
Configuring Oracle BAM Server Basic Properties ... 23-5
Configuring Oracle BAM Server Properties ... 23-5
Configuring the Data Source JNDI ... 23-7
Configuring the Application URL ... 23-7
Configuring Viewset Sharing ... 23-7
Configuring the Report Cache Persistence Manager ... 23-8
Configuring Oracle Data Integrator Integration Properties ... 23-8
Configuring the Outbound Email Account ... 23-8
Configuring the Logger ... 23-9
Configuring Oracle User Messaging Service ... 23-9
Configuring Oracle BAM Distribution Lists ... 23-11
Configuring Oracle BAM Adapter ... 23-13
Configuring Oracle BAM Adapter Properties ... 23-14
Configuring Oracle BAM Connection Factories ... 23-16
Configuring Trusted Domains ... 23-21
Configuring Credential Mapping ... 23-21
Configuring Oracle BAM Batching Properties ... 23-23
Configuring Security ... 23-26
Configuring Credential Mapping ... 23-26
Configuring Oracle BAM User Permissions ... 23-26
Configuring Secure Socket Layer ... 23-26
Using Oracle Internet Directory With Oracle BAM ... 23-27
Securing Oracle BAM JMS Resources ... 23-27
Calling Secure Web Services ... 23-27
Configuring Advanced Properties ... 23-29
Oracle BAM Configuration Property Reference ... 23-31

24 Monitoring Oracle Business Activity Monitoring

Introduction to Monitoring Oracle BAM ... 24-1
Monitoring Oracle BAM Server Components ... 24-1

Monitoring Oracle BAM Active Data Cache 24-2
Monitoring the Event Engine Component 24-4
Monitoring the Report Cache Component 24-6
Monitoring the Enterprise Message Sources 24-7
Monitoring the Client Requests in Progress 24-9
Monitoring Oracle BAM Web Applications 24-9
Monitoring Oracle BAM Report Server 24-10
Monitoring Open Connections 24-11
Monitoring Oracle BAM Web Services 24-11
Monitoring Oracle BAM Performance 24-12
Monitoring Oracle BAM Logs 24-14

25 Managing Oracle Business Activity Monitoring

Introduction to Managing Oracle BAM 25-1
Managing Oracle BAM Availability 25-1
Managing Oracle BAM Users 25-2
Defining Users and Groups 25-4
Using Previously Seeded Group Members 25-5
Adding Members to Application Roles 25-6
Introduction to Oracle BAM Application Roles 25-9
Configuring Oracle WebLogic Server Embedded LDAP Server 25-10
Populating Users in Oracle BAM Administrator 25-11
Managing Oracle BAM Object Ownership 25-14
Removing Invalid Users from Oracle BAM Administrator 25-15

Part X Administering Oracle User Messaging Service

26 Configuring Oracle User Messaging Service

Introduction to User Messaging Service 26-1
Components 26-2
Architecture 26-3
Introduction to Oracle User Messaging Service Configuration 26-3
Accessing User Messaging Service Configuration Pages 26-5
Setting the Storage Method 26-5
Adding or Removing User Messaging Preferences Business Terms 26-5
Configuring User Messaging Service Drivers 26-6
Configuring a Driver 26-6
Configuring User Messaging Service Access to the LDAP User Profile 26-34
Securing the Oracle User Messaging Service 26-36
Web Service Security on Notification 26-37
Enabling UMS Service Security 26-37
Enabling Client Security 26-37
Keystore Configuration 26-38
Client Aliases 26-38
Securing JMS Resources 26-39
Troubleshooting Oracle User Messaging Service 26-40

27 Monitoring Oracle User Messaging Service

Monitoring Oracle User Messaging Service ... 27-1
Using Message Status ... 27-3
Deregistering Messaging Client Applications ... 27-4
Monitoring Drivers Using the All Tab ... 27-4
Viewing Log Files ... 27-5
Configuring Logging ... 27-6
Viewing Metrics and Statistics ... 27-7

28 Managing Oracle User Messaging Service

Deploying Drivers ... 28-1
Deploying Drivers Using WLST Commands ... 28-2
Deploying Drivers Using Oracle Enterprise Manager Fusion Middleware Control ... 28-2
Deploying Drivers Using the Oracle Fusion Middleware Configuration Wizard ... 28-4
Deploying Drivers Using the wsadmin Tool ... 28-5
Undeploying and Unregistering Drivers ... 28-9

Part XI Administering Oracle JCA Adapters

29 Configuring Oracle JCA Adapters

Configuring the End Point Properties for an Inbound Adapter ... 29-1
Editing a Predefined Property for an Inbound Adapter ... 29-1
Adding Predefined Properties for an Inbound Adapter ... 29-2
Creating a New Property for an Inbound Adapter ... 29-3
Deleting a Property for an Inbound Adapter ... 29-4
Reverting a Property Value for an Inbound Adapter ... 29-5
Configuring the End Point Properties for an Outbound Adapter ... 29-6
Editing a Predefined Property for an Outbound Adapter ... 29-6
Adding a Predefined Property for an Outbound Adapter ... 29-8
Creating a New Property for an Outbound Adapter ... 29-8
Deleting a Property for an Outbound Adapter ... 29-9
Reverting a Property Value for an Outbound Adapter ... 29-10

30 Monitoring Oracle JCA Adapters

Monitoring Instances and Faults for an Inbound Adapter ... 30-1
Monitoring Recent Faults and Rejected Messages for an Inbound Adapter ... 30-2
Monitoring Faults and Rejected Messages for an Inbound Adapter ... 30-3
Searching for Rejected Messages for an Inbound Adapter ... 30-4
Deleting Rejected Messages for an Inbound Adapter ... 30-5
Monitoring Properties for an Inbound Adapter ... 30-6
Monitoring Instances and Faults for an Outbound Adapter ... 30-7
Monitoring Recent Faults for an Outbound Adapter ... 30-7
Monitoring Faults for an Outbound Adapter ... 30-8
Searching for Faults for an Outbound Adapter ... 30-9
Monitoring Properties for an Outbound Adapter ... 30-10
Monitoring Adapter Logs ... 30-11

Part XII Administering Oracle B2B

31 Configuring Oracle B2B

Configuring Oracle B2B Server Properties 31-1
Configuring Oracle B2B Operations 31-1
Configuring Oracle B2B Attributes 31-3

32 Monitoring Oracle B2B

Monitoring the Oracle B2B Infrastructure 32-1
Accessing Oracle B2B from the B2B Infrastructure Page 32-3
Viewing the Message Flow of an Oracle B2B Binding Component 32-3
Viewing Services and References 32-6
Accessing Oracle B2B Reports from the Oracle B2B Composite Flow Trace Page 32-7

Part XIII Administering Business Events

33 Managing Business Events

Introduction to the Event Delivery Network 33-1
Subscribing to Business Events 33-1
Managing Business Event Subscribers 33-4
Recovering from Business Event Faults 33-6

Part XIV Administering Binding Components

34 Configuring Service and Reference Binding Components

Configuring Service and Reference Binding Component Properties 34-1
Configuring Properties for Web Services 34-3
Configuring Properties for Oracle JCA Adapters 34-4
Changing the Endpoint Reference and Service Key for Oracle Service Registry Integration 34-29

35 Monitoring Service and Reference Binding Components

Monitoring Binding Component Instances and Faults 35-1
Monitoring Binding Component Rejected Messages 35-3

36 Managing Service and Reference Binding Components

Managing Binding Component Policies 36-1
Override Policy Configuration Property Values 36-4
Publishing Web Services to the UDDI Registry 36-4
Configuring the Environment for Publishing Web Services to UDDI 36-5
Publishing a Web Service to the UDDI Registry 36-7

Part XV Administering Oracle BPMN Process Service Components and Engines

37 Configuring Oracle BPMN Process Service Components and Engines

Configuring BPMN Process Service Engine Properties 37-1
Integrating Oracle BPM with Oracle Business Activity Monitoring 37-5
Task 1: Configure the Oracle BAM Adapter on Oracle BPM Server 37-6
Task 2: Enable Oracle BAM on the Oracle BPM Server 37-6

38 Monitoring BPMN Process Service Components and Engines

Viewing the Audit Trail and Process Flow of a BPMN Process Service Component 38-1
Monitoring BPMN Process Service Component Instances and Faults 38-7
Monitoring BPMN Process Service Component Instances 38-9
Monitoring BPMN Process Service Engine Instances and Faults 38-10
Monitoring BPMN Process Service Engine Request and Thread Statistics 38-12
Monitoring BPMN Process Service Engine Instances 38-13
Monitoring Deployed BPMN Processes in the Service Engine 38-14

39 Managing Oracle BPMN Service Components and Engines

Recovering from BPMN Process Service Component Faults 39-1
Managing BPMN Process Service Component Policies 39-4
Recovering from BPMN Process Service Engine Faults 39-5
Performing BPMN Process Service Engine Message Recovery 39-7

Part XVI Appendixes

A Demo User Community

Using the Demo User Community A-1
Users A-1
Groups A-3
soa-infra Application Roles A-4
SOATestDemoApp Application Roles A-5
Roles Granted to and Owned by Users A-5
WorkflowPermission Class A-6

B Troubleshooting Oracle SOA Suite and Oracle BPM Suite

Setting Logging Levels for Troubleshooting B-1
Log Files and Thread Dumps from All Managed Servers B-4
Connection and Transaction Timeout Troubleshooting B-5
Resolving Connection Timeouts B-5
Increasing Database Connection Values B-6
Updating the EJB Transaction Timeout Value in the Deployment Archive After SOA Infrastructure Failure B-6
Long Running, Synchronous Calls To Remote Web Services Error Out or Asynchronous Transactions Return with an Error after a Long Time B-7
Runtime Diagnostics Troubleshooting B-8
Oracle SOA Suite Runtime Failure with a "Cannot read WSDL" Error B-8
Automatic Recovery of BPEL Instances is Not Recovering A Specific Instance B-8

Some Composites Are Retried Multiple Times on Failure B-10
Application Transaction Does Not Complete and the Underlying Composite Is Stuck in a Running State B-10
Human Workflow Troubleshooting B-13
Task Assignment/Routing/Escalation Issues B-13
Task Action Issues B-14
Notification Issues B-15
Task View Issues B-17
Task Attribute Mapping Issues B-18
Task Report Issues B-19
Task History Issues B-20
Task Form/Action Issues B-22
Task Comments/Attachment Issues B-24
Design Time at Runtime Issues B-25
Human Workflow API (Including SOAP/EJB) Usage Issues B-26
Oracle JDeveloper Data Control / Form Generation Issues B-26
Human Workflow Service/ System MBean Browser Issues B-27
AMX Extension Issues B-29
Oracle BPM Worklist/Task Region Issues B-30
Test-to-Production Issues B-34
Identity Service Issues B-38
Event Delivery Network Troubleshooting B-40
Tuning EDN Event Bus and Delivery B-41
Rolled Back OAOO Event Delivery Messages are Displayed in the Log Files B-41
Events Are Consumed by Multiple Revisions of the Same Composites B-41
Business Event Is Picked Up Twice (Or More) By SOA Server B-42
Some Messages Are Lost Between EDN and Composites or Composites Across Clusters B-42
Performance Troubleshooting B-44
Optimizing the Loading of Pages with Instance and Fault Metrics B-44
Resolving Message Failure Caused by Too Many Open Files B-48
Resolving MaxMessageSizeExceededException Errors Caused By Large Payloads B-49
Extending Tablespaces to Avoid Problems at Runtime B-49
Resolving Database Growth Issues Caused by a High Volume of Transactions B-50
Observing Slow Application Performance Such as Longer Time to Serve Pages or Finish Transactions B-51
Server Troubleshooting B-53
Best Practices for Starting and Stopping a Managed Server B-53
Diagnosing SOA Server Startup Problems B-54
Specifying the Proxy Server B-55
Flow Diagram Does Not Display The First Time on Some Lower End Hosts B-55
Accessing Oracle Enterprise Manager Fusion Middleware Control on Dual Stack Hosts that Support IPv4 and IPv6 ..... B-55
Browser Troubleshooting B-55
Limitation on Using the Safari Browser to View WSDL File Content B-55
Additional Troubleshooting Documentation B-56

C Oracle Enterprise Manager Roles

Roles and Privileges C-1
Overall Role Functionality Matrix C-1
SOA Infrastructure Page Access C-2
SOA Infrastructure Menu Access C-3
SOA Composite Menu Access C-3
Composite Home Page Access C-3
BPEL Process Service Engine Access C-4
Oracle Mediator Service Engine Access C-4
Human Workflow Service Engine Access C-5
Business Rules Service Engine Access C-5
BPEL Process Service Component Home Page Access C-6
Oracle Mediator Service Component Home Page Access C-6
Human Task Service Component Home Page Access C-6
Decision Service Component Home Page Access C-7
Flow Trace Page Access C-7
Audit Trail Access C-7
Services Home Page Access C-7
References Home Page Access C-8
Oracle B2B Pages Access C-8
Business Events Page Access C-8
System MBean Browser Access C-9

Index

Preface

Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite describes how to administer the components of Oracle SOA Suite and Oracle Business Process Management Suite, including

- The SOA Infrastructure and SOA composite applications
- Composite components and service engines such as the BPEL service engine, BPMN Process service engine, Oracle Mediator service engine, human workflow service engine, and business rules service engine
- Oracle B2B, Oracle Adapters, Oracle Business Activity Monitoring, and Oracle User Messaging Service

Audience

This document is intended for administrators managing applications on a SOA platform.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following Oracle resources:

- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*
- *Oracle Fusion Middleware User's Guide for Oracle B2B*
- *Oracle Fusion Middleware User's Guide for Oracle Business Activity Monitoring*
- *Oracle Fusion Middleware User's Guide for Technology Adapters*
- *Oracle Fusion Middleware User's Guide for Oracle Business Rules*

- *Oracle Fusion Middleware Language Reference Guide for Oracle Business Rules*
- *Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management*
- *Oracle Fusion Middleware User's Guide for Oracle Business Process Management*
- *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management*
- *Oracle Fusion Middleware Performance and Tuning Guide*
- *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite*
- *Oracle Fusion Middleware High Availability Guide*

Conventions

The following text conventions are used in this document:

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

What's New in This Guide for Release 11.1.1.6.*x*

For Release 11.1.1.6.*x*, this guide has been updated in several ways. The following table lists the sections that have been added or changed. If a feature was not available in the first release of 11.1.1.6.*x*, the last columns denote which documentation release contains the update.

For a list of known issues (release notes), see the "Known Issues for Oracle SOA Products and Oracle AIA Foundation Pack" at `http://www.oracle.com/technetwork/middleware/docs/soa-aiafp-knownissuesindex-364630.html`.

Sections	Changes Made	11.1.1.6.0	11.1.1.6.1
Chapter 1 Configuring the SOA Infrastructure			
Section 1.4.3, "Management of Oracle SOA Suite and Oracle BPM Suite"	Section revised to briefly describe GridLink support. GridLink data sources and multidata sources protect the SOA Infrastructure against database failures. For more information about Gridlink and Oracle SOA Suite, see Section "Configuring High Availability for Oracle SOA Suite" of *Oracle Fusion Middleware High Availability Guide*.	X	
Section 1.4.3, "Management of Oracle SOA Suite and Oracle BPM Suite"	Section revised to briefly describe the Oracle SOA Suite plug-in introspected by Oracle Virtual Assembly Builder. Oracle Virtual Assembly Builder is a tool for virtualizing installed Oracle components, modifying these components, and deploying them to your environment. For more information about the Oracle SOA Suite plug-in, see *Oracle Virtual Assembly Builder User's Guide*.	X	
Chapter 3 Configuring the SOA Infrastructure			

Sections	Changes Made	11.1.1.6.0	11.1.1.6.1
Section 3.1, "Configuring SOA Infrastructure Properties"	Section revised to describe the **Restrict display of instances and faults to the last *time_period*** checkbox and dropdown list for specifying a time period during which to retrieve recent instances, faults, and count metrics for display on various Oracle Enterprise Manager Fusion Middleware Control pages. By default, this checkbox is selected and the time period duration is set to 24 hours (one day).	X	
Section 3.7, "Configuring Local Optimization"	Section added to describe the condition checking performed to determine if local optimization is possible, how to override or force local optimization, and how to set local optimization logging.	X	
Chapter 4 Monitoring SOA Composite Applications			
Section 4.1, "Monitoring SOA Infrastructure Recent Instances and Faults and Deployed Composites"	Section revised to describe the **BPEL Message Recovery Required** message that indicates the retrieval of recent instances and faults that are displayed on a page is restricted to a specified time period. This message is displayed if BPEL process messages require recovery from the Recovery page of the BPEL process service engine.	X	
Chapter 6 Monitoring SOA Composite Applications			
Section 6.1, "Monitoring SOA Composite Application Recent Instances and Faults and Rejected Messages"	Section revised to describe the **BPEL Message Recovery Required** message that indicates the retrieval of recent instances and faults that are displayed on a page is restricted to a specified time period. This message is displayed if BPEL process messages require recovery from the Recovery page of the BPEL process service engine.	X	
Chapter 7 Deploying and Managing SOA Composite Applications			
Section 7.2, "Redeploying SOA Composite Applications"	Section revised to describe how the Redeploy SOA Composite wizard enables you to select whether to change the states of currently running instances to stale after redeployment of the SOA composite application or continue running instances upon redeployment. (This prevents these instance states from being changed to stale.)	X	
Section 7.4, "Managing the State of Deployed SOA Composite Applications"	Section revised to describe what occurs when you attempt to retire or undeploy the default composite revision.	X	
Chapter 8 Managing SOA Composite Application Instances			

Sections	Changes Made	11.1.1.6.0	11.1.1.6.1
Section 8.2, "Monitoring and Deleting SOA Composite Application Instances from the Application Home Page" and Section 8.3, "Monitoring and Deleting SOA Composite Application Instances at the SOA Infrastructure Level"	Sections revised to describe how to filter the display of instances by execution state (running, completed, terminated, or stale), fault state (with or without faults), and whether a recovery action is required.	X	
Chapter 9 Managing Database Growth			
Section 9.3.4, "Resequenced Message Purge States for Oracle Mediator"	Section added that describes how resequenced messages are purged for Oracle Mediator when the purge scripts are run.	X	
Section 9.3.5, "Purging the Instances of a Specific SOA Composite Application"	Section added that describes how to purge the instances of a specific SOA composite application and leave the instances of other composites unpurged.	X	
Section 9.4.3.1, "Partitioning Constraints"	Section added that describes how to perform complete table partitioning, no table partitioning, or partial partitioning of specific tables with high growth rates.	X	
Section 9.4.6, "Moving Active, Long Running Instances to a Different Partition"	Section added that describes how to move active, long running instances to a different partition. A partition can be dropped only when no instances are active.	X	
Chapter 11 Diagnosing Problems with SOA Composite Applications			
Chapter 11, "Diagnosing Problems with SOA Composite Applications"	Chapter added that describes how to identify Oracle SOA Suite problems early and take the proper corrective actions with the assistance of the WebLogic Diagnostic Framework (WLDF) and the Oracle Fusion Middleware Diagnostic Framework.	X	
Chapter 14 Managing BPEL Process Service Components and Engines			
Chapter 14.5, "Storing Instance and Callback Message Data in Oracle Coherence Distributed Cache on Oracle Exalogic Platforms"	Section added that describes how running Oracle SOA Suite 11*g* Release 1 11.1.1.6 on an Oracle Exalogic platform enables you to use the distributed cache feature of Oracle Coherence to store instance and callback message data from BPEL processes.	X	
Chapter 23 Configuring Oracle Business Activity Monitoring			
Section 23.11, "Oracle BAM Configuration Property Reference" (in Table 23–3)	Added information about property `EDS_Query_Timeout`, a configurable timeout for BAM Report queries.	X	
Appendix B Troubleshooting Oracle SOA Suite			
Section B.6.1, "Optimizing the Loading of Pages with Instance and Fault Metrics"	Section revised to describe the new methods for optimizing the loading of pages with instance and fault metrics.	X	

Part I

Introduction to Oracle SOA Suite and Oracle Business Process Management Suite

This part describes Oracle SOA Suite and Oracle Business Process Management Suite.

This part includes the following chapter:

- Chapter 1, "Introduction and Concepts"

1

Introduction and Concepts

This chapter describes Oracle Fusion Middleware, Oracle Service-Oriented Architecture (SOA) Suite, and Oracle Business Process Management (BPM) Suite and the types of Oracle SOA Suite and BPM Suite administration tasks you perform from Oracle Enterprise Manager Fusion Middleware Control.

This chapter includes the following topics:

- Section 1.1, "What Is Oracle Fusion Middleware?"
- Section 1.2, "What Is Oracle SOA Suite?"
- Section 1.3, "What Is Oracle Business Process Management Suite?"
- Section 1.4, "Administration of Oracle SOA Suite and Oracle BPM Suite"
- Section 1.5, "Administration for Application Developers"

For more information about Oracle Enterprise Manager Fusion Middleware Control administrative tasks and Oracle Fusion Middleware concepts, see the following documents:

- *Oracle Fusion Middleware Administrator's Guide*
- *Oracle Fusion Middleware 2 Day Administration Guide*
- *Oracle Fusion Middleware Concepts*

1.1 What Is Oracle Fusion Middleware?

Oracle Fusion Middleware is a collection of standards-based software products that spans a range of tools and services: from Java EE and developer tools, to integration services, business intelligence, and collaboration. Oracle Fusion Middleware offers complete support for development, deployment, and management of applications.

1.2 What Is Oracle SOA Suite?

Oracle SOA Suite is a middleware component of Oracle Fusion Middleware. Oracle SOA Suite provides a complete set of service infrastructure components for designing, deploying, and managing SOA composite applications. Oracle SOA Suite enables services to be created, managed, and orchestrated into SOA composite applications. Composites enable you to easily assemble multiple technology components into one SOA composite application. Oracle SOA Suite plugs into heterogeneous IT infrastructures and enables enterprises to incrementally adopt SOA.

You can administer Oracle SOA Suite from Oracle Enterprise Manager Fusion Middleware Control. The following sections provide an overview of the components of Oracle SOA Suite:

- Section 1.2.1, "Introduction to the SOA Infrastructure Application"
- Section 1.2.2, "Introduction to SOA Composite Applications"
- Section 1.2.3, "Introduction to SOA Composite Application Instances"
- Section 1.2.4, "Introduction to Service Components and Service Component Instances"
- Section 1.2.5, "Introduction to Binding Components"
- Section 1.2.6, "Introduction to Service Engines"
- Section 1.2.7, "Introduction to the Service Infrastructure"
- Section 1.2.8, "Introduction to the Contents of SOA Composite Applications"

For introductory information about Oracle SOA Suite, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

1.2.1 Introduction to the SOA Infrastructure Application

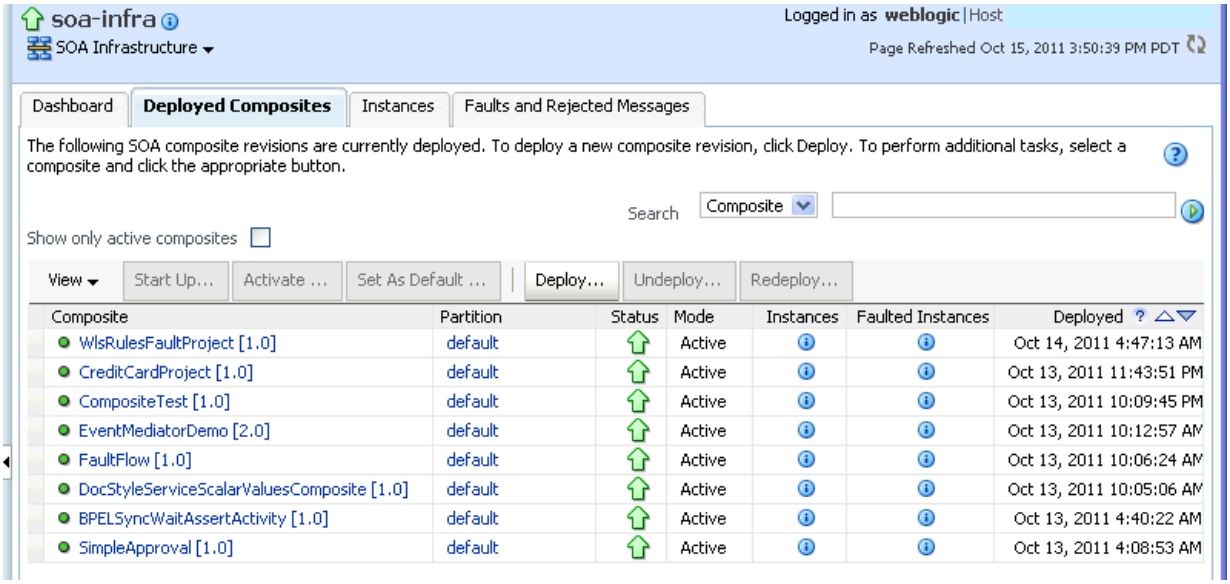
The SOA Infrastructure is a Java EE-compliant application running in Oracle WebLogic Server. The application manages composites and their lifecycle, service engines, and binding components.

You deploy SOA composite applications designed in Oracle JDeveloper to a partition of your choice on the SOA Infrastructure. Partitions are separate sections of your SOA Infrastructure that enable you to logically group the composite applications for ease of management.

In the example shown in Figure 1–1, many SOA composite applications are deployed to the SOA Infrastructure and are visible in Oracle Enterprise Manager Fusion Middleware Control.

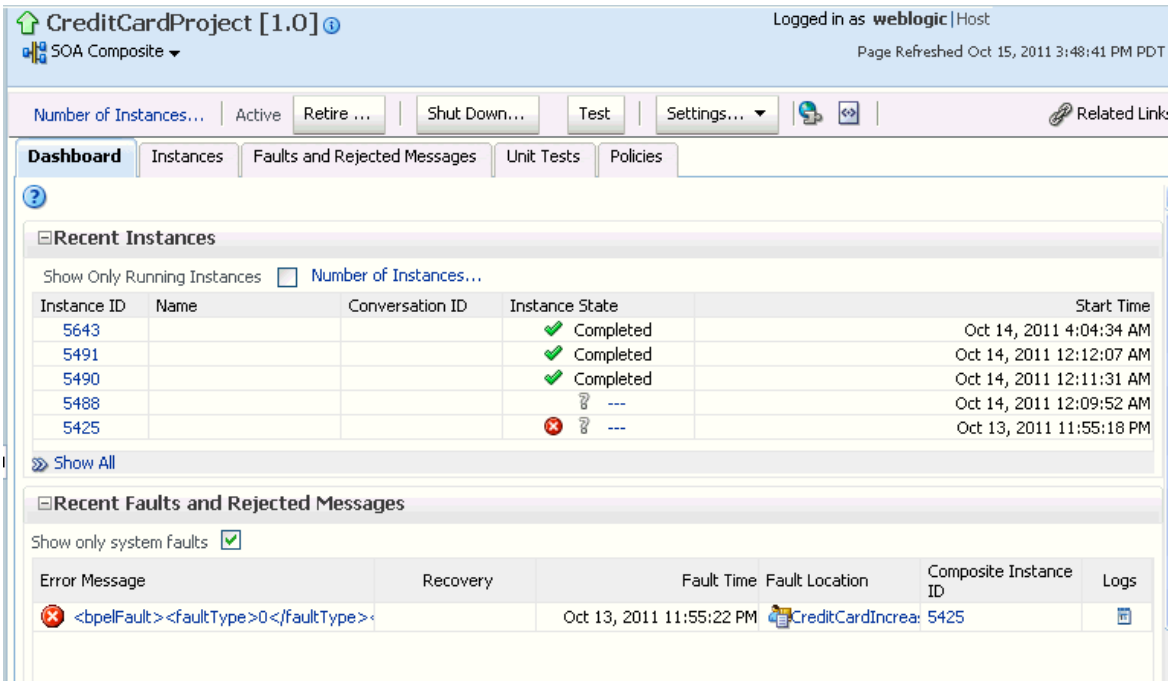
From the SOA Infrastructure home page, you can perform administration tasks such as monitoring SOA composite applications, monitoring individual composite instances, and updating the state of SOA composite applications and individual composite instances. You can also perform corrective actions such as fault recovery.

Figure 1–1 SOA Composite Applications Deployed in the SOA Infrastructure



You can click a specific SOA composite application in the **Composite** table to access its home page. Figure 1–2 shows the upper part of the home page for a SOA composite application. From the SOA composite application home page, you can perform administration tasks such as monitoring instances, recovering from faults, managing the state of application instances, and attaching policies. You can also perform a limited number of configuration tasks at the SOA composite application level, such as specifying the composite audit level and payload validation. These tasks are displayed as buttons at the top of the page.

Figure 1–2 SOA Composite Application Home Page (Upper Part)



If messages are awaiting recovery, a **BPEL Message Recovery Required** message is displayed at the top of the Dashboard page and all other SOA composite application home pages.

Figure 1–3 shows the lower part of the home page for this SOA composite application. The service components and service and reference binding components included in the composite are shown.

Figure 1–3 SOA Composite Application Home Page (Lower Part)

Component Metrics

Name	Component Type	Total Instances	Running Instances	Faulted Instances	
				Recoverable	Non Recoverable
Mediator1	Mediator	2	0	0	0
POProcessingInitiatorBPEL	BPEL	2	0	0	0

Services and References

Name	Type	Usage	Faults	Total Messages	Average Processing Time (sec)
poprocessinginitiatorbpel_client_ep	Web Service	Service	10	8	0.111
Service1	Web Service	Reference	0	2	0.027
POApprovalFileOut	JCA Adapter	Reference	0	2	0.005

For more information, see the following sections:

- Section 1.2.2, "Introduction to SOA Composite Applications"
- Part III, "Administering the SOA Infrastructure"
- Section 6.1, "Monitoring SOA Composite Application Recent Instances and Faults and Rejected Messages"

1.2.2 Introduction to SOA Composite Applications

SOA composite applications such as those shown in the Deployed Composites page in Figure 1–1 consist of the following:

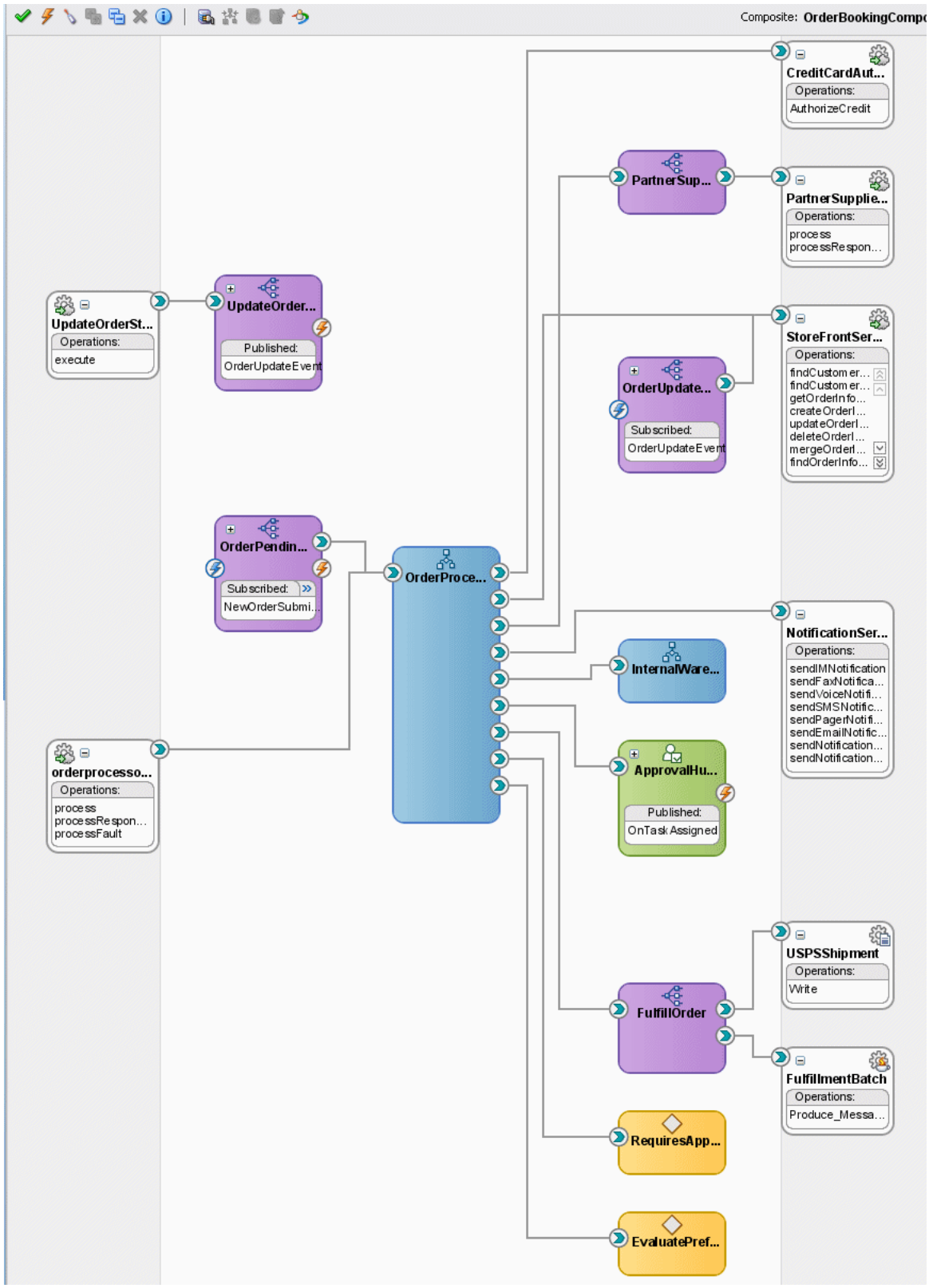
- Service components such as Oracle Mediator for routing, BPEL processes for orchestration, BPMN processes for orchestration (if Oracle BPM Suite is also installed), human tasks for workflow approvals, spring for integrating Java interfaces into SOA composite applications, and decision services for working with business rules.
- Binding components (services and references) for connecting SOA composite applications to external services, applications, and technologies.

These components are assembled into a single SOA composite application. Having the components assembled into one unit of deployment (the application) greatly simplifies the management and lifecycle of SOA applications.

Figure 1–4 provides an example of a SOA composite application in the SOA Composite Editor in Oracle JDeveloper. Service binding components (such as **orderprocessor_client_ep**) advertise their capabilities to external consumers. The service exposes a public interface of the SOA composite application (**OrderBookingComposite**) consisting of BPEL process, Oracle Mediator, human task, and decision service components. A wire connects the service to a specific component or reference in the composite. Reference binding components (such as **CreditCardAuthorizationService** and **PartnerSupplierService**) enable messages to be sent from the SOA composite application to external services. The service binding

components, service components, and reference binding components are wired (connected) for communication.

Figure 1–4 SOA Composite Application



The service components and binding components included in a SOA composite application appear in the lower part of an application home page, as shown in Figure 1–3 and Figure 1–5. The example in Figure 1–5 shows two service components in the **Component Metrics** section and three binding components in the **Services and References** section. You can click a specific service component or binding component to access its home page.

Figure 1–5 Service Components and Binding Components of a SOA Composite Application

Component Metrics

Name	Component Type	Total Instances	Running Instances	Faulted Instances	
				Recoverable	Non Recoverable
Mediator1	Mediator	2	0	0	0
POProcessingInitiatorBPEL	BPEL	2	0	0	0

Services and References

Name	Type	Usage	Faults	Total Messages	Average Processing Time (sec)
poprocessinginitiatorbpel_client_ep	Web Service	Service	10	8	0.111
Service1	Web Service	Reference	0	2	0.027
POApprovalFileOut	JCA Adapter	Reference	0	2	0.005

For more information, see the following documentation:

- Section 1.2.4, "Introduction to Service Components and Service Component Instances"
- Section 1.2.5, "Introduction to Binding Components"
- Part IV, "Administering SOA Composite Applications and Instances"
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

1.2.3 Introduction to SOA Composite Application Instances

When a SOA composite application is invoked, a new composite instance is created. This instance is identified by a unique instance ID that is displayed in pages of Oracle Enterprise Manager Fusion Middleware Control. For example, Figure 1–6 shows instance IDs displayed for SOA composite applications in the Instances page of the SOA Infrastructure. You can click these IDs to access more specific details about the state of SOA composite application instances. From the Instances page, you can also monitor the state of SOA composite application instances.

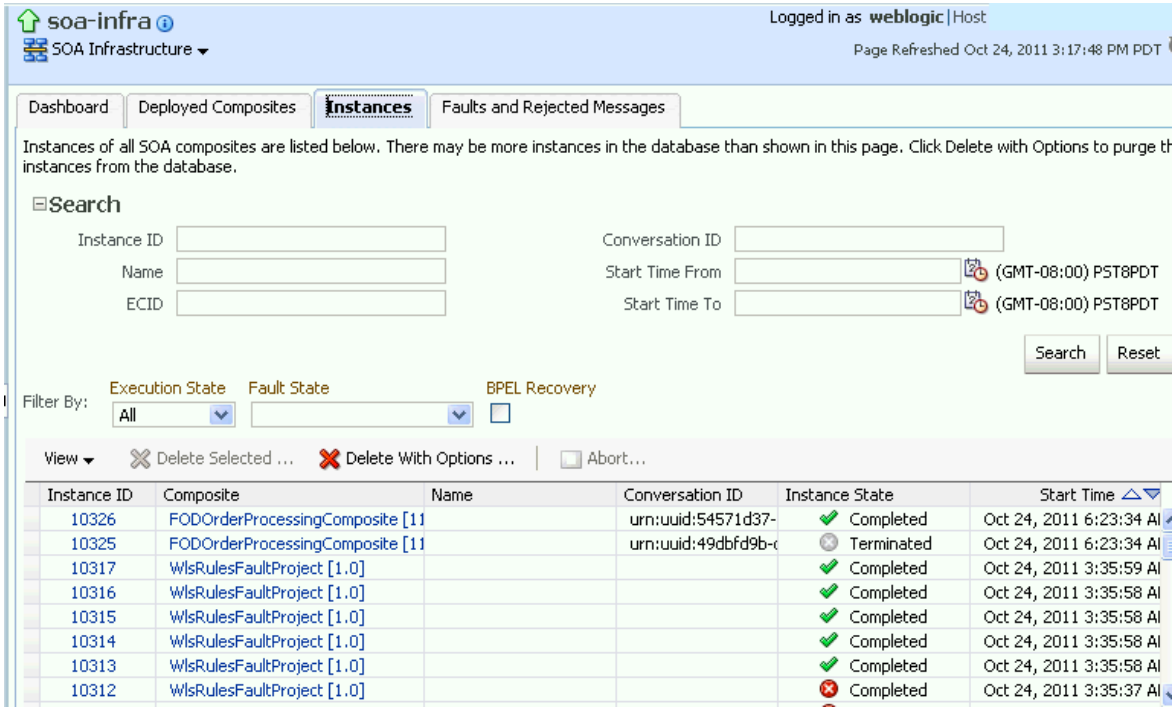
Instances that you create as unit tests from the Test Runs page are distinguished from those created automatically or created manually from the Test Web Service page by a little yellow box. This box is displayed to the left of the instance ID. This box is visible in both the Instances page and in the **Recent Instances** table of the Dashboard page of the SOA Infrastructure and SOA composite application.

For some SOA composite applications, conversation IDs are also generated. Conversation IDs provide another method for distinctly identifying a set of generated instances. As shown in Figure 1–6, conversation IDs are not automatically displayed for all instances. To see a conversation ID generated, perform one of the following tasks:

- Programatically invoke the service and pass a unique ID through a WS-Addressing header (`messageId`).
- Create an instance using the Test Web Service page. The only exception to this is when the **Enable Stress Test** checkbox of the **Additional Test Options** section of

the Test Web Service page is selected. In that case, a conversation ID is not created for the instance.

Figure 1–6 SOA Composite Application Instance IDs



For more information, see the following sections:

- Section 1.2.4, "Introduction to Service Components and Service Component Instances"
- Part IV, "Administering SOA Composite Applications and Instances"
- Section 8.1, "Initiating a SOA Composite Application Test Instance"

1.2.4 Introduction to Service Components and Service Component Instances

SOA composite applications include service components. Service components are the basic building blocks of SOA composite applications. Service components implement a part of the overall business logic of the SOA composite application.

The following service components can be used in a SOA composite application:

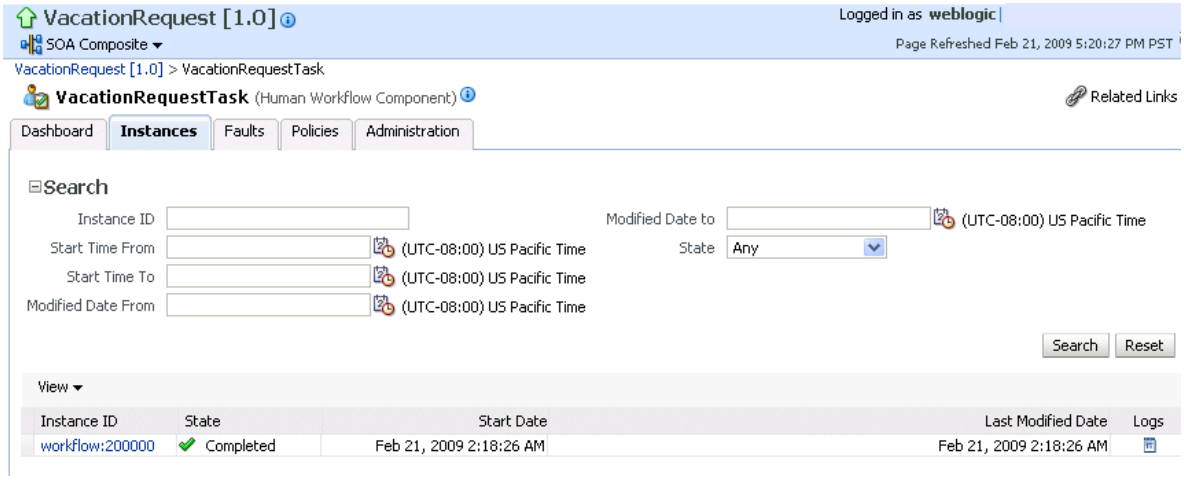
- BPEL process: For process orchestration of synchronous and asynchronous processes
- BPMN process (if Oracle BPM Suite is installed): For creating and modeling business processes using Business Process Management Notation and Modeling (BPMN)
- Oracle Mediator: For content transformation and routing events (messages) between service producers and consumers
- Human task: For modeling a human task (for example, manual order approval) that describes the tasks for users or groups to perform as part of an end-to-end business process flow
- Spring: For integrating Java interfaces into SOA composite applications

- Decision service: For making a decision or for processing based on business rules

From the service component home page in Oracle Enterprise Manager Fusion Middleware Control, you can perform administration tasks such as monitoring instances, recovering from faults, and attaching policies.

As described in Section 1.2.3, "Introduction to SOA Composite Application Instances," each application instance has its own instance ID. Each service component instance included in a SOA composite application instance also has its own instance ID that is displayed in Oracle Enterprise Manager Fusion Middleware Control. Figure 1–7 shows an instance ID (**workflow:200000**) displayed in the **Instance ID** column for the **VacationRequestTask** human task service component of the **VacationRequest** SOA composite application. You can monitor the state of that service component instance from the Instances page. You can also click this instance to access more specific details about the service component.

Figure 1–7 Service Component Instance IDs



Note: Oracle SOA Suite provides support for the spring service component. Note the following details about spring support in Oracle Enterprise Manager Fusion Middleware Control:

- There are no spring service engine management pages.
- A spring composite is displayed in the flow trace, but there is no audit trail for it.
- Spring composite metrics are shown in the composite application home page (for example, in the **Component Metrics** section of the Dashboard page).

For more information about administering service components, see the following sections:

- Part V, "Administering BPEL Process Service Components and Engines"
- Part VI, "Administering Oracle Mediator Service Components and Engines"
- Part VII, "Administering Decision Service Components and Business Rules Service Engines"

- Part VIII, " Administering Human Task Service Components and Human Workflow Service Engines"
- Part XV, "Administering Oracle BPMN Process Service Components and Engines"

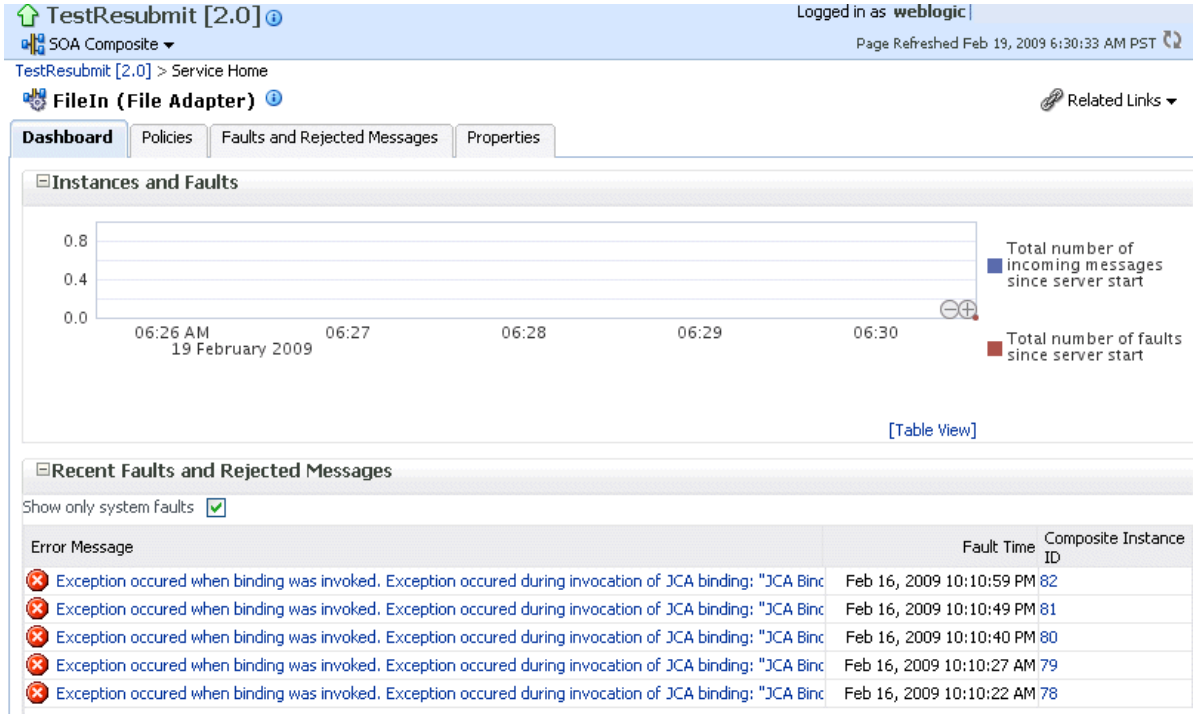
1.2.5 Introduction to Binding Components

Binding components connect SOA composite applications to external services, applications, and technologies (such as messaging systems or databases). Binding components are organized into two groups:

- Services: Provide the outside world with an entry point to the SOA composite application. The WSDL file of the service advertises its capabilities to external applications. The service bindings define how a SOA composite service can be invoked (for example, through SOAP).
- References: Enable messages to be sent from the SOA composite application to external services (for example, the same functionality that partner links provide for BPEL processes, but at the higher SOA composite application level).

In Oracle Enterprise Manager Fusion Middleware Control, you can perform binding component administration tasks such as attaching policies, monitoring rejected messages, and setting binding component properties. Figure 1–8 shows the home page of a service binding component (in this example, a JCA adapter).

Figure 1–8 Binding Components



For more information, see Section XIV, "Administering Binding Components."

1.2.6 Introduction to Service Engines

The SOA Infrastructure includes a set of service engines (BPEL process, human workflow, decision service, Oracle Mediator, and spring) that execute the business logic of their respective components within the SOA composite application (for

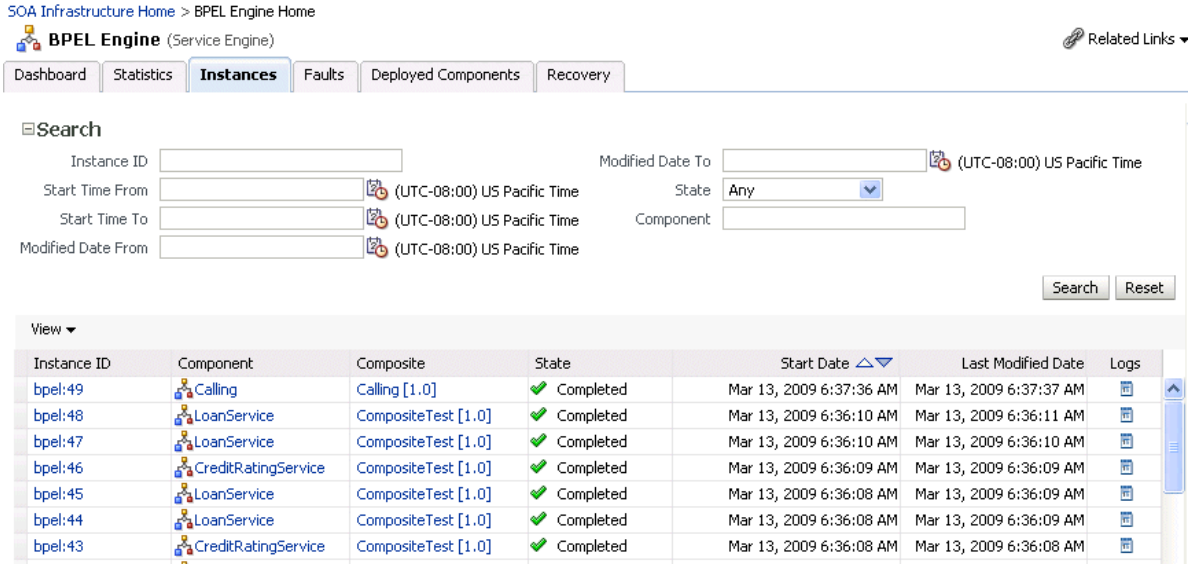
example, a BPEL process). If Oracle BPM Suite is installed, the SOA Infrastructure also includes the BPMN process service engine.

Figure 1–9 provides an example in Oracle Enterprise Manager Fusion Middleware Control of the BPEL process service engine. In this service engine, the **Calling**, **LoanService**, and **CreditRatingService** BPEL process service components run. Note the multiple instance IDs for **LoanService** and **CreditRatingService**. The BPEL process service components are included in two separate SOA composite applications:

- **Calling** is included in the **Calling** SOA composite application.
- **LoanService** and **CreditRatingService** are included in the **CompositeTest** SOA composite application.

However, each BPEL process service component runs in the *same* BPEL process service engine. You can click the links on the page to see more details about each BPEL process service component instance, the service component itself, or the SOA composite application in which it is included.

Figure 1–9 Service Components Running in a Service Engine



In Oracle Enterprise Manager Fusion Middleware Control, you can perform service engine administration tasks such as monitoring instances, recovering from faults, manually recovering (BPEL) failed messages, and configuring properties specific to a service engine. These configuration properties impact all service components that execute in the service engine, no matter the SOA composite application in which the service components are included. The service engine pages also include service engine-specific statistics and performance metrics.

For more information about administering service engines, see the following sections:

- Part V, "Administering BPEL Process Service Components and Engines"
- Part VI, "Administering Oracle Mediator Service Components and Engines"
- Part VII, "Administering Decision Service Components and Business Rules Service Engines"
- Part VIII, " Administering Human Task Service Components and Human Workflow Service Engines"
- Part XV, "Administering Oracle BPMN Process Service Components and Engines"

Note: Oracle Enterprise Manager Fusion Middleware Control does not include pages for managing the spring service engine.

1.2.7 Introduction to the Service Infrastructure

The service infrastructure provides the internal message transport infrastructure for connecting components and enabling data flow. The service infrastructure is responsible for routing messages along the wire connections between services, service components, and references.

For more information, see the following sections:

- Section 4.2, "Monitoring Message Delivery Processing Requests"
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite* for details about wiring

1.2.8 Introduction to the Contents of SOA Composite Applications

Your SOA composite application can consist of a variety of service components, binding components, and services that you administer from Oracle Enterprise Manager Fusion Middleware Control:

- BPEL processes
- BPMN processes (if Oracle BPM Suite is installed)
- Human workflows
- Oracle Mediator
- Decision services (Oracle Business Rules)
- Spring
- JCA adapters
- HTTP binding
- EJB service
- Direct binding service
- Oracle Application Development Framework (ADF) Business Component service
- Oracle BAM
- Oracle B2B
- Business events
- Oracle User Messaging Service

For conceptual information about these service components, binding components, and services, see *Oracle Fusion Middleware Getting Started with Oracle SOA Suite* and *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

1.3 What Is Oracle Business Process Management Suite?

Oracle BPM Suite provides an integrated environment for developing, administering, and using business applications centered around business processes.

Oracle BPM Suite provides the following:

- Enables you to create process models based on standards with user-friendly applications. It enables collaboration between process developers and process analysts. Oracle BPM supports BPMN 2.0 and BPEL from modeling and implementation to runtime and monitoring.
- Enables process analysts and process owners to customize business processes and Oracle Business Rules.
- Provides a web-based application for creating business processes, editing Oracle Business Rules, and task customization using predefined components.
- Expands business process management to include flexible, unstructured processes. It adds dynamic tasks and supports approval routing using declarative patterns and rules-driven flow determination.
- Enables collaboration with Process Space, which drives productivity and innovation.
- Unifies different stages of the application development lifecycle by addressing end-to-end requirements for developing process-based applications. Oracle BPM Suite unifies the design, implementation, runtime, and monitoring stages based on a service component architecture (SCA) infrastructure. This allows different personas to participate through all stages of the application lifecycle.

Oracle BPM Suite provides a seamless integration of all stages of the application development lifecycle from design-time and implementation to runtime and application management.

Oracle BPM Suite is layered on Oracle SOA Suite and shares many of the same product components, including:

- Oracle Business Rules
- Human workflow
- Oracle adapter framework for integration

1.4 Administration of Oracle SOA Suite and Oracle BPM Suite

You can perform a variety of Oracle SOA Suite and Oracle BPM Suite administration (configuration, monitoring, and management) tasks from Oracle Enterprise Manager Fusion Middleware Control. This section provides an overview of these tasks:

- Section 1.4.1, "Configuration of Oracle SOA Suite and Oracle BPM Suite"
- Section 1.4.2, "Monitoring of Oracle SOA Suite and Oracle BPM Suite"
- Section 1.4.3, "Management of Oracle SOA Suite and Oracle BPM Suite"

The administrative tasks that you can perform are based on the roles to which you are mapped; each role corresponds to a different set of privileges. Certain users can be mapped to simple monitoring privileges (for instance view-only access), while other users can be granted full access, including the ability to update configurations, restart servers, and so on. For more information about roles in Oracle Enterprise Manager Fusion Middleware Control, see Appendix C, "Oracle Enterprise Manager Roles."

Note: While this guide primarily describes how to use Oracle SOA Suite with Oracle WebLogic Server, most of the information is also applicable to using Oracle SOA Suite with other third-party application servers. However, there may be some differences with using third-party application servers.

For information about these differences, see *Oracle Fusion Middleware Third-Party Application Server Guide*.

1.4.1 Configuration of Oracle SOA Suite and Oracle BPM Suite

You can perform Oracle SOA Suite and Oracle BPM Suite configuration tasks in Oracle Enterprise Manager Fusion Middleware Control. Configuration tasks consist of setting properties such as audit levels and payload validation for your environment. Properties can be set at the following levels:

- SOA Infrastructure (impacting all SOA composite applications)
- Service engines (impacting all service components that execute in the service engine, no matter the SOA composite application in which they are included)
- SOA composite application (impacting all service components that are included in that composite application)
- Oracle B2B bindings
- Service and reference binding components message header properties

In terms of order of precedence, inherited SOA composite application property settings (such as audit level settings and payload validation) take the highest precedence, followed by service engine settings, followed by SOA Infrastructure settings. However, most properties do not have this type of precedence to consider.

1.4.1.1 Introduction to the Order of Precedence for Audit Level Settings

Audit tracking enables you to select the level of information to be collected by the message tracking infrastructure. Audit level tracking can be set at the following levels:

- BPEL process or BPMN process service component
- SOA composite application
- Service engine
- SOA Infrastructure

If you set audit tracking at multiple levels, it is important to understand which setting takes precedence. Table 1–1 provides examples of the order of precedence for audit level settings.

Table 1–1 Examples of Order of Precedence

Component	Composite	Service Engine	SOA Infrastructure	Which Setting Takes Precedence?
No property	Off	Production	Development	Composite. The audit level is set to **Off**. The service engine and SOA Infrastructure audit levels do not take effect.

Table 1–1 (Cont.) Examples of Order of Precedence

Component	Composite	Service Engine	SOA Infrastructure	Which Setting Takes Precedence?
No property	Inherit	Development	Production	Service engine. The audit level is set to **Development**. The payload is shown in the assign activity. The SOA Infrastructure audit level does not take effect.
No property	Inherit	Inherit	Production	SOA Infrastructure. The audit level is set to **Production**.
No property	Inherit	Production/ Development /Off/Inherit	Off	The overall audit is not shown. The composite inherits the audit level from the SOA Infrastructure. The payload is shown in the assign activity based on the service engine audit level setting.
Development	Off	Production	Development	Composite. Since the composite audit level is set to **Off**, the overall audit is not shown. The service engine audit level is shown, but the **Development** setting for the component takes precedence. The payload is shown in the assign activity based on the component audit level setting of **Development**.
Inherit	Off	Production	Development	Composite. Since the composite audit level is set to **Off**, the overall audit is not shown. The service engine audit level is not shown because **Off** is inherited from the composite.

Notes:

- When the composite audit level is set to **Off**, there is no audit trail generated for this composite and all service engines used within the composite.
- When the composite audit level is set to **Inherit**, it always inherits the settings of the SOA Infrastructure.
- When the composite audit level is set to **Off**, the component inherits the service engine settings.

For more information, see the following sections:

- Chapter 3, "Configuring the SOA Infrastructure"
- Section 12.1, "Configuring BPEL Process Service Engine Properties"
- Section 12.5, "Setting the Audit Level at the BPEL Process Service Component Level"

- Chapter 20, "Configuring Human Workflow Service Components and Engines"
- Chapter 34, "Configuring Service and Reference Binding Components"
- Section 37.1, "Configuring BPMN Process Service Engine Properties"

1.4.2 Monitoring of Oracle SOA Suite and Oracle BPM Suite

You can perform Oracle SOA Suite and Oracle BPM Suite monitoring tasks in Oracle Enterprise Manager Fusion Middleware Control, including monitoring the following:

- Instances, faults, and rejected messages in the SOA Infrastructure, SOA composite applications, service components, service engines, and service and reference binding components.
- Service engine, service infrastructure, and binding component processing request performance.
- Service and reference binding component message processing totals and average processing times.
- Audit trail and process flow behavior in service components. For BPMN processes, the entire BPMN process flow is displayed, and the path taken by the process instance is highlighted.
- Service engine request and thread states in BPEL processes, BPMN processes, and human workflows.

Note: You can also monitor and diagnose problems in Oracle SOA Suite through use of the WebLogic Diagnostic Framework (WLDF) and Diagnostics Framework. For more information, see Chapter 11, "Diagnosing Problems with SOA Composite Applications."

1.4.3 Management of Oracle SOA Suite and Oracle BPM Suite

You can perform Oracle SOA Suite and Oracle BPM Suite management tasks in Oracle Enterprise Manager Fusion Middleware Control, including managing the following:

- Creation and deletion of partitions. Once you create partitions, you can deploy a composite to the appropriate partition. This action enables you to logically group SOA composite applications into partitions. This is similar to the concept of domains in the 10.1.*x* releases of Oracle BPEL Process Manager.
- Composite state (activating, retiring, starting, stopping, and setting the default composite version).
- Deletion and termination of composite instances.
- Deployment, undeployment, and redeployment actions for SOA composite applications.
- Export of a deployed SOA composite application to a JAR file.
- Manual initiation of SOA composite application test instances from the Test Web Service page.
- Recovery from faults in SOA composite applications, service components, service engines, and business events.
- Manual recovery of failed messages in BPEL processes.
- Automated unit testing of SOA composite applications.

- Attachment of policies to SOA composite applications, service components, and binding components.
- Incoming and outgoing notification messages in human workflow.
- Subscriptions to business events and testing of event publications.
- Publication of web services to the Universal Description, Discovery, and Integration (UDDI) registry.
- Disabling of business monitors (BPEL sensors, BPEL monitors, and BPMN measurements).
- Storage of instance and callback message data in Oracle Coherence distributed cache on Oracle Exalogic platforms.
- Management of the Oracle SOA Suite plug-in introspected by Oracle Virtual Assembly Builder. Oracle Virtual Assembly Builder is a tool for virtualizing installed Oracle components, modifying these components, and deploying them to an environment. Using Oracle Virtual Assembly Builder, you capture the configuration of existing software components in artifacts called software appliances that can then be grouped and their relationships defined into artifacts called software assemblies. For more information about the Oracle SOA Suite plug-in, see *Oracle Virtual Assembly Builder User's Guide*.

The following sections provide a more specific overview of several management tasks:

- Section 1.4.3.1, "Introduction to Fault Recovery"
- Section 1.4.3.2, "Introduction to Policies"
- Section 1.4.3.3, "Introduction to the Lifecycle State of SOA Composite Applications"
- Section 1.4.3.4, "Introduction to SOA Composite Application Automated Testing"
- Section 1.4.3.5, "Introduction to Partitioning of the SOA Infrastructure"

Notes:

- Backup and recovery of Oracle SOA Suite is described in *Oracle Fusion Middleware Administrator's Guide*.
- GridLink data sources and multidata sources protect the SOA Infrastructure against database failures. You typically configure GridLink and multidata sources during system setup (defining multipools directly at installation time). When an Oracle Real Application Clusters (Oracle RAC) database instance fails, the connections are reestablished with available database instances. For more information about Gridlink and Oracle SOA Suite, see "Configuring High Availability for Oracle SOA Suite" of *Oracle Fusion Middleware High Availability Guide*.

1.4.3.1 Introduction to Fault Recovery

You can perform fault recovery actions on BPEL process, BPMN process, Oracle Mediator, human workflow, and business event subscription faults (which include database and component subscription faults) identified as recoverable in Oracle Enterprise Manager Fusion Middleware Control. The following types of fault recovery are supported.

- Recovery from individual faults, where you have access to the most granular recovery options specific to each type of fault
- Recovery from multiple (bulk) faults, where you select multiple faults for recovery

You can perform individual and bulk recovery actions on recoverable faults at the following levels:

- Faults occurring in all SOA composite applications in the SOA Infrastructure
- Faults occurring in an individual SOA composite application
- Faults occurring in service components
- Faults occurring in service engines
- Faults occurring in business events

You perform fault recovery on faults identified as recoverable in Oracle Enterprise Manager Fusion Middleware Control. For BPEL process faults to be identified as recoverable, there must be a fault policy defined that is bound to the fault (through the `fault-bindings.xml` file) and which triggers the action `ora-human-intervention`. A BPEL component fault can be recovered in only this case. If no fault policy is defined as part of the composite, then a recoverable BPEL process fault is not possible.

You define a fault recovery policy in the `fault-policies.xml` and `fault-bindings.xml` files outside of Oracle Enterprise Manager Fusion Middleware Control. These files are packaged with the SOA composite application that you deploy to the SOA Infrastructure and administer in Oracle Enterprise Manager Fusion Middleware Control.

Oracle Mediator and human workflow faults do not have the same behavior; they can create recoverable faults without any fault policy. For errors in human task service components or human workflow service engines, you perform fault recovery on faults identified as recoverable from the Oracle BPM Worklist.

The following types of faults can be displayed in Oracle Enterprise Manager Fusion Middleware Control:

- Business: Application-specific faults that are generated when there is a problem with the information being processed (for example, a social security number is not found in the database).
- System: Network errors or other types of errors such as a database server or a web service being unreachable.
- Oracle Web Service Manager (OWSM): Errors on policies attached to SOA composite applications, service components, or binding components. Policies apply security to the delivery of messages.

Faults can also be classified as either of the following:

- Recoverable or nonrecoverable:

  Only certain types of faults are identified as recoverable. Table 1–2 provides examples of several recoverable and nonrecoverable faults.

- Rejected Messages:

  A fault is classified as a rejected message based on where it occurs. If a fault occurs before entering a SOA composite, without generating a composite instance, it is classified as a rejected message. A system or a policy fault can be identified as a rejected message.

Table 1–2 Faults

Recoverable Faults	Nonrecoverable Faults
▪ Business faults and some specific system faults ▪ Oracle Mediator input file path and output directory mismatch ▪ An Oracle BPM Worklist user is not authorized to perform relevant (expected) actions	▪ Rejected messages ▪ Most system faults ▪ Non-existent references ▪ Service invocation failures ▪ Policy faults

For more information on performing fault recovery, see the following sections:

- Section 8.4, "Recovering from SOA Composite Application Faults at the SOA Infrastructure Level"
- Section 8.5, "Recovering from SOA Composite Application Faults in the Application Home Page"
- Section 14.1, "Recovering from BPEL Process Service Component Faults"
- Section 14.3, "Recovering from BPEL Process Service Engine Faults"
- Section 17.3, "Recovering From Oracle Mediator Service Engine Faults"
- Section 22.2, "Recovering from Human Workflow Service Engine Faults"
- Section 22.4, "Recovering from Human Task Service Component Faults"
- Section 33.4, "Recovering from Business Event Faults"
- Section 39.1, "Recovering from BPMN Process Service Component Faults"
- Section 39.3, "Recovering from BPMN Process Service Engine Faults"

1.4.3.2 Introduction to Policies

You can attach and detach policies at the following levels in Oracle Enterprise Manager Fusion Middleware Control:

- SOA composite applications
- Service components
- Service and reference binding components

Policies apply security to the delivery of messages. Oracle Fusion Middleware uses a policy-based model to manage web services. The following types of policies are supported:

- Security: Implements WS-Security 1.0 and 1.1 standards. They enforce authentication and authorization of users, identity propagation, and message protection (message integrity and message confidentiality).
- Reliable Messaging: Supports the WS-ReliableMessaging protocol, guaranteeing the end-to-end delivery of messages.
- Message Transmission Optimization Mechanism (MTOM): Ensures that attachments are in MTOM format, a format for efficiently sending binary data to and from web services.
- WS-Addressing: Verifies that SOAP messages include WS-Addressing headers in conformance with the WS-Addressing specification. Transport-level data is

included in the XML message rather than relying on the network-level transport to convey this information.

- Management: Logs request, response, and fault messages to a message log. Management policies can include custom policies.

Policies are part of an enterprise policy framework that allows policies to be centrally created and managed.

For more information, see the following documentation:

- Section 7.6, "Managing SOA Composite Application Policies"
- Section 14.2, "Managing BPEL Process Service Component Policies"
- Section 22.1, "Managing Human Task Service Component Policies"
- Section 36.1, "Managing Binding Component Policies"
- Section 39.2, "Managing BPMN Process Service Component Policies"
- *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for definitions of available policies and details about which policies to use for your environment

1.4.3.2.1 Introduction to How Policies are Executed Policies are executed *before* a message reaches the component with the attached policy. This causes the error to be displayed in the component preceding the component with the attached policy. For example:

- A policy attached to an Oracle Mediator service component is executed on the wire before the message is passed to the Oracle Mediator. This causes the fault to be displayed in the service binding component instead of the Oracle Mediator.
- A policy attached to a human task service component is executed in the preceding BPEL process service component before the message is passed to the human task service component. This causes the fault to be displayed in the BPEL process service component instead of the human task service component.
- A policy attached to a human task service component is executed inside the BPMN process in the human steps associated with the human service component before the message is passed to the human task service component. This causes the fault to be displayed in the BPMN process service component instead of the human task service component.

To see the exact location of the policy error, view the audit trail.

1.4.3.3 Introduction to the Lifecycle State of SOA Composite Applications

You can administer the lifecycle state of deployed SOA composite applications from Oracle Enterprise Manager Fusion Middleware Control. An application is automatically activated when you deploy it to the SOA Infrastructure. During deployment, you can specify a specific revision number for the application. A revision is a specific deployed version of the application. You can deploy multiple revisions of an application, enabling all to run at the same time.

This is a key benefit of revisions. For example, you may have an older revision of an application running with one customer that is still valid. You then begin a partnership with a different customer that requires a slight modification to the design of the application. At some point, you plan to migrate the old customer to the newer revision of the application, but for now that is not necessary. Revisions enable you to run both applications.

The revision value is added to the application name in Oracle Enterprise Manager Fusion Middleware Control. For example, in Figure 1–1, revision 1.0 is the version for many deployed SOA composite applications. If a new request comes in for a specific composite application revision, that composite application revision is invoked. If a new request comes in without specifying a revision, the default revision is invoked. A small green dot distinguishes the default revision from other revisions.

You can perform the following lifecycle administration tasks on a SOA composite application from Oracle Enterprise Manager Fusion Middleware Control:

- Create an instance.
- Stop and restart application revisions. An application revision is typically started instantly after deployment.
- Retire and activate application revisions. Application revisions are instantly activated upon deployment.
- Set an application as the default version.
- Deploy, undeploy, and redeploy application revisions.
- Delete specific instances of an application revision.

With the addition of Oracle SOA Governance tools for lifecycle management, you can perform additional lifecycle management tasks on a SOA composite application, or any component or service within the composite:

- Collect important information on each component in an Oracle Enterprise Repository to help producers, providers, consumers, or other participants in the lifecycle for better understanding. For example, you can show the relationships between the previous and next versions.
- Associate a lifecycle stage categorization to components or service endpoints (for example, build, test, stage, or production).
- Automatically advance and track components and service endpoints through various lifecycle stages, automatically publishing them to an appropriate UDDI service registry for their lifecycle stage.
- Manage their lifecycle and associated approvals using repeatable processes.
- Manage their performance in production, and inform prospective consumers of services for better design-time decisions.

SOA Governance Suite provides Oracle SOA Suite and Oracle BPM Suite users with options to specify and automate a complete lifecycle for applications and their components (for example, planning, design, implementation, testing, staging, production, changes, and retirement).

For more information about administering the lifecycle states of a SOA composite application and SOA governance, see the following sections:

- Section 8.1, "Initiating a SOA Composite Application Test Instance"
- Section 7.4, "Managing the State of Deployed SOA Composite Applications"
- Section 8.2, "Monitoring and Deleting SOA Composite Application Instances from the Application Home Page"
- Section 8.3, "Monitoring and Deleting SOA Composite Application Instances at the SOA Infrastructure Level"
- *Oracle Fusion Middleware User Guide for Oracle Enterprise Repository*
- *Oracle Fusion Middleware Integration Guide for Oracle Enterprise Repository*

- *Oracle Fusion Middleware Configuration Guide for Oracle Enterprise Repository*

1.4.3.4 Introduction to SOA Composite Application Automated Testing

You can create, deploy, and run test cases that automate the testing of SOA composite applications. Test cases enable you to simulate the interaction between a SOA composite application and its references before deployment in a production environment. Test suites consist of a logical collection of one or more test cases. Each test case contains a set of commands to perform as the test instance is executed. The execution of a test suite is known as a test run. Each test corresponds to a single SOA composite application instance. You can also create BPEL process service component test cases in the SOA composite application test case. Instances generated by the execution of these tests are distinguished as test instances by a little yellow box next to their instance ID on the Instances page of the SOA Infrastructure.

The test suite framework provides the following features:

- Uses emulations to simulate the behavior of components with which your SOA composite application interacts during execution. Instead of invoking a specific component, you can specify a response from the component.
- Uses assertions to validate data during process execution.

For information about the following:

- Creating and running test cases, see Section 7.5, "Automating the Testing of SOA Composite Applications"
- Designing test cases for SOA composite applications, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

1.4.3.5 Introduction to Partitioning of the SOA Infrastructure

You can deploy SOA composite applications into separate sections of the SOA Infrastructure known as partitions. Deploying to partitions enables you to logically group SOA composites and perform bulk lifecycle management tasks on large numbers of composites. Partitioning is similar to the concept of domains in the 10.1.*x* releases of Oracle BPEL Process Manager. However, note that you *cannot* perform specific configuration tasks on partitions, such as restricting login access to a specific partition or configuring partitions (such as configuring threading).

At least one partition is required for deploying SOA composite applications. A default partition named **default** is automatically included with Oracle SOA Suite.

Once you create a partition, you can perform the following tasks:

- Deploy SOA composite applications into the partition using Oracle Enterprise Manager Fusion Middleware Control, Oracle JDeveloper, WebLogic Scripting Tool (WLST) commands, or `ant` commands.
- Access the partition and its deployed composites through the navigation tree.
- Perform the following bulk lifecycle management tasks on the composites in a specific partition:
  - Start all composites
  - Shut down all composites
  - Undeploy all composites
  - Retire all composites
  - Activate all composites

– List all composites

Note: Partitions are *not* associated with a particular state such as started, stopped, activated, or retired. Only the composites within the partition are associated with a particular state. Therefore, you *cannot* start, stop, activate, or retire a partition.

For more information, see Section 7.8, "Grouping SOA Composite Applications into Partitions."

1.5 Administration for Application Developers

If your role is that of an application developer, manage and test SOA composites using a combination of Oracle JDeveloper and Oracle Enterprise Manager Fusion Middleware Control. See the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite* to develop SOA composite applications with Oracle JDeveloper, and refer to the following sections to deploy, monitor, and initiate a test instance of the composite application with Oracle Enterprise Manager Fusion Middleware Control:

- Chapter 6, "Monitoring SOA Composite Applications"
- Chapter 7, "Deploying and Managing SOA Composite Applications"
- Section 8.1, "Initiating a SOA Composite Application Test Instance"

To create and model business processes with Oracle BPM Suite, see *Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management*.

Part II

Getting Started with Administration

This part describes how to navigate to Oracle SOA Suite and Oracle BPM Suite administration tasks in Oracle Enterprise Manager Fusion Middleware Control.

This part includes the following chapter:

- Chapter 2, "Getting Started with Administering Oracle SOA Suite and Oracle BPM Suite"

2

Getting Started with Administering Oracle SOA Suite and Oracle BPM Suite

This chapter describes how to log in to and navigate the menus of Oracle Enterprise Manager Fusion Middleware Control to perform Oracle SOA Suite and Oracle BPM Suite configuration, monitoring, and management tasks. It also describes how to access Oracle WebLogic Server Administration Console from Oracle Enterprise Manager Fusion Middleware Control.

This chapter includes the following topics:

- Section 2.1, "Logging In to Oracle Enterprise Manager Fusion Middleware Control"
- Section 2.2, "Navigating to Oracle SOA Suite and Oracle BPM Suite Administration Tasks"
- Section 2.3, "Logging Out of Oracle Enterprise Manager Fusion Middleware Control"

For more information about service engines, service components, binding components, and the SOA Infrastructure, see Chapter 1, "Introduction and Concepts."

2.1 Logging In to Oracle Enterprise Manager Fusion Middleware Control

This section describes how to log in to Oracle Enterprise Manager Fusion Middleware Control.

To log in to Oracle Enterprise Manager Fusion Middleware Control:

1. Use Microsoft Internet Explorer 7, Mozilla Firefox 2.0.0.2, Apple Safari 4.0, or Mozilla Firefox 3.0.*x* to access the following URL:

   `http://host_name:port/em`

   Where *host_name* is the name of the host on which Oracle Enterprise Manager Fusion Middleware Control is installed and *port* is a number that is dynamically set during installation. This port is typically `7001`, but is the HTTP port associated with Oracle HTTP Server. For environments in which the SSL port was enabled during configuration, the default port is `7002`.

2. Enter `weblogic/password` and click **Login**.

   where:

   - `weblogic` is the default administrator user name for Oracle Enterprise Manager Fusion Middleware Control (you can change this during installation).

- *password* is the password you entered during Oracle SOA Suite installation.

The Accessibility Preference dialog appears the first time you log in. If you want, you can select to not display this dialog again.

3. Select an appropriate action and click **Continue**.

   The farm home page is displayed. From there, you can navigate to Oracle SOA Suite and Oracle BPM Suite in several different ways, as described in the following sections.

For more information about installation, see *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

2.2 Navigating to Oracle SOA Suite and Oracle BPM Suite Administration Tasks

This section describes methods for navigating to Oracle SOA Suite and Oracle BPM Suite administration tasks in Oracle Enterprise Manager Fusion Middleware Control:

- Section 2.2.1, "Navigating Through the SOA Infrastructure Home Page and Menu"
- Section 2.2.2, "Navigating Through the SOA Composite Application Home Page and Menu"
- Section 2.2.3, "Navigating Through the Partition Home Page and Menu"
- Section 2.2.4, "Navigating to Deployed Java EE Applications"
- Section 2.2.5, "Navigating to the Oracle WebLogic Server Administration Console and Other Pages"
- Section 2.2.6, "Navigating to the SOA Infrastructure or SOA Composite Application Home Page from the Farm Home Page"

Note: The **Farm** menu is always displayed at the top of the navigator. As you expand the **SOA** folder in the navigator and click the links displayed beneath it, the **SOA Infrastructure** menu becomes available at the top of the page.

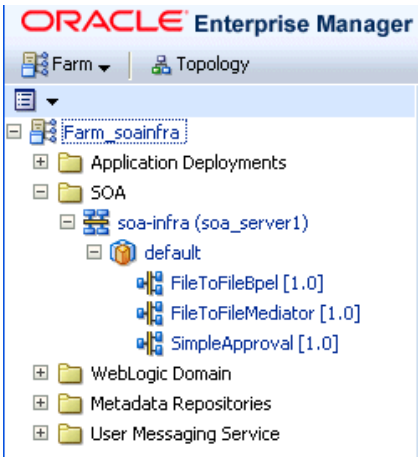
2.2.1 Navigating Through the SOA Infrastructure Home Page and Menu

You can navigate to Oracle SOA Suite and Oracle BPM Suite administration tasks through the SOA Infrastructure home page and menu. The SOA Infrastructure provides you with access to all deployed SOA composite applications, service engines, service components, business events, and other elements.

To navigate through the SOA Infrastructure home page and menu:

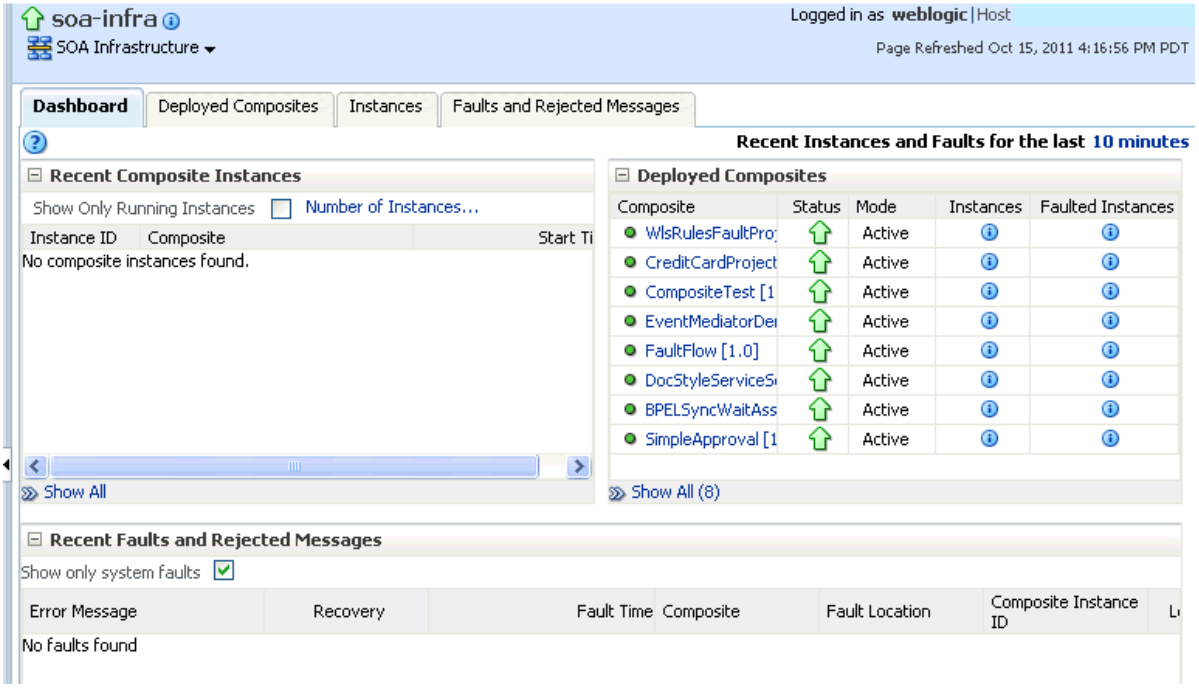
1. Expand **SOA** > **soa-infra** in the navigator.
2. Expand the specific partition (for example, **default**).

   This displays all SOA composite applications running in the selected partition of the SOA Infrastructure for that managed server. Composites are grouped into the partition in which they are deployed.



3. Click **soa-infra**.

   This displays the Dashboard page of the SOA Infrastructure. Click the help icon at the top of this page to access the Resource Center for the entire Oracle SOA Suite. The upper part of the page displays details about recently deployed SOA composite application instances, deployed composites, recent faults, and rejected messages. You can click a specific SOA composite application name or instance ID to access additional details. You can also click **Show All** at the bottom of each section to see more information about all items.



   The lower part of the page displays details about the service engines provided by the SOA Infrastructure and a graphical representation of the number of instances and faults for all SOA composite applications hosted in the SOA Infrastructure. You must expand these sections to see this information. Click a service engine name to access more specific details.

4. Note that the **SOA Infrastructure** menu appears below the **soa-infra** name at the top of the page.

Note: Depending upon your current location, the context of this menu changes to provide you with the administrative options most relevant to your current location. For example, when you are within the pages of a SOA composite application, the **SOA Composite** menu is displayed, or on the home page of a specific partition, the **SOA Partition** menu is displayed.

5. Select the **SOA Infrastructure** menu.

These administrative options enable you to perform the following tasks:

Option	Description
Home	This option displays the following details for the SOA Infrastructure Dashboard page: ▪ Recent SOA composite application instances. ▪ Deployed SOA composite details (status, instance, and fault counts). ▪ Recent faults and rejected messages in all SOA composite application instances, including whether the instance is recoverable. ▪ Number of components hosted by service engines and the faults in those components. ▪ Graphical representation of the number of instances and faults for all SOA composite applications hosted in the SOA Infrastructure. For more information, see Section 4.1, "Monitoring SOA Infrastructure Recent Instances and Faults and Deployed Composites."
Monitoring	This option displays the following details: ▪ A summary of performance statistics in the SOA infrastructure. ▪ Request processing details that show the breakup of time spent in handshaking requests between the binding components, service infrastructure, and service engines. For more information, see Section 4.2, "Monitoring Message Delivery Processing Requests."

Option	Description
Logs	This option enables you to view and configure the logging levels for runtime loggers. For more information, see Section 3.4, "Configuring Log Files" and Section B.1, "Setting Logging Levels for Troubleshooting."
SOA Deployment	This option enables you to deploy, undeploy, or redeploy SOA composite applications. For more information, see Chapter 7, "Deploying and Managing SOA Composite Applications."
Manage Partitions	This option enables you to logically group your SOA Infrastructure into separate sections known as partitions in which you deploy your SOA composite applications. This helps you to logically group composites so that you can perform bulk lifecycle management tasks on large numbers of composites. For more information, see Section 1.4.3.5, "Introduction to Partitioning of the SOA Infrastructure" and Section 7.8, "Grouping SOA Composite Applications into Partitions."
Service Engines	This option provides access to monitoring and management tasks for the BPEL process, Oracle BPMN, Oracle Mediator, human workflow, and business rules service engines.
Bindings	This option displays details about recently active document types and trading partners, and inbound and outbound end points for Oracle B2B. For more information, see Section 32.1, "Monitoring the Oracle B2B Infrastructure."
Services and References	This option displays message processing metrics for service and reference binding components in all SOA composite applications. For more information, see Section 4.3, "Monitoring Service and Reference Binding Components in the SOA Infrastructure."
Business Events	This option displays available business events, current event subscribers, and fault details. For more information, see Chapter 33, "Managing Business Events."

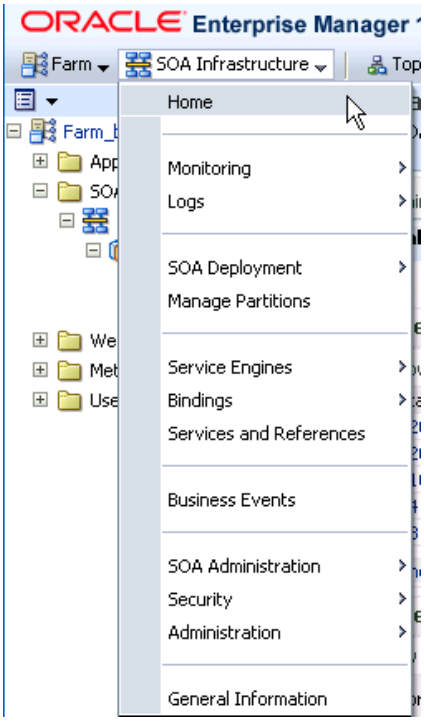
Option	Description
SOA Administration	This option provides access to the following configuration tasks for the SOA Infrastructure and each service engine: ■ **Common Properties**: For setting properties that impact the entire SOA Infrastructure, such as viewing and setting the SOA Infrastructure audit level, capturing the state of the SOA composite application instance, enabling the payload validation of incoming messages, specifying the callback server and server URLs, setting Universal Description, Discovery and Integration (UDDI) registry properties, viewing the data source JNDI locations, setting the nonfatal connection retry count, setting web service binding properties, and optimizing the loading performance of pages. For more information, see Section 3.1, "Configuring SOA Infrastructure Properties." ■ **BPEL Properties**: For setting the audit trail size, maximum document size for a variable, payload validation for incoming and outgoing messages, audit trail level, BPEL monitor and sensor disabling status, and dispatcher thread level for invoke message threads, system threads, and service engine threads. For more information, see Section 12.1, "Configuring BPEL Process Service Engine Properties." ■ **BPMN Properties**: For setting the audit trail size, maximum document size for a variable, payload validation for incoming and outgoing messages, audit trail level, and dispatcher thread level for invoke message threads, system threads, and service engine threads. For more information, see Section 37.1, "Configuring BPMN Process Service Engine Properties" ■ **Mediator Properties**: For setting the audit level, metrics level, number of parallel worker threads, number of maximum rows retrieved for parallel processing, parallel thread sleep values, error thread sleep values, container ID refresh time, and container ID lease timeout values. For more information, see Section 15.1, "Configuring Oracle Mediator Service Engine Properties." ■ **Workflow Config**: Displays the following tabs: **Mailer** tab: For setting the workflow service notification mode and actionable email address value. For more information, see Section 20.1, "Configuring Human Workflow Notification Properties." **Task** tab: For setting the actionable email account, adding the worklist application URL, selecting the pushback assignee, adding portal realm mapping, and adding the task auto release configuration priority. For more information, see Section 20.2, "Configuring Human Workflow Task Service Properties." ■ **B2B Server Properties**: For enabling Dynamic Monitoring Service (DMS) metrics. ■ **Cross References**: For selecting cross-reference values. For more information, see Section 18, "Managing Cross-References."
Security	This option displays the following selections: ■ **Application Policies**: For creating application policies that an application relies upon for controlling access to resources. ■ **Application Roles**: For creating application roles for applications. For more information about application roles, see *Oracle Fusion Middleware Application Security Guide*. This option is available for all deployed Java EE applications, including the SOA Infrastructure (**soa-infra**) application. These options do *not* configure security policies for SOA composites. For more information about attaching policies to composite applications, see Section 7.6, "Managing SOA Composite Application Policies."

Option	Description
Administration	This option displays the following selections: ▪ **MDS Connections**: For managing metadata service (MDS) connections. ▪ **TopLink Sessions**: For managing cache for the Oracle TopLink persistence framework. ▪ **System MBean Browser**: For advanced configuration of properties across applications and components. For more information, see *Oracle Fusion Middleware Administrator's Guide* for instructions on using the System Mean Browser. In addition to configuring the System MBean Browser from Oracle Enterprise Manager Fusion Middleware Control, you also can invoke SOA configuration MBean operations with the WebLogic Scripting Tool (WLST). For information, see *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*. **Note:** The System MBean Browser includes an MBean for Oracle BPMN, even if Oracle BPM Suite is not installed. If Oracle BPM Suite is not installed, this MBean cannot be used, and should be ignored. The presence of this MBean does not impact the runtime behavior of other Oracle SOA Suite components.
General Information	This option displays general details about the SOA Infrastructure, such as the Oracle Enterprise Manager Fusion Middleware Control version, Oracle home, and Oracle instance.

Note: Starting with 11*g* Release 1 (11.1.1.4.0), you can no longer stop and start the SOA Infrastructure from the **SOA Infrastructure** menu.

6. Select a specific SOA composite application from the **soa-infra** list shown in Step 2.

   The **SOA Infrastructure** menu is now displayed above the navigator, providing you with access to the same high-level administrative options, even when you are within the pages of a SOA composite application. This menu changes positions when you select a composite application.



7. Right-click **soa-infra** in the navigator. The menu that is displayed provides you with access to the same administrative options that appear in the **SOA Infrastructure** menu.

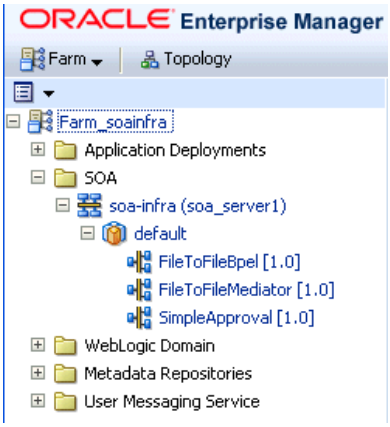
2.2.2 Navigating Through the SOA Composite Application Home Page and Menu

You can navigate directly to Oracle SOA Suite and Oracle BPM Suite administration tasks for a specific SOA composite application.

To navigate through the SOA composite application home page and menu:

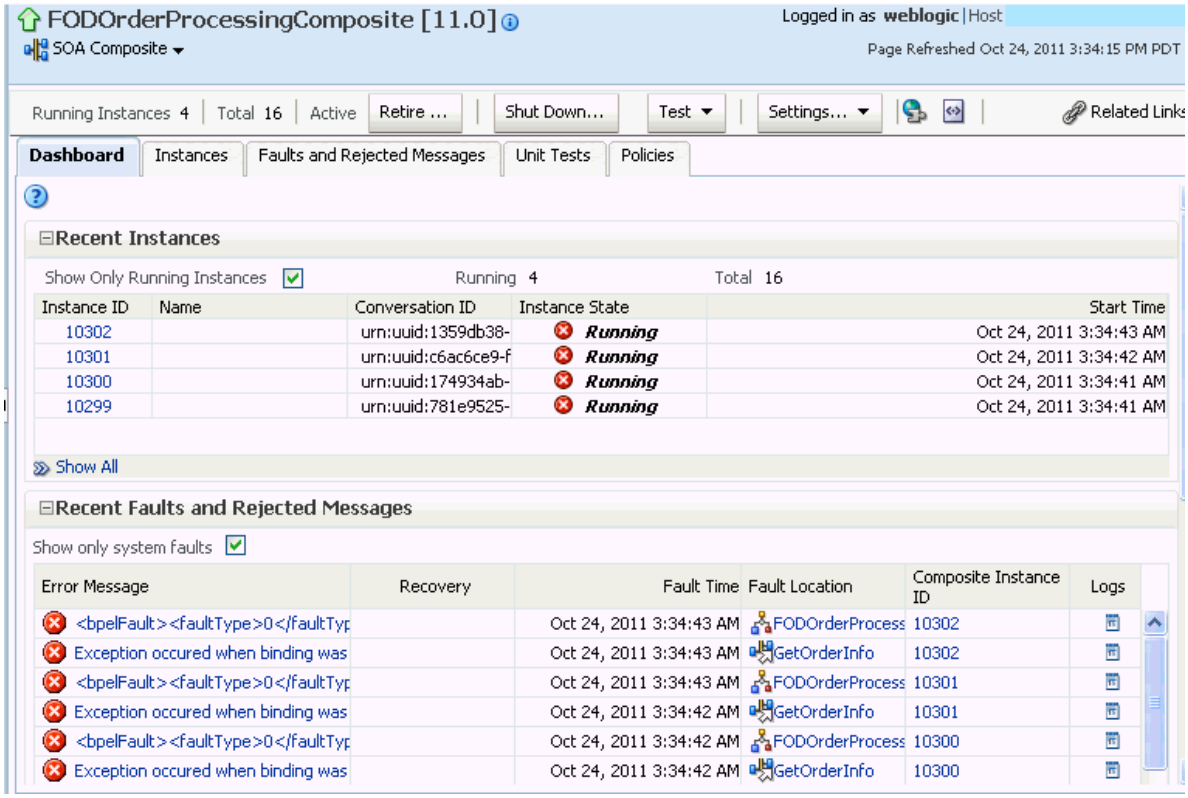
1. Expand **SOA** > **soa-infra** in the navigator.
2. Expand the partitions.

   This displays all SOA composite applications running in the partitions of the SOA Infrastructure.



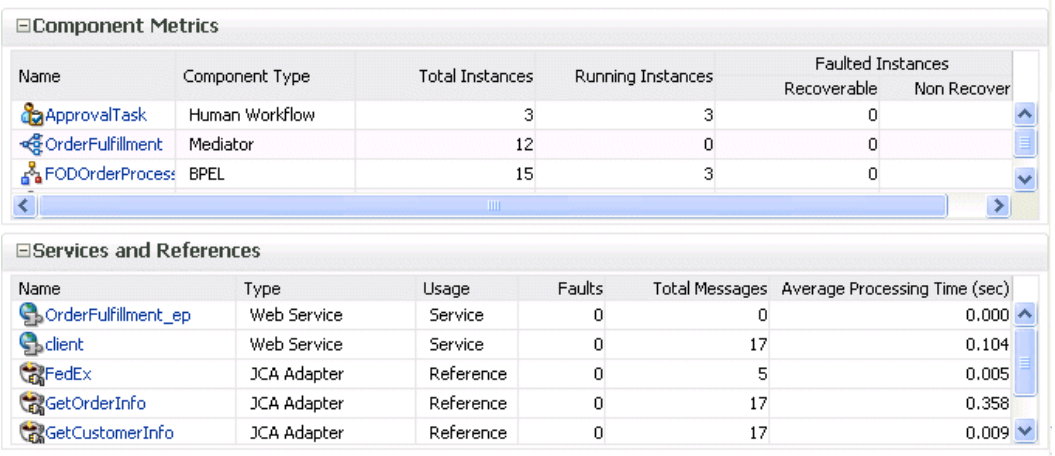
3. Select a specific SOA composite application.

   This displays the home page for the selected SOA composite application. The upper part of the page displays details about recent instances and faults and rejected messages. This part of the page also provides you with access to important administrative and configuration tasks at the composite level.



4. Click **Show All** at the bottom of each section to see more of these items. You can also click links to access more details about a specific composite instance or fault.

   The lower part of the page displays details about the service components (in this example, Oracle Mediator, BPEL process, and human workflow) and binding components (services and references) included in the SOA composite application. You can click a specific service component or service or reference binding component in the **Name** column to access more specific details.



   When you are within the pages of a SOA composite application, the **SOA Composite** menu appears below the application name at the top of the page. This menu provides you with administrative tasks specific to the current composite application.

5. Select the **SOA Composite** menu.

These administrative options enable you to perform the following tasks.

Option	Description
Home	This option displays the following details for the SOA composite application home page. ▪ Recent instances. ▪ Recent faults and rejected messages, including whether the instance is recoverable. ▪ Service components included in the SOA composite application. ▪ Services and references included in the SOA composite application. For more information, see Chapter 7, "Deploying and Managing SOA Composite Applications."
Monitoring	This option displays the performance summary statistics for the selected SOA composite application.
SOA Deployment	This option enables you to undeploy or redeploy this SOA composite application, or deploy another SOA composite application. For more information, see Section 7.1, "Deploying SOA Composite Applications."
Export	This option enables you to export a deployed SOA composite application to a JAR file. For more information, see Section 7.7, "Exporting a Deployed SOA Composite Application."
Test Service	This option enables you to manually initiate an instance of this deployed SOA composite application through the Test Web Service page. For more information, see Section 8.1, "Initiating a SOA Composite Application Test Instance."
Unit Tests	This option enables you to run test cases that simulate the interaction between the current SOA composite application and its web service partners before deployment to a production environment. This generates test instances of the composite. For more information, see Section 1.4.3.4, "Introduction to SOA Composite Application Automated Testing" and Section 7.5, "Automating the Testing of SOA Composite Applications."
Policies	This option enables you to view and attach or detach policies to or from the SOA composite application. For more information, see Section 1.4.3.2, "Introduction to Policies" and Section 7.6, "Managing SOA Composite Application Policies."
SOA Infrastructure	This option takes you to the SOA Infrastructure home page.

Option	Description
SOA Infrastructure Common Properties	This option enables you to view and set the audit level, capture the state of the SOA composite application instance, enable the payload validation of incoming messages, set UDDI registry properties, specify the callback server and server URLs, view the data source JNDI locations, set the nonfatal connection retry counts, set web service binding properties, and optimize the loading performance of pages. The SOA composite application typically inherits the settings defined at the SOA Infrastructure level. For more information, see Section 3.1, "Configuring SOA Infrastructure Properties."
Service/ Reference Properties	This option enables you to configure WSDL file properties for the service and reference binding components included in the SOA composite application. For more information, see Section 34.1, "Configuring Service and Reference Binding Component Properties."
General Information	This option displays general details about this SOA composite application, such as Oracle Enterprise Manager Fusion Middleware Control version, Oracle home, Oracle instance, and partition in which the composite is deployed.

6. Right-click the name of a SOA composite application in the navigator. A menu is displayed that provides you with access to the same administrative options that appear in the **SOA Composite** menu.

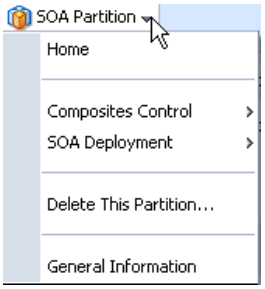
2.2.3 Navigating Through the Partition Home Page and Menu

You can navigate to administrative tasks for a specific partition in the SOA Infrastructure.

To navigate through the partition home page and menu:

1. Expand **SOA** > **soa-infra** in the navigator.
2. Click a specific partition.

3. At the top of the page, select the **SOA Partition** menu.



These administrative options enable you to perform the following tasks:

Option	Description
Home	This option displays the following details for the partition home page: ▪ SOA composite applications in this partition. ▪ Status of the SOA composite applications in this partition (either Up (on) or Down (off)). ▪ Mode of the SOA composite applications in this partition (active or retired). ▪ Total number of instances and faulted instances of the SOA composite applications in this partition. ▪ Time at which the SOA composite applications in this partition were last deployed, redeployed, or reconfigured.
Composites Control	This option displays the following bulk lifecycle management tasks: ▪ **Start Up All**: Invokes a dialog for starting all composites in this partition. ▪ **Shut Down All**: Invokes a dialog for shutting down all composites in this partition. ▪ **Activate All**: Invokes a dialog for activating all composites in this partition. ▪ **Retire All**: Invokes a dialog for retiring all composites in this partition. For information about these composite states, see Section 7.4.1, "Managing the State of All Applications at the SOA Infrastructure Level."
SOA Deployment	This option displays the following management tasks. ▪ **Deploy To This Partition**: Invokes the Deploy SOA Composite wizard for selecting a composite to deploy in this partition. ▪ **Undeploy All From This Partition**: Invokes a dialog for undeploying all composites in this partition. ▪ **Redeploy**: Invokes the Redeploy SOA Composite wizard to select composites to redeploy in this partition. For more information, see Chapter 7, "Deploying and Managing SOA Composite Applications."
Delete This Partition	This option invokes a dialog to delete this partition. All composites in the partition are automatically undeployed before the partition is deleted.
General Information	This option displays general details about the partition, such as the partition (target) name, Oracle Fusion Middleware version, Oracle Fusion Middleware home directory, domain home directory, hostname, and deployed server.

4. Right-click the name of a partition in the navigator. A menu is displayed that provides you with access to the same administrative options that appear in the **SOA Partition** menu.

> **Note:** You can also access the partition home page from the **Related Topics** menu on the home page of a SOA composite application.

For information about partitions, see Section 1.4.3.5, "Introduction to Partitioning of the SOA Infrastructure" and Section 7.8, "Grouping SOA Composite Applications into Partitions."

2.2.4 Navigating to Deployed Java EE Applications

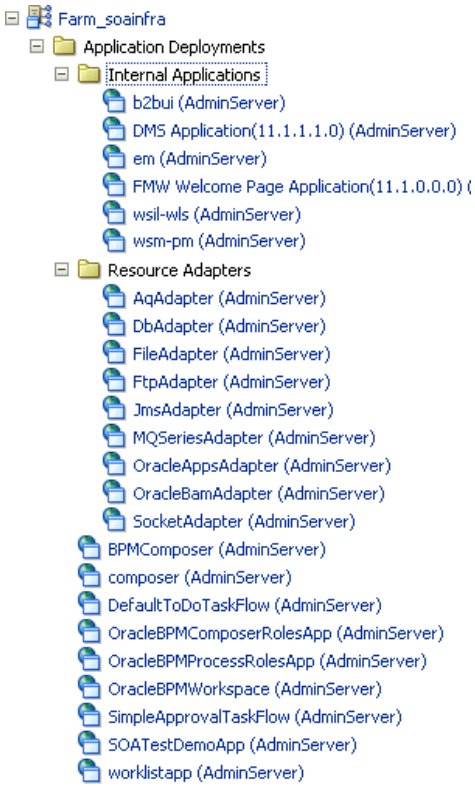
You can navigate to deployed Java EE applications related to Oracle SOA Suite and (if installed) Oracle BPM Suite components. These applications are Java EE applications that represent the SOA system components, such as the technology adapters, Oracle B2B, Oracle BPM Worklist, and so on. You can deploy a web service and see it listed here. You can also click individual applications (for example, the deployed web

service), and manage and test that you can deploy WAR and EAR files from here. If you have deployed your own Java EE applications, they also display here.

To navigate to deployed Java EE applications:

1. Expand **Application Deployments** in the navigator.
2. Expand **Internal Applications**.
3. Expand **Resource Adapters**.

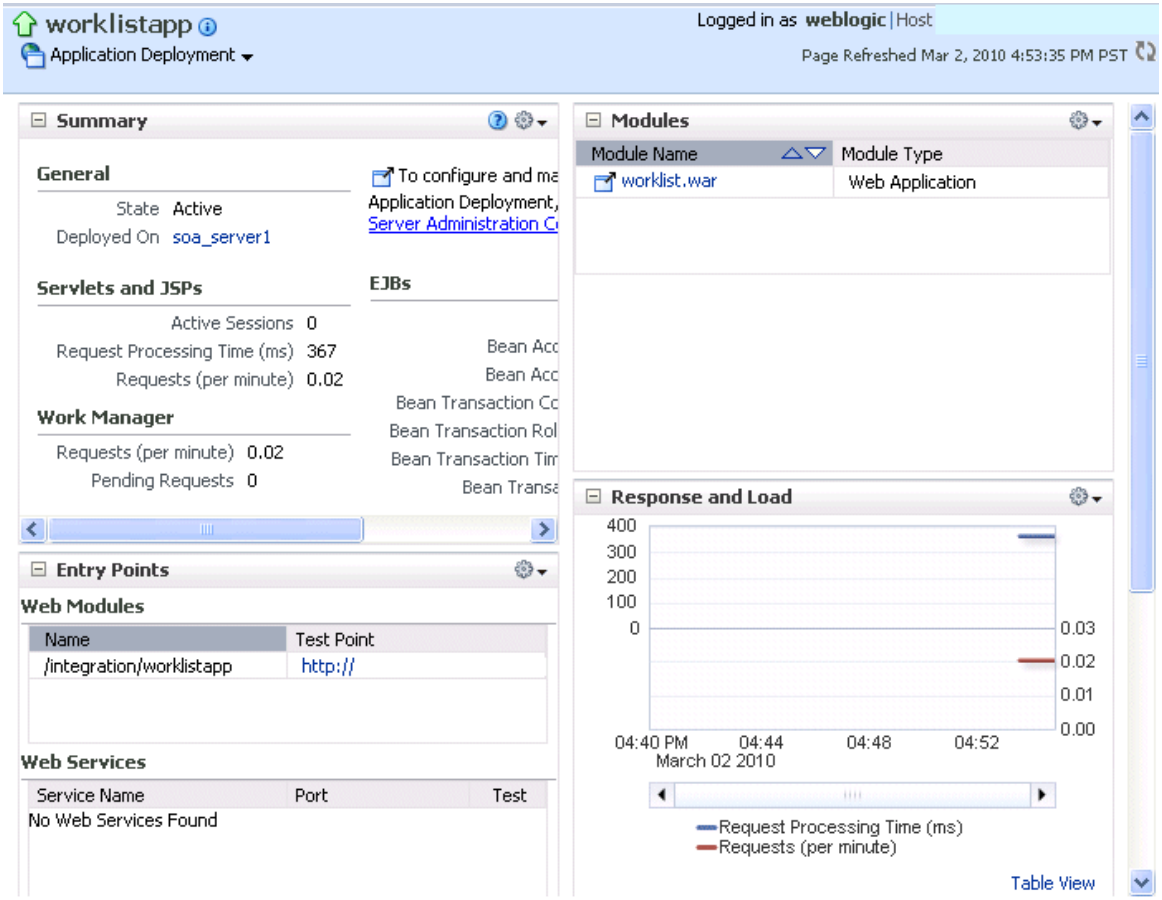
   A list of deployed Java EE applications related to Oracle SOA Suite and Oracle BPM Suite components appears.



   Oracle B2B (**b2bui**) appears in the **Internal Applications** folder, while Oracle BPM Worklist (**worklistapp**) appears in the top level **Application Deployments** folder.

4. Click a specific application (for this example, **worklistapp** is selected).

   The page displays details about application performance.

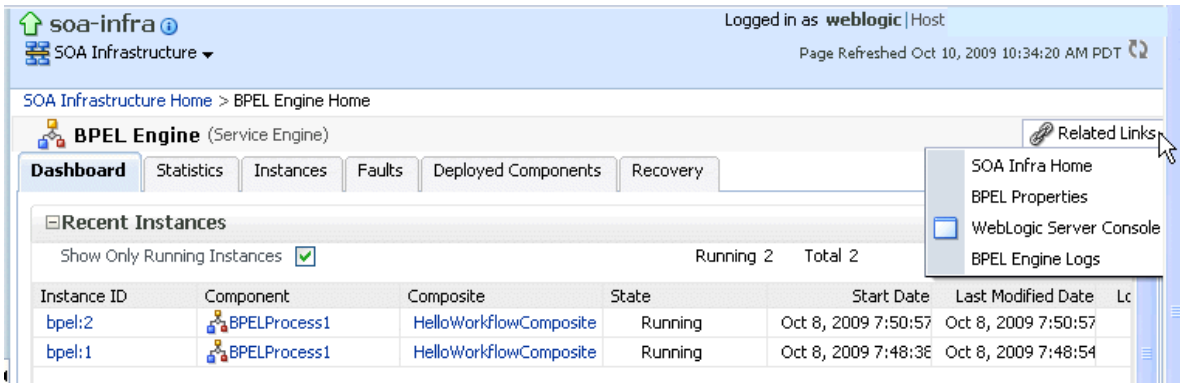


While Oracle Enterprise Manager Fusion Middleware Control displays the URLs for all deployed modules, you cannot directly invoke them from this page.

2.2.5 Navigating to the Oracle WebLogic Server Administration Console and Other Pages

Most pages in Oracle Enterprise Manager Fusion Middleware Control include a **Related Links** menu in the upper right-hand corner. Depending upon your current location, the context of the menu changes to provide links to relevant pages. For example, when you are on the BPEL process service engine page, the **Related Links** menu provides links to the SOA Infrastructure home page, the BPEL process configuration properties page, the Oracle WebLogic Server Administration Console, and a page for service engine log files. Figure 2–1 provides details. You can also click **soa-infra** at the top of the BPEL service engine page to go directly to the SOA Infrastructure home page.

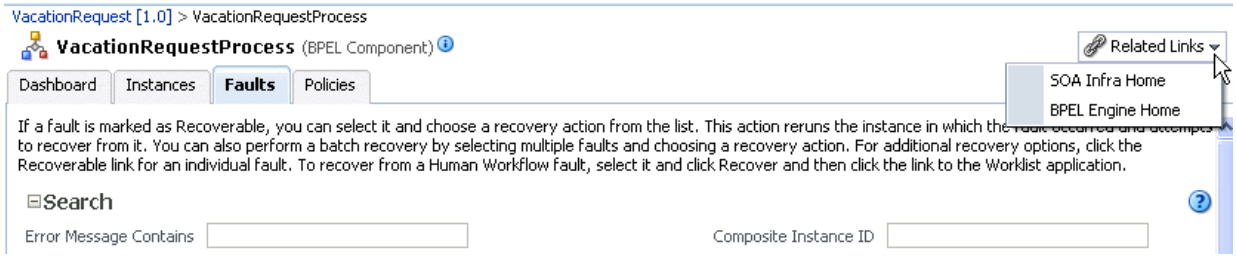
Figure 2–1 Related Links Menu for a SOA Composite Application



When you are on the page of a service component of a SOA composite application, several navigational menus and links are available:

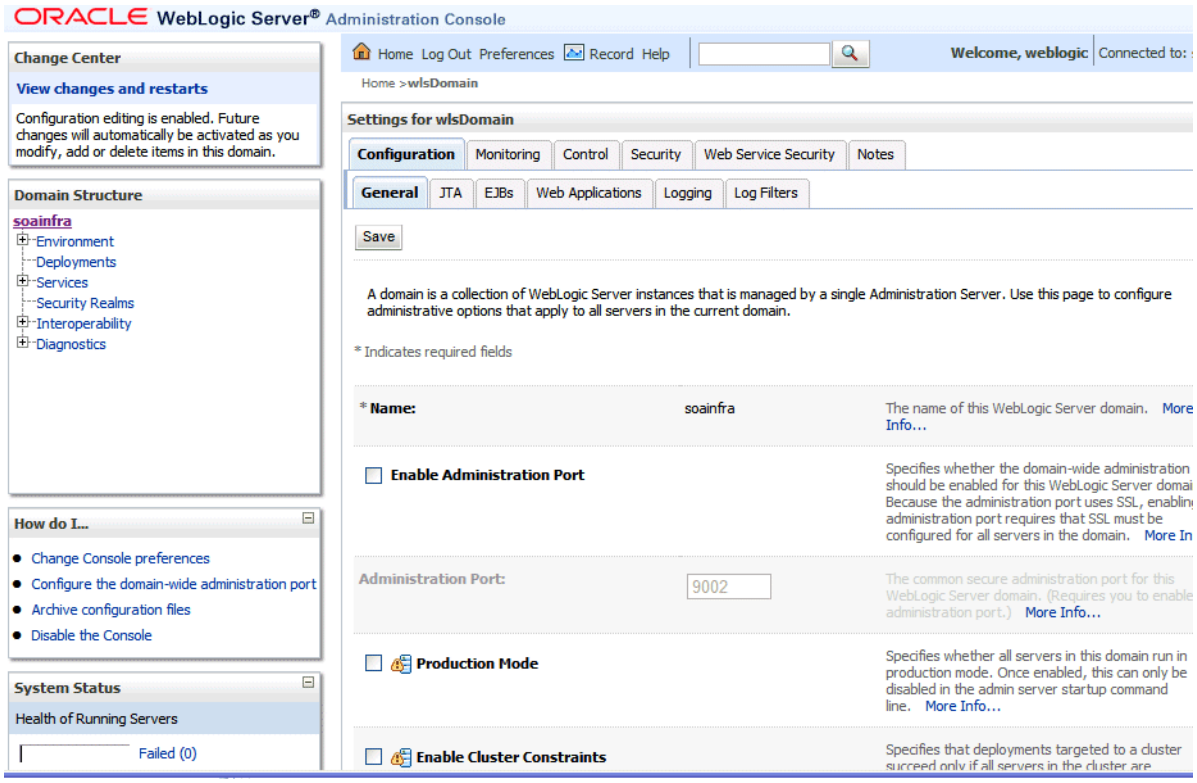
- The **Related Links** menu provides links to the SOA Infrastructure home page and the applicable service engine home page. Figure 2–2 provides details.
- Breadcrumbs are displayed in the upper left-hand corner as you traverse further into a SOA composite application.
- Within any SOA composite application page (including the service component pages), links to the SOA Infrastructure pages also remain available through the **SOA Infrastructure** menu that is displayed next to the **Farm** menu above the navigator. For example, this enables you to go from the home page of a specific BPEL service component directly to the BPEL service engine configuration properties page.
- Names at the top of the page can be clicked to navigate to parent pages. For example, clicking the name of a composite at the very top of a service component page enables you to go to the composite that includes that component.

Figure 2–2 Related Links Menu for a Service Component of a SOA Composite Application



The service engines, SOA administration (such as the SOA Infrastructure Common Properties page), and business event pages all provide access to Oracle WebLogic Server Administration Console from the **Related Topics** list. Selecting **WebLogic Server Console** opens a new browser page and takes you to the login prompt for the Oracle WebLogic Server Administration Console. Your current page in Oracle Enterprise Manager Fusion Middleware Control is not lost. After logging in, the home page is displayed. Figure 2–3 provides details.

Figure 2–3 Oracle WebLogic Server Administration Console



You can perform the following Oracle SOA Suite tasks from Oracle WebLogic Server Administration Console:

- Configure adapter connections for composite references.
- Create and manage data sources utilized by adapters.
- Create and manage JMS resources utilized by adapters.
- Manage SOA Infrastructure data sources (for example, modifying connection pool settings).
- Administer security of human workflow users.
- Manage Oracle WebLogic Server transaction (JTA) settings (for example, the transaction timeout value).
- Deploy human task user interfaces.

2.2.6 Navigating to the SOA Infrastructure or SOA Composite Application Home Page from the Farm Home Page

You can access the home page of the SOA Infrastructure, a specific SOA partition, or a specific SOA composite application from the Farm home page.

To navigate to the SOA Infrastructure or SOA composite application home page:

1. In the **Deployments** section of the Farm home page, click **soa-infra** or a specific SOA composite application (for example, **SimpleApproval**).

The home page for your selection is displayed.

2.3 Logging Out of Oracle Enterprise Manager Fusion Middleware Control

This section describes how to log out of Oracle Enterprise Manager Fusion Middleware Control.

To log out of Oracle Enterprise Manager Fusion Middleware Control:

1. Note the following details about logging out.
   - If multiple pages are open (for example, the help dialog, topology viewer, and flow trace), logging out of any page logs you out of the entire application in all open pages.
   - If you log out with any unsaved configuration changes, you receive no warning message and your changes are lost.
2. In the upper right-hand corner of any page, click the L**og Out** link.

Part III

Administering the SOA Infrastructure

This part describes how to administer the SOA Infrastructure.

This part includes the following chapters:

- Chapter 3, "Configuring the SOA Infrastructure"
- Chapter 4, "Monitoring the SOA Infrastructure"

3

Configuring the SOA Infrastructure

This chapter describes how to configure the properties of the SOA Infrastructure, including audit levels, composite instance states, and payload validation. These property settings can apply to all SOA composite applications running in the SOA Infrastructure. It also describes how to configure local optimization and stop and start the managed server and SOA Infrastructure.

This chapter includes the following topics:

- Section 3.1, "Configuring SOA Infrastructure Properties"
- Section 3.2, "Stopping and Starting the Managed Server and SOA Infrastructure"
- Section 3.3, "Changing the SOA Infrastructure Server URL Property Port"
- Section 3.4, "Configuring Log Files"
- Section 3.5, "Changing the Driver Name to Support Custom XA Drivers"
- Section 3.6, "Specifying a Nondefault XA Transaction Timeout Value for XA Data Sources"
- Section 3.7, "Configuring Local Optimization"

For more information, see Section 1.2.1, "Introduction to the SOA Infrastructure Application."

3.1 Configuring SOA Infrastructure Properties

You can configure the following properties for the SOA Infrastructure:

- Audit level
- Composite instance state to capture
- Payload validation
- Universal Description, Discovery, and Integration (UDDI) registry
- Callback server and server URLs
- Display of recent instances, faults, and count metrics on pages
- Search criteria for the retrieval of recent instances and faults
- Java Naming and Directory Interface (JNDI) data source
- Web service binding properties

The properties set at this level impact all deployed SOA composite applications, except those composites for which you explicitly set different audit level values at the composite application or service engine levels.

Additional advanced properties for the SOA Infrastructure can be configured through the System MBean Browser. You can access these properties from the **More SOA Infra Advanced Configuration Properties** link on the Common Properties page as described in this section or from the **SOA Infrastructure** menu by selecting **Administration** > **System MBean Browser** > **Application Defined MBeans** > **oracle.as.soainfra.config**.

SOA Infrastructure properties are stored in the Oracle Metadata Services (MDS) Repository associated with the SOA Infrastructure. For Oracle SOA Suite, the MDS Repository is configured by default to store its contents in the database.

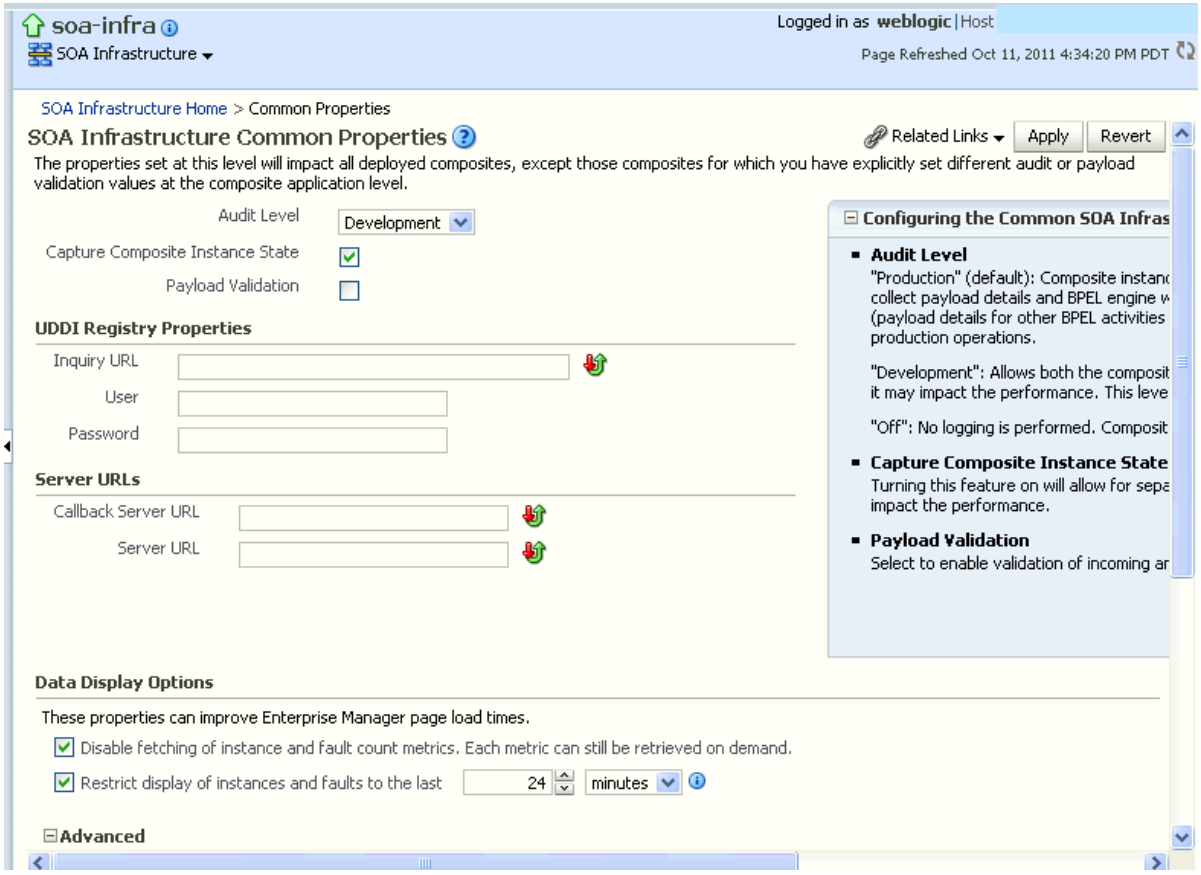
To configure SOA Infrastructure properties:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...	From the SOA Composite Menu...
1. Select **SOA Administration** > **Common Properties**.	1. Right-click **soa-infra**. 2. Select **SOA Administration** > **Common Properties**.	1. Select **SOA Infrastructure Common Properties**.

The SOA Infrastructure Common Properties page displays the following properties.

Note: Some property fields are designated with an icon showing green and red arrows. If you change these properties, you must restart the SOA Infrastructure.



Descriptions for the properties at the top of the page are provided in the following table.

Element	Description
Audit Level	Select the level of information to be collected by the message tracking infrastructure. This information is collected in the instance data store (database) associated with the SOA Infrastructure. This setting has no impact on what gets written to log files. ▪ **Off**: No composite instance tracking and payload tracking information is collected. No more composite instances can be created. No logging is performed. Disabling logging and the display of instances in Oracle Enterprise Manager Fusion Middleware Control can result in a slight performance increase for processing instances. Instances are created, but are not displayed. ▪ **Development**: Enables both composite instance tracking and payload detail tracking. However, this setting may impact performance. This level is useful largely for testing and debugging purposes. ▪ **Production**: Composite instance tracking is collected, but the Oracle Mediator service engine does not collect payload details and the BPEL process service engine does not collect payload details for assign activities (payload details for other BPEL activities are collected). This level is optimal for most normal production operations.

Element	Description
Capture Composite Instance State	Select to capture the SOA composite application instance state. Enabling this option may result in additional runtime overhead during instance processing. This option provides for separate tracking of the running instances. All instances are captured as either running or not running. This information appears later in the **State** column of the composite instances tables for the SOA Infrastructure and SOA composite application, where: ■ It shows the counts of running instances versus total instances ■ You can also limit the view to running instances only Valid states are running, completed, faulted, recovery needed, stale, terminated, and state not available. The running and completed states are captured only if this checkbox is selected. Otherwise, the state is set to unknown. The conditional capturing of these states is done mainly to reduce the performance overhead on SOA Infrastructure runtime. **Note:** If this property is disabled and you create a new instance of a SOA composite application, a new instance is created, but the instance does not display as running, faulted, stale, terminated, completed, or requiring recovery in the Dashboard page table of the composite application. This is because capturing the composite state of instances is a performance-intensive process. For example, if you enable this property and create a SOA composite application instance in the Test Web Service page, a new instance appears in the Dashboard page of the composite application. If you click **Show Only Running Instances** in the Dashboard page, the instance is displayed as running. If you then disable this property and create another instance of the same composite application, a new, running instance is created. However, if you then select **Show Only Running Instances**, the new instance is *not* listed in the running instances table. In addition, to terminate a running instance, the instance must have a state (for example, running, faulted, and so on). This activates the **Abort** button on the Instances page of a SOA composite application. If this checkbox is not enabled before creating an instance, the **Abort** button is inactive, and you cannot terminate the instance.
Payload Validation	Select to enable validation of incoming and outgoing messages. Nonschema-compliant payload data is intercepted and displayed as a fault.

2. Make changes appropriate to your environment.

   The **UDDI Registry Properties** section displays the following properties. You can integrate SOA composite applications running in the SOA Infrastructure with the UDDI registry. The UDDI registry provides a standards-based foundation for locating published services and managing metadata about services (security, transport, or quality of service). You can browse and select published services that meet your needs.

   The **User** and **Password** properties are applicable if the UDDI registry is secured. These are only used for the secure HTTP configuration of Oracle Service Registry (OSR). The **Inquiry URL** property is public.

Element	Description	Example
Inquiry URL	Enter the URL of the master registry you want to query. The URL must not refer to the slave registry itself. Otherwise, you can lose some data. The inquiry URL obtains full-standard UDDI version 3 structures. This is the same UDDI inquiry URL that you specified in the Create UDDI Registry Connection wizard.	`http://master.mycompany.com:8888/registry/uddi/inquiry`
User	Enter the registry inquiry user.	`admin`
Password	Enter the password for the master registry inquiry user.	Enter a password that utilizes good security practices.

For information about setting the endpoint reference and service key, see Section 34.1.3, "Changing the Endpoint Reference and Service Key for Oracle Service Registry Integration."

3. Make changes appropriate to your environment.

   The **Server URLs** section displays the following properties. If not explicitly set here, these values are determined at runtime by querying the Oracle WebLogic Server cluster, the web server, or the local server properties.

Element	Description
Callback Server URL	Enter the callback server URL. This URL is sent by the server as part of the invocation to a foreign service provider.
Server URL	Enter the server URL. This URL is published as part of the SOAP address of a service in the concrete WSDL file. **Note:** In 10.1.*x* releases, you manually configured SOAP optimization with the `optSoapShortcut` property. For release 11*g*, SOAP optimization is automatically configured. Therefore, if you upgrade to 11*g* and are using the optimized shortcut approach in existing applications, optimized calls are activated only when the hostname value (as referred to in the WSDL URL in the `composite.xml` file) matches the **Server URL** value. Either set both values to the hostname (for example, **myhost**) or to the full domain name (for example, **myhost.domain.com**). If these values do not match, a regular SOAP call is performed instead of an optimized local call. For information about local optimization, see Section 3.7, "Configuring Local Optimization."

> **Note:** If you change the **Callback Server URL** and **Server URL** values (for example, when moving from a test to a production environment), you must restart Oracle WebLogic Server for the WSDLs to be regenerated.

4. Make changes appropriate to your environment.

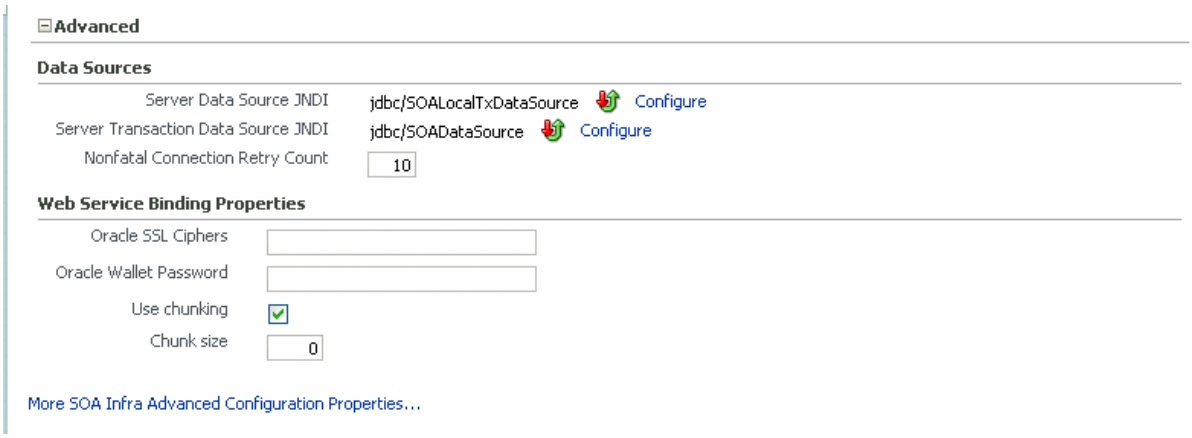
   The **Display Data Counts** section displays the following properties for improving the time it takes to load pages.

Note: Any changes to these properties impact all SOA farms associated with this Oracle Enterprise Manager instance.

Element	Description
Disable fetching of instance and fault count metrics. Each metric can still be retrieved on demand.	Select to disable the display of instance and fault count metrics on the Dashboard pages of the SOA Infrastructure, SOA composite applications, service engines, and service components. Instead, these metrics are replaced with links that you click to retrieve the instance and fault count metrics when you need this information. This setting can improve the time it takes to load pages. If you click a link to retrieve instance and fault count metrics, and Oracle Enterprise Manager Fusion Middleware Control times out, increase the transaction timeout property. For more information, see Section B.2.1, "Resolving Connection Timeouts" and Section B.6.1, "Optimizing the Loading of Pages with Instance and Fault Metrics."
Restrict display of instances and faults to the last *time_period*	Select this checkbox and specify a time period during which to retrieve recent instances, faults, and count metrics for display on the following pages: ▪ Dashboard pages of the SOA Infrastructure, SOA composite applications, service engines, and service components ▪ Deployed Composites page of the SOA Infrastructure and service engines ▪ Partition home page By default, this checkbox is selected and the time period duration is set to 24 hours (one day). If this checkbox is not selected, all instances and faults (including count metrics) in the SOA Infrastructure since the last purging are displayed. **Note**: It is highly recommended that you set a time period duration because it has a significant impact on the performance of multiple Oracle Enterprise Manager Fusion Middleware Control pages and queries. The time period you specify appears by default in the **Fault Time From** field on faults pages on which you can search for faults and the **Start Time From** field on instances pages on which you can search for instances. When you select this checkbox and specify a time period, the **Show All** links change to **Show More** links on the Dashboard pages of the SOA Infrastructure, SOA composite applications, service engines, and service components.

For additional details, see Section B.6.1, "Optimizing the Loading of Pages with Instance and Fault Metrics."

5. Make changes appropriate to your environment.
6. Expand the **Advanced** section.



The **Data Sources** section displays the following properties. A data source enables you to retrieve a connection to a database server.

Element	Description	Example
Server Data Source JNDI	Displays the JNDI location for the server data source. Click **Configure** to go to the data source configuration page of the Oracle WebLogic Server Administration Console. Global transaction support should be disabled for this data source.	`jdbc/SOALocalTxDataSource`
Server Transaction Data Source JNDI	Displays the JNDI location for the server transactional data source. Click **Configure** to go to the data source configuration page of the Oracle WebLogic Server Administration Console. You must configure the data source for global transactions.	`jdbc/SOADataSource`
Nonfatal Connection Retry Count	Enter the maximum number of times a nonfatal connection error can be retried before failing. These type of errors occur for any connection error with the dehydration store (for example, Oracle Real Application Clusters failover, database shutdown, and so on).	`10`

7. Make changes appropriate to your environment.

   The **Web Service Binding Properties** section displays the following options.

Element	Description	Example
Oracle SSL Ciphers	Enter the list of supported Oracle ciphers. A cipher suite is a set of algorithms that provide security for data transmissions. Before data can flow through an SSL connection, both sides of the connection must negotiate common algorithms to use.	`SSL_RSA_WITH_RC4_128_MD5`
Oracle Wallet Password	Enter the wallet password for the keystore.	Enter a password that utilizes good security practices.
Use Chunking	Select to enable chunking of data for SOAP over HTTP deliveries.	`--`

Element	Description	Example
Chunk Size	Specify a chunk size. The value must be less than or equal to `999`. The size is used for SOAP over HTTP deliveries and is specified in bytes.	`500`

8. Make changes appropriate to your environment.
9. Click **Apply**.
10. If you make changes and want to reset these properties to their previous values, click **Revert**.
11. To change advanced parameters, click **More SOA Infra Advanced Configuration Properties**. This opens the System MBean Browser. The properties that display include, but are not limited to, the following. Descriptions are provided for each property.
    - **AuditConfig**: The status of BPEL message recovery. By default, the **bpelRecoveryStatus** key of this property is set to **All**. If there are BPEL messages requiring recovery in the Recovery page of the BPEL process service engine, this setting enables a `BPEL Message Recovery Required` inline warning message and recovery icon to display in the **Trace** table of the Flow Trace page and the home pages of the SOA Infrastructure and SOA composite applications. If this key is set to **Off**, no message recovery information is displayed. For more information, see the following sections:
      - Section 4.1, "Monitoring SOA Infrastructure Recent Instances and Faults and Deployed Composites"
      - Section 6.1, "Monitoring SOA Composite Application Recent Instances and Faults and Rejected Messages"
      - Section 13.1, "Monitoring the Audit Trail and Process Flow of a BPEL Process Service Component"

      This property also includes the **excludeBpelMaxCreationTime** key. This key enables you to set the time period for excluding messages that require recovery. For more information, see Section 4.1, "Monitoring SOA Infrastructure Recent Instances and Faults and Deployed Composites."
    - **GlobalTxMaxRetry**: The maximum number of times an invocation exception can be retried.
    - **GlobalTxRetryInterval**: The number of seconds between retries for an invocation exception.
    - **HttpProxyAuthRealm**: The HTTP proxy authentication realm.
    - **HttpProxyAuthType**: The HTTP proxy authentication type.
    - **HttpProxyHost**: The HTTP proxy host.
    - **HttpProxyPassword**: The password for HTTP proxies that require authentication.
    - **HttpProxyPort**: The HTTP proxy port number.
    - **HttpProxyUsername**: The user name for HTTP proxies that require authentication.
    - **HttpServerURL**: The HTTP protocol URL published as part of the SOAP address of a process in the WSDL file.

- **HttpsServerURL**: The HTTPS protocol URL published as part of the SOAP address of a process in the WSDL file.
- **KeystoreLocation**: The path to the Oracle SOA Suite keystore.
- **UddiCacheLifetime:** The UDDI endpoint cache life span.

3.1.1 Disabling Instance and Fault Count Metrics Retrieval with the System MBean Browser

As described in Section 3.1, "Configuring SOA Infrastructure Properties," you can disable the retrieval of instance and fault count metrics in the **Display Data Counts** section of the SOA Infrastructure Common Properties page. You can also change this property through the System MBean Browser.

To disable instance and fault count metrics retrieval with the System MBean Browser.

1. Select **Application Defined MBeans** > **emom.props** > **Server:AdminServer** > **Application: em** > **Properties** > **emoms.properties**.

   Note that **emoms.properties** is *only* available for selection if you previously modified the **Disable fetching of instance and fault count metrics** option of the **Display Data Counts** section of the SOA Infrastructure Common Properties page.
2. In the **Name** column of the **Attributes** tab, click **Properties**.
3. In the **Value** column, expand **Element_20**.
4. In the **Element** column, enter `false` to disable metrics retrieval.
5. Click **Apply**.
6. Restart the SOA Infrastructure. A restart is not required if you instead change the **Disable fetching of instance and fault count metrics** option through the **Display Data Counts** section of the SOA Infrastructure Common Properties page.

3.2 Stopping and Starting the Managed Server and SOA Infrastructure

You can stop and start the SOA Infrastructure in Oracle Enterprise Manager Fusion Middleware Control for maintenance or for configuration restarts. To do so, stop and start the managed server on which the SOA Infrastructure is installed. This restarts both the managed server and the SOA Infrastructure.

Notes:

- Starting with 11*g* Release 1 (11.1.1.4.0), you can no longer stop and start the SOA Infrastructure from the **soa-infra** menu in the navigator.
- You can also have a developer configuration that only includes an administration server, and no managed servers.

To stop and start the managed server and SOA Infrastructure:

1. Access this page through one of the following options:

From the WebLogic Server Menu...	From the WebLogic Domain Folder in the Navigator...
1. Select **Control**.	1. Right-click the managed server (for example, **soa_server1**). 2. Select **Control**.

2. To shut down the managed server and SOA Infrastructure, select **Shut Down**.
3. Click **OK** when prompted to shut down the managed server and SOA Infrastructure.
4. Wait for shutdown to complete.
5. To start the managed server and SOA Infrastructure, select **Start Up**.

For information on stopping and starting managed servers with Node Manager, see *Oracle Fusion Middleware Node Manager Administrator's Guide for Oracle WebLogic Server*.

For information on starting and stopping managed servers with WLST commands, see *Oracle Fusion Middleware Administrator's Guide*.

3.2.1 Waiting for SOA Infrastructure Startup Initialization to Complete

After the SOA Infrastructure is started, it may not be completely initialized to administer incoming requests until all deployed composites are loaded. Therefore, the response metrics that are displayed on some Oracle Enterprise Manager Fusion Middleware Control pages may not reflect their actual status. This is most apparent when the SOA Infrastructure is in a cluster with multiple managed servers and a large number of deployed composites.

During the initialization stage, Oracle Enterprise Manager Fusion Middleware Control does *not* prevent you from executing operations such as composite deployment, composite undeployment, and others, even though these operations may not complete successfully. Instead, a warning message is displayed at the top of the Oracle Enterprise Manager Fusion Middleware Control pages shown in Table 3–1. Do not perform operations such as composite deployment, composite undeployment, and others while this message is displayed. Once initialization completes, the message is no longer displayed. You see this after you refresh the page. You can then perform operations.

Table 3–1 SOA Infrastructure Initialization Message

This Warning Message Is Displayed...	At the Top of These Pages...
`Initializing SOA` `Even though the soa-infra target is up, some SOA Fabric components and composite applications are still loading. You may need to allow some time for the initialization to complete, and later click the Refresh Page icon.` **`It is not adivsable to execute any operations on this soa-infra until this warning goes away.`**	▪ SOA Infrastructure home page ▪ SOA composite application home page ▪ Manage Partitions page ▪ Partition home page

3.2.2 SOA Composite Application States and SOA Infrastructure Shutdown

SOA composite application states are not updated to indicate that they are down after SOA Infrastructure shutdown. If you attempt to access the composite, you receive an error message stating that composite details cannot be retrieved:

```
soa-infra runtime connection error  An error happened while connecting to
soa-infra runtime at  t3://152.61.150.106:8001/soa-infra.
```

This message may lead you to believe that another issue exists in the system. However, this is not the case.

These composite states display as up, or in some cases pending, because this metric indicates whether the composite is enabled, and is independent of whether the SOA Infrastructure is started. In addition, the composite is still active and can receive requests on other managed servers in a cluster.

3.2.3 Restarting the SOA Infrastructure Does Not Activate Endpoints When a Retired Composite is Activated

If a SOA composite application with adapter endpoints is in a retired state, the endpoints are not activated if you perform the following actions:

- Restart the SOA Infrastructure
- Activate the SOA composite application

This is because files, records, and so on are not picked up by the endpoint adapters. As a workaround, redeploy the SOA composite application after restarting the SOA Infrastructure.

3.2.4 SOA Infrastructure Startup Failure When cwallet.sso Includes the SOA Map

When `cwallet.sso` has the SOA map, you receive an error message similar to the following when attempting to start the SOA Infrastructure.

```
Caused By: java.security.UnrecoverableKeyException: Password verification
failed
        at
sun.security.provider.JavaKeyStore.engineLoad(JavaKeyStore.java:769)
        at
sun.security.provider.JavaKeyStore$JKS.engineLoad(JavaKeyStore.java:38)
        at java.security.KeyStore.load(KeyStore.java:1185)
        at oracle.j2ee.ws.saaj.util.SSLUtil.loadKeyStore(SSLUtil.java:73)
        at
oracle.j2ee.ws.saaj.util.SSLUtil.getKeyManagerFactory(SSLUtil.java:88)
        at oracle.j2ee.ws.saaj.util.SSLUtil.getKeyManagers(SSLUtil.java:97)
        at
oracle.j2ee.ws.saaj.util.SSLUtil.createSSLSocketFactory(SSLUtil.java:50)
        at
oracle.integration.platform.common.SSLSocketFactoryManagerImpl.getSSLSocketFac
tory(SSLSocketFactoryManagerImpl.java:58)
        at oracle.fabric.common.wsdl.WSDLManager.init(WSDLManager.java:356)
        at oracle.fabric.common.wsdl.WSDLManager.<init>(WSDLManager.java:101)
        at
oracle.fabric.common.metadata.MetadataManagerImpl.getWSDLManager(MetadataManag
erImpl.java:283)
        at
oracle.fabric.composite.model.CompositeModel.getWSDLManager(CompositeM
```

Perform the following steps to resolve this issue.

1. Perform one of the following actions:
   - Delete the SOA map in `cwallet.sso`.

- Remove `$DOMAIN_HOME/config/fmwconfig/default-keystore.jks`. Oracle Web Services Manager (OWSM) uses this file.

2. Restart the SOA Infrastructure.

3.3 Changing the SOA Infrastructure Server URL Property Port

In addition to the SOA Infrastructure Common Properties page, you can also change the SOA Infrastructure **ServerURL** property port in the System MBean Browser of Oracle Enterprise Manager Fusion Middleware Control.

When changing the port, note the following details:

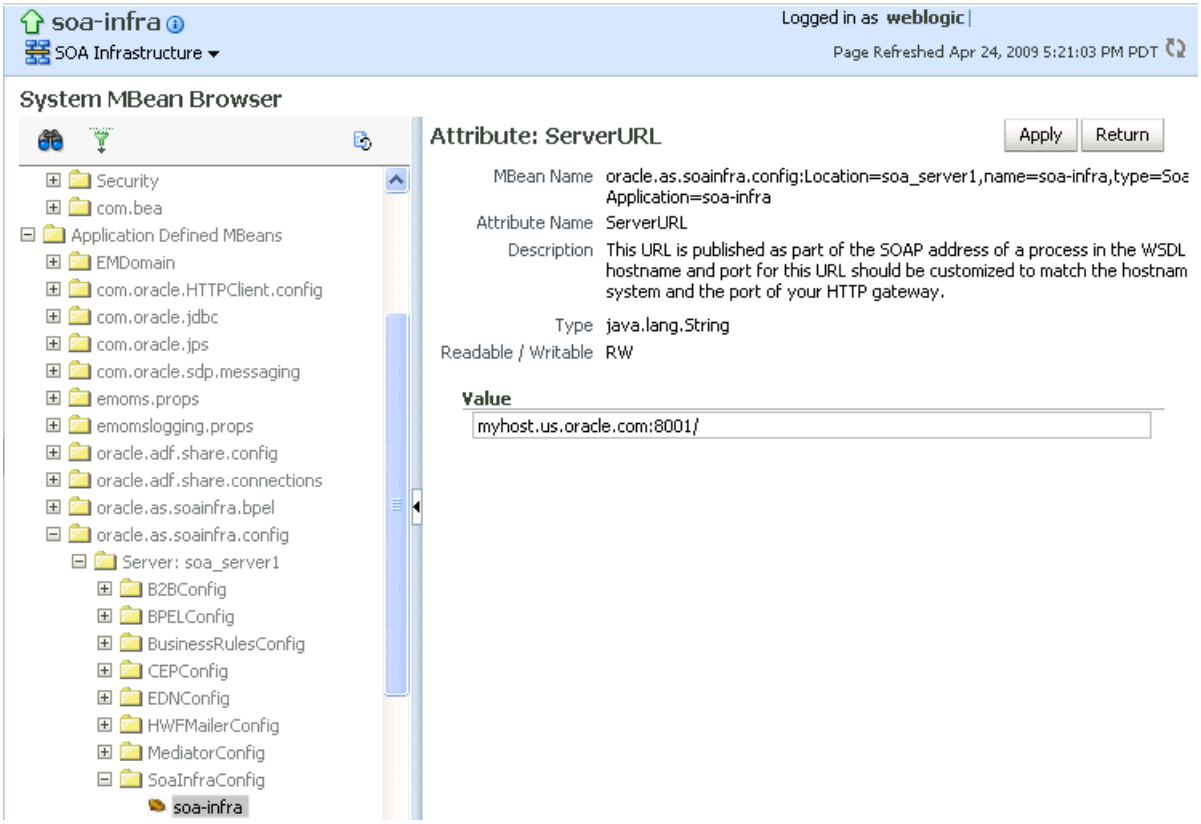
- If the SOA Infrastructure and managed Oracle WebLogic Server port numbers are different, you receive a `ConnectException` error when trying to connect to the Oracle BPM Worklist. Ensure that these port numbers match.
- You *cannot* change the SOA Infrastructure port from the Oracle WebLogic Server Administration Console. Only the port for the managed Oracle WebLogic Server can be changed from the Oracle WebLogic Server Administration Console.

To change the SOA Infrastructure port:

1. From the **SOA Infrastructure** menu, select **Administration** > **System MBean Browser**.
2. Under **Application Defined MBeans**, expand **oracle.as.soainfra.config** > **Server:** ***server_soa*** > **SoaInfraConfig** > **soa-infra**.

   where ***server_soa*** is the name of the server provided during post installation configuration. By default, this name is **soa_server1**.
3. In the **Name** column, click **ServerURL**.

   The Attribute: ServerURL page appears.



4. In the **Value** field, change the port.
5. Click **Apply**.
6. Change the managed Oracle WebLogic Server port in the Oracle WebLogic Server Administration Console to the same value.

   In environments in which a load balancer is used in front of an Oracle WebLogic Server cluster, the **ServerURL** property host and port can be different from the Oracle WebLogic Server host and port. This is typical for enterprise deployment environments in which a load balancer distributes requests across the managed servers in the Oracle WebLogic Server cluster. For more details, see *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite*.

3.4 Configuring Log Files

Oracle SOA Suite 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.

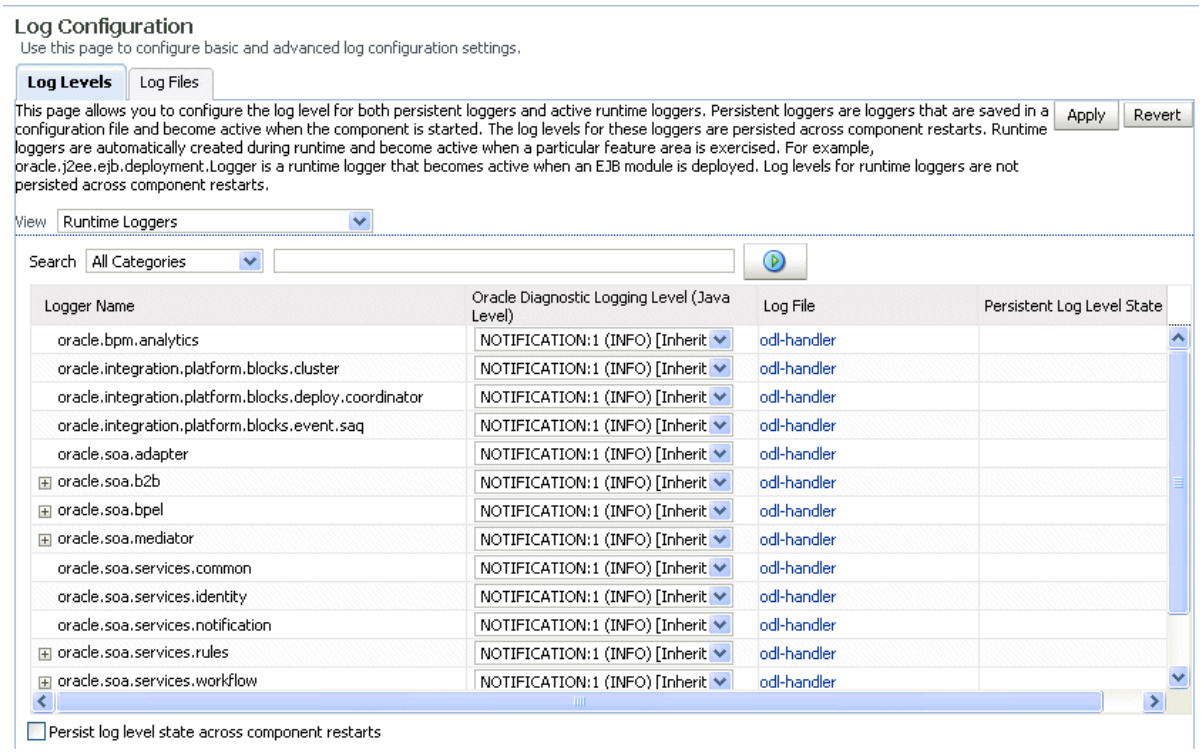
To configure log files:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Logs** > **Log Configuration**.	1. Right-click **soa-infra**. 2. Select **Logs** > **Log Configuration**.

The Log Configuration page displays the following details:

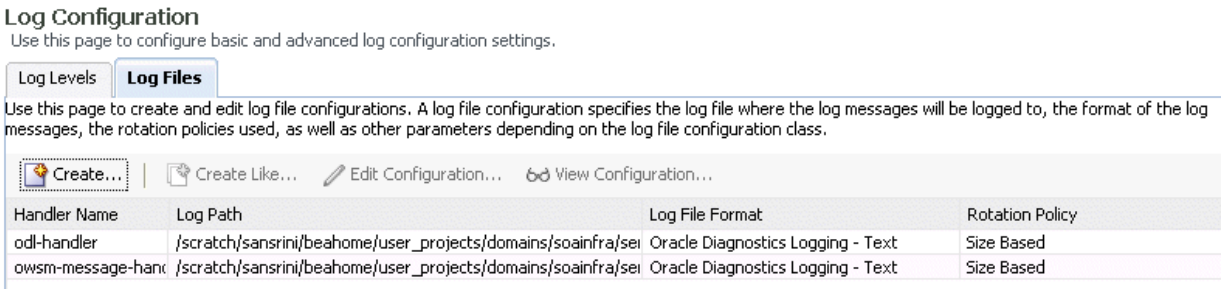
- A **View** list for selecting the type of loggers for which to view information:
  - Persistent: Loggers that become active when a component is started. Their configuration details are saved in a file and their log levels are persisted across component restarts.
  - Active runtime: Loggers that are automatically created during runtime and become active when a particular feature area is exercised (for example, **oracle.soa.b2b** or **oracle.soa.bpel**). Their log levels are not persisted across component restarts.
- A table that displays the logger name, Oracle Diagnostic Logging (ODL) level for setting the amount and type of information to write to a log file, the log file, and the log level state.



2. Perform the following log file tasks on this page:
   a. In the **Logger Name** column, expand a logger name. This action enables you to specify more specific logging levels within a component.
   b. In the **Oracle Diagnostic Logging Level** columns, select the level and type of information to write to a log file.
   c. In the **Log File** column, click a specific log file to create and edit log file configurations.

      For more information about ODL log files and the level and type of logging information to write to a log file, see *Oracle Fusion Middleware Administrator's Guide*.
3. Click the **Log Files** tab.

   This page enables you to create and edit log file configurations, including the log file in which the log messages are logged, the format of the log messages, the rotation policies used, and other parameters based on the log file configuration class.



For more information about logging, see *Oracle Fusion Middleware Administrator's Guide*.

For information on setting logging levels and Oracle SOA Suite logging files to view, see Section B.1, "Setting Logging Levels for Troubleshooting."

3.4.1 Configuring the Logging File Encoding Property

The `oracle-soa-handler` log handler property of the `soa-diagnostic.log` file has no encoding property specified in the *`SOA_Domain`*`/config/fmwconfig/servers/server_soa/logging.xml` file. Instead, the `soa-diagnostic.log` file is written in the operating system's default encoding format. This can cause the following problems:

- Non-ASCII error messages can become unreadable because logging information is written to `soa-diagnostic.log` in the server's default encoding format.
- On Windows operating systems, writing in the default encoding format can lead to non-ASCII data loss.

To avoid this problem, specify a value of `UTF-8` for the `oracle-soa-handler` log handler property in the `logging.xml` file.

```
<?xml version='1.0'?>
<logging_configuration>
 <log_handlers>
  <log_handler name='wls-domain'
 class='oracle.core.ojdl.weblogic.DomainLogHandler' level='WARNING'/>
  <log_handler name='oracle-soa-handler'
 class='oracle.core.ojdl.logging.ODLHandlerFactory'>
   <property name='path' value='c:\soa1210.1411\user_
projects\domains\soa/servers/server_soa/logs/soa-diagnostic.log'/>
   <property name='maxFileSize' value='10485760'/>
   <property name='maxLogSize' value='104857600'/>
   <property name='supplementalAttributes' value='J2EE_APP.name,J2EE_
MODULE.name,WEBSERVICE.name,WEBSERVICE_PORT.name,composite_instance_id,component_
instance_id,composite_name,component_name'/>
    <property name='encoding' value='UTF-8'/>
  </log_handler>
 </log_handlers>
...
```

Log files are written with ODL. You can view the content of log files from Oracle Enterprise Manager Fusion Middleware Control.

For more information about logging, see *Oracle Fusion Middleware Administrator's Guide*.

3.5 Changing the Driver Name to Support Custom XA Drivers

The default SOA Infrastructure data source is always XA-enabled. If your data sources require support for custom drivers, you must change the driver name on Oracle WebLogic Server.

To change the driver name through one of the following methods:

- Edit in Oracle WebLogic Server Administration Console.
  1. Log in to Oracle WebLogic Server Administration Console.
  2. In the left pane, select **Domain Structure**.
  3. Select **Services** > **JDBC** > **Data Source** > **SOADataSource** > **Connection Pool.**
  4. For the **Driver Class Name**, change the value to a custom data source (for example, `oracle.jdbc.xa.client.myDataSource`.
  5. Restart the server.
- Edit the `soaDataSource-jdbc.xml` file.
  1. Open the `soaDataSource-jdbc.xml` file on Oracle WebLogic Server.
  2. Change the `SOADataSource` driver name from `oracle.jdbc.OracleDriver` to `oracle.jdbc.xa.client.myDataSource`.

```
<?xml version="1.0" encoding="UTF-8"?>
<jdbc-data-source
/. . .
. . .
/  <name>SOADataSource</name>
 <jdbc-driver-params>
      <url>jdbc:oracle:thin:@adc60086fems.us.oracle.com:1537:co0yd570</url>
      <driver-name>*oracle.jdbc.xa.client.myDataSource*</driver-name>
   <properties>
     <property>
       <name>user</name>
       <value>fusion_soainfra</value>
     </property>
   </properties>
/  . . .
 . . ./
 </jdbc-driver-params>
/. . .
. . ./
</jdbc-data-source>
```

3.6 Specifying a Nondefault XA Transaction Timeout Value for XA Data Sources

The default XA transaction timeout value for XA data sources is `0` seconds. You can change the default value in the Oracle WebLogic Server Administration Console. Follow these steps.

To specify a nondefault XA transaction timeout value for XA data sources:

1. Log in to Oracle WebLogic Server Administration Console.
2. Under **Domain Structure** on the left side of the page, select **Services** > **JDBC** > **Data Sources**.

3. In the **Name** column of the **Data Sources** table, select EDNDataSource (for event delivery network transactions) or **SOADataSource** (for all other types of transactions).
4. Under the **Configuration** tab at the top, click the **Transaction** subtab.
5. In the **XA Transaction Timeout** field, enter a value in seconds.
6. Select the **Set XA Transaction Timeout** checkbox. You *must* select this checkbox for the new XA transaction timeout value to take effect.
7. Click **Save**.

3.7 Configuring Local Optimization

Local optimization is the process of one SOA composite application invoking another SOA composite application through direct Java invocations in an environment in which both composites are on the same SOA server (JVM).

Direct Java invocations are generally more efficient than SOAP over HTTP calls. Therefore, whenever the conditions are met for direct Java invocations, Oracle SOA Suite optimizes the service calls for the co-located composites.

3.7.1 Condition Checks for Using Local Optimization

Oracle SOA Suite performs the following condition checks to determine if local optimization is possible.

- It must be a composite-to-composite invocation. This is the most fundamental criteria that makes direct Java calls possible when both the client and target services are implemented based on the same SOA Infrastructure (that is, the same SOA server).
- The composite implementing the reference (target) service must be *active*. This condition requires the target composite to be up and running, which in turn ensures that the reference service is available.

  Note: The state of the target composite must be on and active. A stopped or retired state is *not* eligible for local optimization.

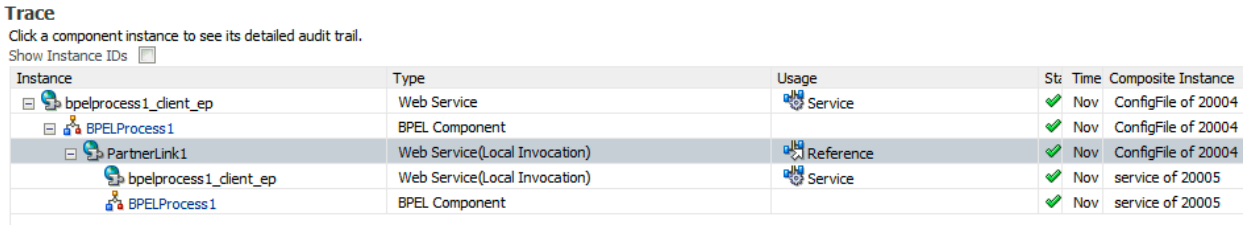
- The client and target composites must be co-located on the same server. This is an obvious requirement for direct Java invocations. It is also a critical step in which Oracle SOA Suite compares the server (on which the client composite is deployed) host configuration with the host and port values specified in the reference (target) service endpoint URI. If the host and port values match, it can be concluded that the client and target composites are located on the same server. However, the comparison is not necessarily straightforward given that working with both standalone and clustered server setups and potential load balancer configurations is necessary. Therefore, here are the step-by-step condition checks that determine the correct server configuration on all platforms:
  - Checks the **Server URL** configuration property value on the SOA Infrastructure Common Properties page, as described in Section 3.1, "Configuring SOA Infrastructure Properties."
  - If not specified, checks the **FrontendHost** and **FrontendHTTPPort** (or **FrontendHTTPSPort** if SSL is enabled) configuration property values from the cluster MBeans.

  - If not specified, checks the **FrontendHost** and **FrontendHTTPPort** (or **FrontendHTTPSPort** if SSL is enabled) configuration property values from the Oracle WebLogic Server MBeans.
  - If not specified, uses the DNS-resolved Inet address of `localhost`.
  - Checks if the port value specified in the reference service endpoint URL matches the configured server port value. If no port value is specified in the endpoint URL, Oracle SOA Suite assumes `80` for HTTP and `443` for HTTPS URLs.
  - If the port values match, the server URL (that is, `http(s)://`*`host`*`:`*`port`*, where *`host`* and *`port`* are obtained from the checks mentioned above) is then compared to the server URL in the reference endpoint address. The URLs are resolved to canonical values and the comparison also takes into account the cases in which the endpoint URL host is `localhost` or `127.0.0.1`.
  - Oracle SOA Suite concludes that the composites are co-located if the server URL comparison returns a value of true.

- The security policy configurations, if applied on either or both the client and server composites, must allow for local optimization. For information about policy configurations and local optimization, see Section 7.6.2, "Policy Attachments and Local Optimization in Composite-to-Composite Invocations."

You can confirm if a call went over local optimization in the **Trace** section of the Flow Trace page. The **(Local Invocation)** text for the reference and service of the invoking and invoked composites is displayed, as shown in Figure 3–1.

Figure 3–1 Local Optimization Details in the Trace Section of the Flow Trace Page



For more information about the Flow Trace page, see Section 13.1, "Monitoring the Audit Trail and Process Flow of a BPEL Process Service Component."

3.7.2 Overriding or Forcing Local Optimization

Two configuration properties are provided for either overriding or forcing local optimization.

By default, Oracle SOA Suite prefers local optimization. However, you can override this behavior with the `oracle.webservices.local.optimization` binding property in the `composite.xml` file. When this property is set to `false`, local optimization is not performed and cross-composite calls are performed through SOAP and HTTP. Use this property where appropriate. For information about setting this property, see Section 7.6.2, "Policy Attachments and Local Optimization in Composite-to-Composite Invocations."

You can override the `oracle.webservices.local.optimization` property and force optimization to be performed by setting the `oracle.soa.local.optimization.force` property to `true`. Use this property in the following scenarios:

- The server configuration is sufficiently complicated (for example, there are fire wall or proxy settings in an intranet), which may cause the co-location checks described in Section 3.7.1, "Condition Checks for Using Local Optimization" to not deliver the correct result.
- You clearly understand the semantics of local optimization, the system setup qualifies for local optimization, and local optimization is absolutely preferred.

There can be other scenarios for forcing optimization that are not described in this section.

> **Note:** If `oracle.webservices.local.optimization` is set to `false` and `oracle.soa.local.optimization.force` is set to `false`, local optimization is *not* performed.

The `oracle.soa.local.optimization.force` property has a default value of `false`. When this property is set to `true`, Oracle SOA Suite skips the condition checks described in Section 3.7.1, "Condition Checks for Using Local Optimization," except for policy configuration checking, which is necessary to ensure and enforce the integrity of service invocations.

Another important note about this property is that Oracle SOA Suite always honors the setting of this property (if policy checks allow the optimization). However, if local invocation fails due to nonapplication faults or exceptions (that is, runtime errors mostly related to the direct Java invocation), the value of this setting is ignored for subsequent invocations on the configured endpoint and for all the valid endpoint addresses configured on the endpoint.

To enable the `oracle.soa.local.optimization.force` property:

1. Add `oracle.soa.local.optimization.force` as a binding component level property in the `reference` section of the composite being invoked. For example, if composite `comp_comp2` invokes `comp_comp1`, then define this property in the `reference` section of the `composite.xml` file of `comp_comp2`.

```
<reference name="Service1"
ui:wsdlLocation="http://localhost:8001/soa-infra/services/default/comp_
comp1!1.0/BPELProcess1.wsdl">
<interface.wsdl interface="http://xmlns.oracle.com/comp_comp/comp_
comp1/BPELProcess1#wsdl.interface(BPELProcess1)"
                            callbackInterface="http://xmlns.oracle.com/comp_
comp/comp_comp1/BPELProcess1#wsdl.interface(BPELProcess1Callback)"/>
<binding.ws port="http://xmlns.oracle.com/comp_comp/comp_
comp1/BPELProcess1#wsdl.endpoint(bpelprocess1_client_ep/BPELProcess1_pt)"
location="http://localhost:8001/soa-infra/services/default/comp_
comp1!1.0/bpelprocess1_client_ep?WSDL">
<property name="oracle.webservices.local.optimization">false</property>
<property name="oracle.soa.local.optimization.force">true</property>
</binding.ws>
```

3.7.3 Local Optimization Logging

Oracle SOA Suite provides **NOTIFICATION:1(INFO)** level logging for every critical decision made for local optimization versus SOAP processing. For more details or debugging information, set the **oracle.integration.platform.blocks.local** and **oracle.integration.platform.blocks.soap** loggers at the **TRACE:1(FINE)** level in Oracle Enterprise Manager Fusion Middleware Control.

3.7.4 Local Optimization Calls Use Case

This local optimization calls use case describes the following:

- How local optimization calls work in an environment in which composite A calls a co-located composite B on the same server, and composite B is unreachable (Table 3–2).
- What happens when you create a load balancer address with a port value that is not the port on which the Oracle WebLogic Servers are listening (Table 3–3).

Table 3–2 Local Optimization Calls When a Composite is Unreachable

Scenario	Description
What happens when a local optimization call fails?	A check of composite B is performed *before* trying a local optimization call. If the check fails (composite B is unreachable), an exception is thrown in the SOA Infrastructure that is converted to a BPEL fault. The BPEL fault contains information about composite B being unreachable.
Is it possible to retry the co-located call?	After the basic check is performed, a local optimization call is tried. If that call fails, it is reinvoked over SOAP. However, the following conditions must be met for the reinvocation over SOAP to occur: ▪ The `oracle.soa.local.optimization.force` WS-binding property must be set to `true` to force local optimization. This condition is provided for backward compatibility. ▪ The exception must not be a business exception.
If a retry of the co-located call is possible (for example, the fault policy is set to retry on the composite or endpoint), is the call optimized again or does it attempt to leave the local container and access the load balancer?	If the call fails the first time you attempt to send it locally, the information is cached (that it failed locally). For subsequent calls, the call is sent over SOAP (the local optimization call is not retried this time).

Table 3–3 Creating a Load Balancer Address with a Port Value That is Not the Port on which the Oracle WebLogic Servers are Listening

Scenario	Description
If you want to create a load balancer address with a port value that is not the port on which the Oracle WebLogic Servers are listening, can you specify the server URL (in the SOA Infrastructure Common Properties page) and the frontend host/port (in the Oracle WebLogic Server HTTP tab) as the address of the load balancer?	Yes. The server URL or frontend host/port are more identifiers of the address for the local optimization rules than the actual addresses to which to send network requests.
Does local optimization not use the port to make a call (for example, composite A calling composite B over port `2011`)?	Yes, the port is used only to make comparisons to see if the target is co-located, so as to make a local call.

4

Monitoring the SOA Infrastructure

This chapter describes how to monitor recent instances and faults, deployed SOA composite applications, message delivery processing requests, and service and reference binding components in the SOA Infrastructure. All SOA composite applications are deployed to the SOA Infrastructure. It also describes how to display details about BPEL process messages that require recovery and how to access the Recovery page of the BPEL process service engine to perform message recovery.

This chapter includes the following topics:

- Section 4.1, "Monitoring SOA Infrastructure Recent Instances and Faults and Deployed Composites"
- Section 4.2, "Monitoring Message Delivery Processing Requests"
- Section 4.3, "Monitoring Service and Reference Binding Components in the SOA Infrastructure"

For more information, see Section 1.2.1, "Introduction to the SOA Infrastructure Application."

4.1 Monitoring SOA Infrastructure Recent Instances and Faults and Deployed Composites

You can monitor the SOA composite applications deployed to the SOA Infrastructure.

To monitor SOA Infrastructure recent instances and faults:

1. Access this page through one of the following options:

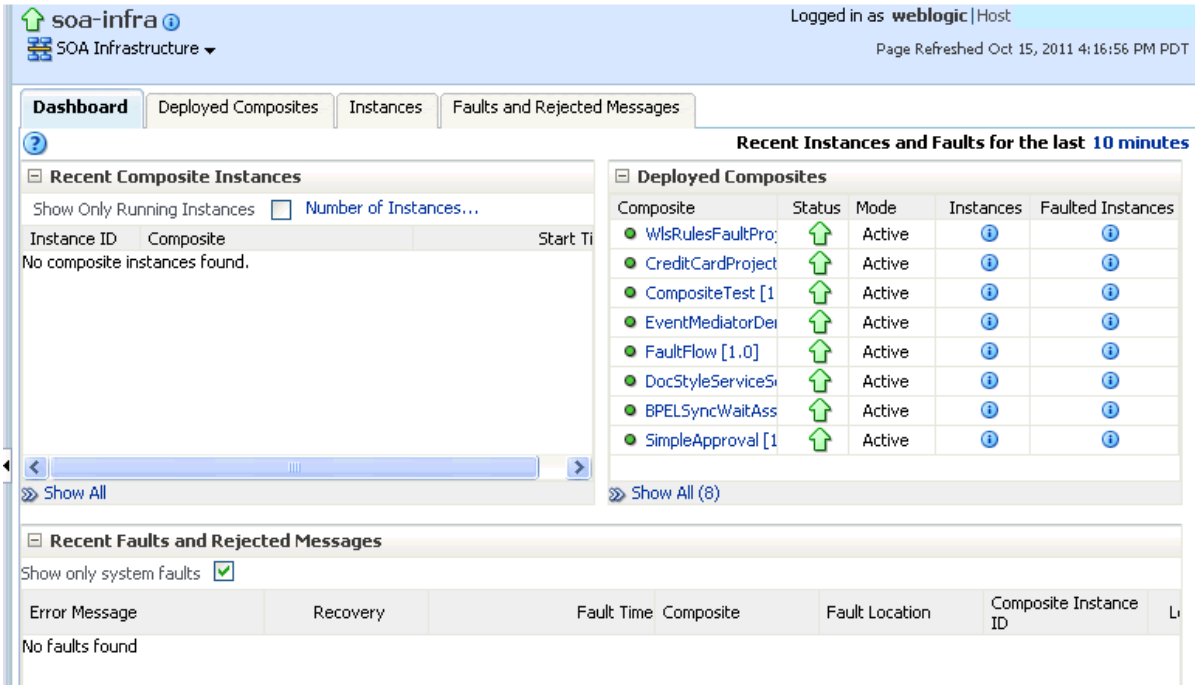
From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...	From the SOA Composite Menu...
1. Select **Home**.	1. Select **soa-infra**.	1. Select **SOA Infrastructure**.

The upper part of the SOA Infrastructure Dashboard page displays the following details:

- A message indicating that the retrieval of recent instances and faults that are displayed on this page is restricted to the specified time period. This message is displayed if the **Restrict display of instances and faults to the last *time_period*** checkbox is selected on the SOA Infrastructure Common Properties page (it is selected by default). The default time period value is 24 hours, but you can change this value. If this checkbox is not selected, all instances and faults (including count metrics) in the SOA Infrastructure since the last purging are displayed.

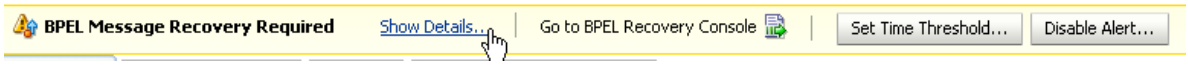
- Recent SOA composite application instances, instance IDs, and starting times. By default, only running instances are shown.
- The status of deployed SOA composite applications and their revision numbers, the number of instances created for each application, and the number of faulted instances in each application. The total number of deployed composites also is displayed in parentheses next to the **Show All** link.
- Recent faults and rejected messages, including the error message, whether you can recover from the fault, the time at which the fault occurred, the SOA composite application in which the fault occurred, the location of the fault (service binding component, service component, or reference binding component), the instance ID of the SOA composite application, and a link to log messages describing the fault or rejected message. You can recover from faults identified as recoverable at the SOA Infrastructure, SOA composite application, service engine, and service component levels.

  If messages are awaiting recovery from the Recovery page of the BPEL process service engine, a message is displayed at the top of each SOA Infrastructure home page.



Note: After the SOA Infrastructure is started, it may not be completely initialized to administer incoming requests until all deployed composites are loaded. During SOA Infrastructure initialization, a warning message is displayed at the top of the SOA Infrastructure home page. Do not perform operations such as composite deployment, composite undeployment, and others while this message is displayed. For more information, see Section 3.2.1, "Waiting for SOA Infrastructure Startup Initialization to Complete."

2. If BPEL process messages require recovery from the Recovery page of the BPEL process service engine, the **BPEL Message Recovery Required** message is displayed at the top of each SOA Infrastructure home page.



The display of this message recovery information is controlled by the **bpelRecoveryStatus** and **excludeBpelMaxCreationTime** keys of the **AuditConfig** property in the System MBean Browser. This property is accessible from the **More SOA Infra Advanced Configuration Properties** section of the SOA Infrastructure Common Properties page. By default, the **bpelRecoveryStatus** key is set to **All** and the **excludeBpelMaxCreationTime** key is set to exclude the display of messages requiring recovery in the last five minutes.

a. Perform the following tasks:

Click This Link...	To...
Show Details	View a message showing the number of invoke, callback, and activity messages that require recovery from the Recovery page of the BPEL process service engine. Manual recovery is not required if automatic recovery of BPEL messages is enabled. In that case, only exhausted messages must be manually recovered. Click **Refresh** to recalculate the number of invoke, callback, and activity messages requiring recovery.
Go to BPEL Recovery Console	Access the Recovery page of the BPEL process service engine to perform message recovery. You can also access the Recovery page later by selecting **Service Engines** > **BPEL** from the **SOA Infrastructure** menu and clicking the **Recovery** tab in the resulting page.
Set Time Threshold	View a message showing the time period for excluding messages that require recovery. By default, this setting excludes the last five minutes of messages. This value is controlled by the **excludeBpelMaxCreationTime** key of the **AuditConfig** property on the SOA Infrastructure Common Properties page. To change this value, click **Yes** to access this property in the System MBean Browser.
Disable Alert	View a message enabling you to prevent the display of this message recovery information on the Dashboard page. If you click **Yes**, this message recovery information is not displayed. To display this information again on the Dashboard page, set the **bpelRecoveryStatus** key to **All** for the **AuditConfig** property in the **More SOA Infra Advanced Configuration Properties** section of the SOA Infrastructure Common Properties page. For more information, see Section 3.1, "Configuring SOA Infrastructure Properties."

3. In the **Recent Composite Instances** section, perform the following tasks:

a. Click the **Number of Instances** link to display a message showing the numbers of running and total instances in the SOA Infrastructure. The link is not displayed by default. To enable this link to be displayed, select the **Disable fetching of instance and fault count metrics. Each metric can still be retrieved on demand** checkbox in the SOA Infrastructure Common Properties page.

If you selected the **Restrict display of instances and faults to the last *time_period*** checkbox on the SOA Infrastructure Common Properties page and specified a time period or accepted the default value of 24 hours, the numbers of running and total instances in the SOA Infrastructure for that time period are displayed. If you did not select this checkbox, all instances and faults in the SOA Infrastructure since the last purging are displayed. Click **Recalculate** to recalculate the numbers.

b. In the **Instance ID** column, click a specific instance ID to show the message flow through the various service components and binding components.

c. In the **Composite** column, click a specific SOA composite application to access its home page.

d. Click **Show All** below the section to access the Instances page of the SOA Infrastructure.

4. In the **Deployed Composites** section, perform the following tasks:

   a. In the **Composite** column, click a specific SOA composite application to access its home page.

   b. Click **Show All** below the section to access the Deployed Composites page of the SOA Infrastructure.

5. In the **Recent Faults and Rejected Messages** section, perform the following tasks:

   a. In the **Error Message** column, click an error message to display complete information about the fault. If the fault is identified as recoverable, click the **Recover Now** link to perform fault recovery.

   b. In the **Recovery** column, if a fault is identified as recoverable, click **Recover** to perform fault recovery.

   c. In the **Composite** column, click a SOA composite application to access its home page.

   d. In the **Fault Location** column, click a specific location to access the home page of the service, component, or reference in which the fault occurred.

   e. In the **Composite Instance ID** column, click a composite instance ID to access the flow trace of the message that contains that fault.

   f. In the **Logs** column, click a specific log to access the Log Messages page, with the search criteria prefiltered to display any log messages related to the fault.

   g. Click **Show All** below the section to access the Recent Faults and Rejected Messages page of the SOA Infrastructure.

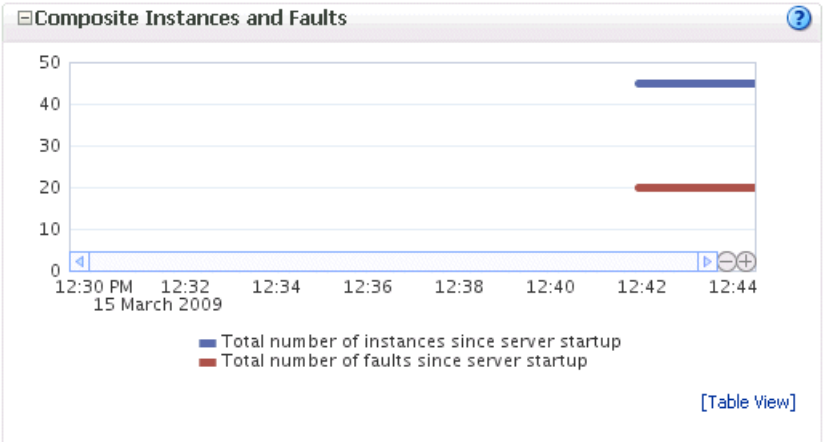
   The lower part of the SOA Infrastructure Dashboard page displays the following details:

   - The number of service components running in the service engines (BPEL process, BPMN process (if Oracle BPM Suite is installed), Oracle Mediator, human workflow, business rules, and spring) and the number of faulted instances for each service engine.

Service Engines

Name	Number of Components	Faults
BPEL Engine	94	186
BPMN Engine	0	0
Mediator Engine	1	0
Human Workflow Engine	2	0
Business Rules Engine	1	0
Spring Engine	0	0

   - A graphical representation of the total number of instances and faults for all SOA composite applications since the SOA Infrastructure was last restarted.



6. In the **Name** column of the **Service Engines** section, click a specific service engine to access its home page. If you click **Spring Engine**, a message is displayed indicating that it cannot be managed.

For more information, see the following sections:

- Section 1.2.4, "Introduction to Service Components and Service Component Instances"
- Section 1.2.6, "Introduction to Service Engines"
- Section 1.4.3.1, "Introduction to Fault Recovery"
- Section 8.1, "Initiating a SOA Composite Application Test Instance"
- Section 14.4, "Performing BPEL Process Service Engine Message Recovery"
- *Oracle Fusion Middleware Administrator's Guide* for details about viewing and searching log files

4.2 Monitoring Message Delivery Processing Requests

You can monitor SOA Infrastructure message delivery processing requests. These are metrics for the message delivery between the service engines, service infrastructure, and binding components. Once a message is handed over to a service engine, the amount of time it takes to process that message (instance processing time) is *not* captured in these metrics.

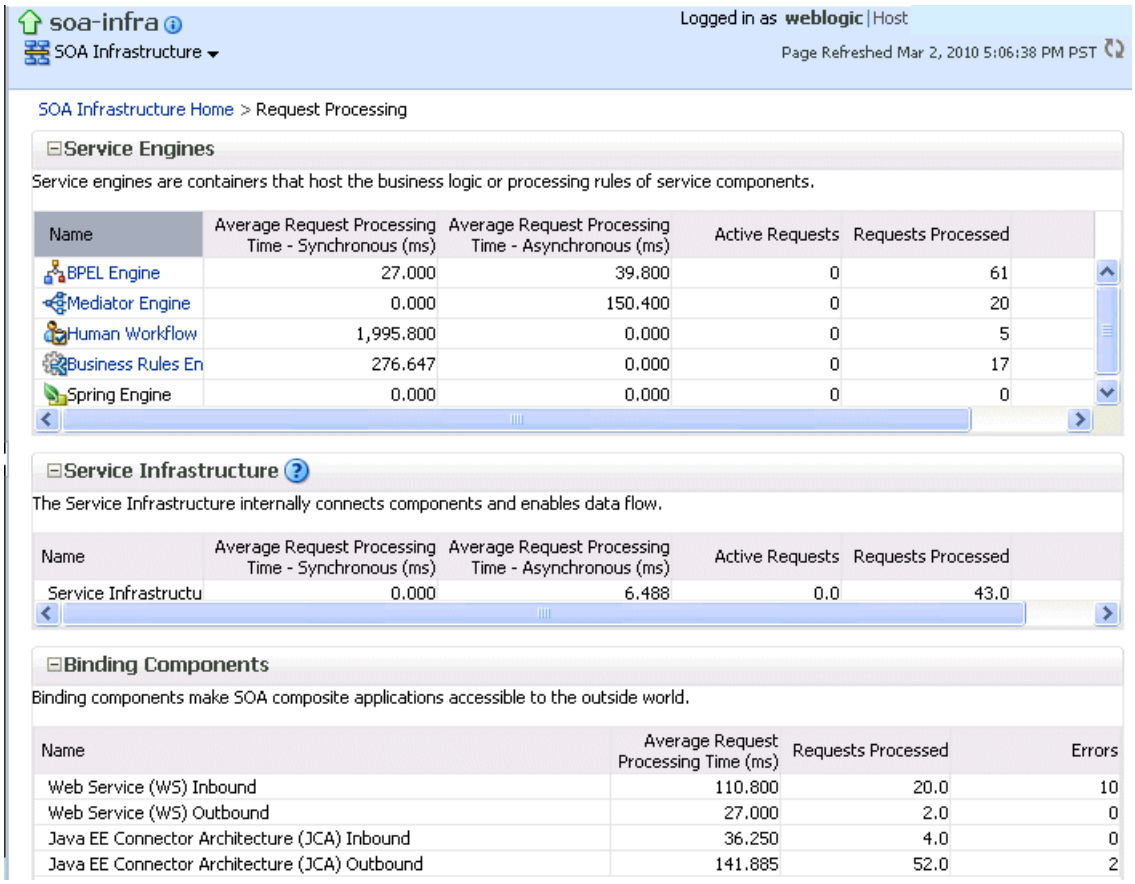
To monitor processing requests:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Monitoring** > **Request Processing**.	1. Right-click **soa-infra**. 2. Select **Monitoring** > **Request Processing**.

The Request Processing page enables you to monitor the following details:

- The average request processing time for both synchronous and asynchronous messages, active requests, requests processed, and faulted requests in the service engines and service infrastructure.
- The average request processing time, requests processed, and errors occurring in service (inbound) and reference (outbound) binding components.



2. In the **Service Engines** section, click a specific service engine (for example, **BPEL Engine**) to access details such as recent instances using this service engine, components using this service engine, and recent fault occurrences.

For more information, see the following sections:

- Section 1.2.5, "Introduction to Binding Components"
- Section 1.2.6, "Introduction to Service Engines"
- Section 1.2.7, "Introduction to the Service Infrastructure"

4.3 Monitoring Service and Reference Binding Components in the SOA Infrastructure

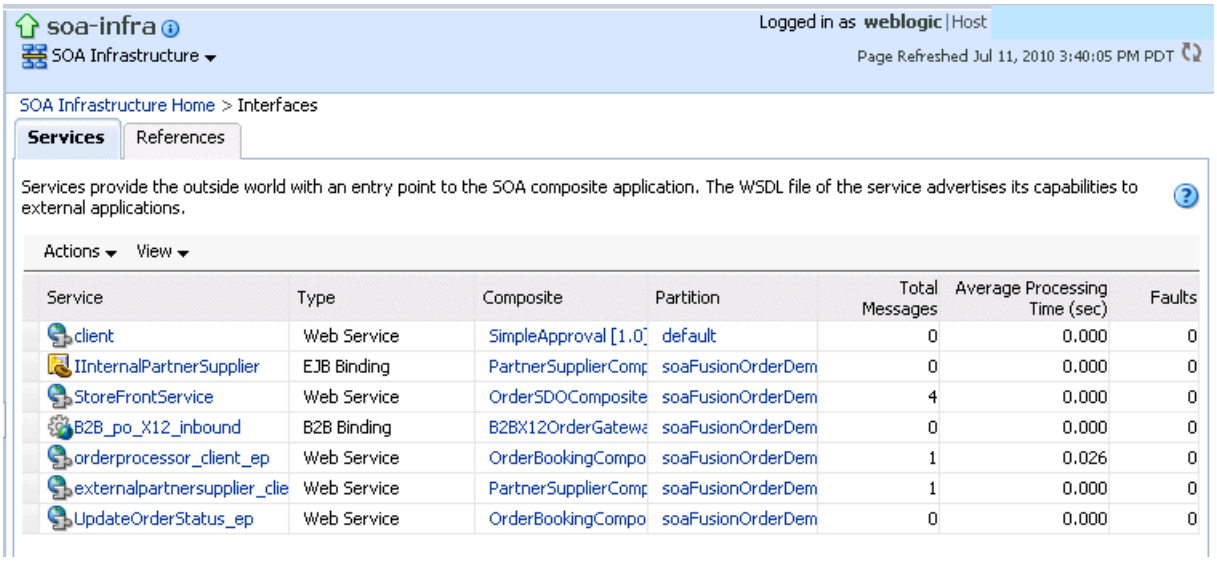
You can monitor all service and reference binding components used in all SOA composite applications deployed to the SOA Infrastructure. Services provide the outside world with an entry point to the SOA composite application. The WSDL file of the service advertises its capabilities to external applications. References enable messages to be sent from the SOA composite application to external services in the outside world.

To monitor service and reference binding components in the SOA Infrastructure:

1. Access this page through one of the following options:

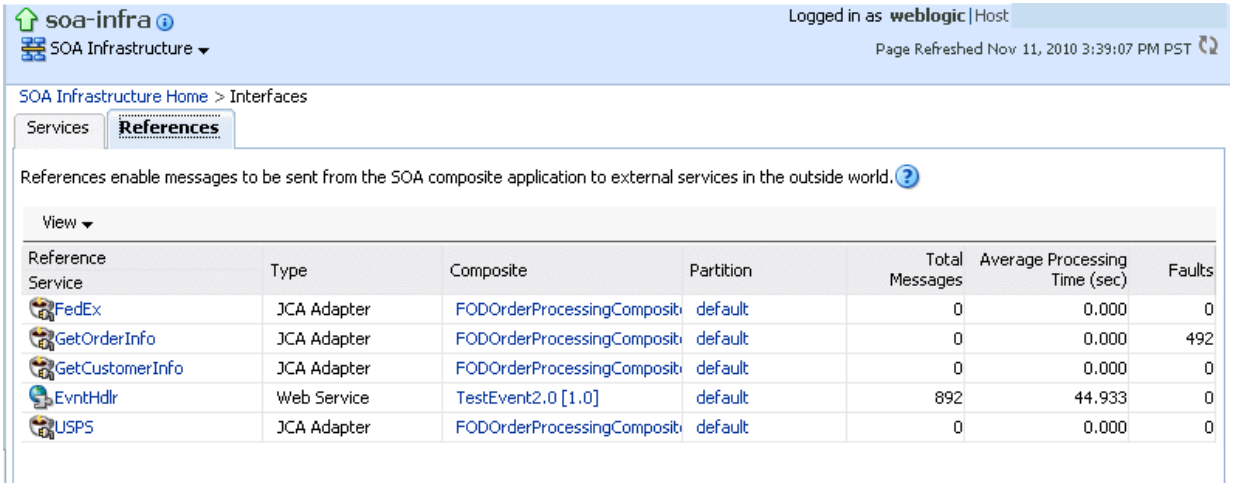
From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Services and References**.	1. Right-click **soa-infra**. 2. Select **Services and References.**

The Services page displays details about the names and types of the services, the SOA composite applications in which the services are used, the partitions in which the SOA composite applications are deployed, the total number of messages processed, the average processing time, and the number of faults occurring in the services.



2. In the **Service** column, click a specific service to access its home page.
3. In the **Composite** column, click a specific SOA composite application to access its home page.
4. In the **Partitions** column, click a specific partition to access its home page.
5. Click the **References** tab.

   The References page displays details about the names and types of the references, the SOA composite applications in which the references are used, the partitions in which the SOA composite applications are deployed, the total number of messages processed, the average processing time, and the number of faults occurring in the references.



6. In the **Reference** column, click a specific reference to access its home page.
7. In the **Composite** column, click a specific SOA composite application to access its home page.
8. In the **Partitions** column, click a specific partition to access its home page.

For more information about services and references, Section 1.2.5, "Introduction to Binding Components."

Part IV

Administering SOA Composite Applications and Instances

This part describes how to administer SOA composite applications and instances.

This part includes the following chapters:

- Chapter 5, "Securing SOA Composite Applications"
- Chapter 6, "Monitoring SOA Composite Applications"
- Chapter 7, "Deploying and Managing SOA Composite Applications"
- Chapter 8, " Managing SOA Composite Application Instances"
- Chapter 9, "Managing Database Growth"
- Chapter 10, "Programmatically Managing SOA Composite Applications"
- Chapter 11, "Diagnosing Problems with SOA Composite Applications"

5

Securing SOA Composite Applications

This chapter describes security configuration procedures unique to SOA composite applications. Most SOA composite application security procedures do not require SOA-unique configuration steps. References are provided to additional documentation for performing those tasks.

This chapter includes the following topics:

- Section 5.1, "Introduction to Securing SOA Composite Applications"
- Section 5.2, "Configuring Oracle HTTP Server with Oracle BPM Worklist"
- Section 5.3, "Setting up SAML Message-Protected Policy Configuration for the SOA Infrastructure"
- Section 5.4, "Automatically Authenticating Oracle BPM Worklist and Oracle Business Process Management Users"
- Section 5.5, "Listing the Authentication Provider"
- Section 5.6, "Configuring SSL"
- Section 5.7, "Configuring Security for Human Workflow WSDL Files"

Note: See the following sections for information on attaching and detaching policies:

- Section 7.6, "Managing SOA Composite Application Policies"
- Section 14.2, "Managing BPEL Process Service Component Policies"
- Section 17.2, "Managing Oracle Mediator Policies"
- Section 22.1, "Managing Human Task Service Component Policies"
- Section 36.1, "Managing Binding Component Policies"

5.1 Introduction to Securing SOA Composite Applications

This chapter describes security procedures unique to SOA composite applications. Most SOA composite application security procedures do not require SOA-unique steps and can be performed by following the documentation listed in Table 5–1.

Table 5–1 Security Documentation

For Information On...	See The Following Guide...
Securing Oracle Fusion Middleware, including Oracle Single Sign-On (OSSO) configuration	*Oracle Fusion Middleware Application Security Guide*
Securing and administering web services	*Oracle Fusion Middleware Security and Administrator's Guide for Web Services*
Understanding Oracle WebLogic Server security	*Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server*
Securing an Oracle WebLogic Server production environment	*Oracle Fusion Middleware Securing a Production Environment for Oracle WebLogic Server*
Securing Oracle WebLogic Server	*Oracle Fusion Middleware Securing Oracle WebLogic Server*
Developing new security providers for use with Oracle WebLogic Server	*Oracle Fusion Middleware Developing Security Providers for Oracle WebLogic Server*
Securing web services for Oracle WebLogic Server	*Oracle Fusion Middleware Securing WebLogic Web Services for Oracle WebLogic Server*
Programming security for Oracle WebLogic Server	*Oracle Fusion Middleware Programming Security for Oracle WebLogic Server*
Oracle Business Activity Monitoring	Section 23.9, "Configuring Security"
Securing Oracle User Messaging Service	Section 26.6, "Securing the Oracle User Messaging Service"

5.2 Configuring Oracle HTTP Server with Oracle BPM Worklist

You must add the `/integration` location in the `mod_wl_ohs.conf` file of Oracle HTTP Server for Oracle BPM Worklist to work through Oracle HTTP Server.

```
<Location /integration>
     SetHandler weblogic-handler
 #   PathTrim /weblogic
     ErrorPage  http:/WEBLOGIC_HOME:WEBLOGIC_PORT/
</Location>
```

5.3 Setting up SAML Message-Protected Policy Configuration for the SOA Infrastructure

This section describes how to set up and validate Security Assertion Markup Language (SAML) message-protected policy configuration for the SOA Infrastructure with the Oracle WebLogic Scripting Tool (WLST). The example in this section describes task query service configuration. However, these instructions are relevant to all human workflow services that support SAML-token ports:

- Activity guide query service
- Activity guide metadata service
- Activity guide admin service
- Task query service
- Task service

- Task metadata service
- Runtime config service
- Task evidence service
- User metadata service

If you want to change the policy for another service, you must apply the same WLST commands to that service's SAML-token port.

To set up an SAML message-protected policy configuration:

1. Log in to the SOA domain (for example, named `base_domain`) using WLST.
2. Detach the existing out-of-the-box service policy named `wss10_saml_token_service_policy`.

```
wls:/base_domain/domainRuntime> detachWebServicePolicy('/base_domain/soa
_server1/soa-infra','integration/services/TaskQueryService','web',
'WorkflowProvider','TaskQueryServicePortSAML','oracle/
wss10_saml_token_service_policy')
```

3. Restart the application to activate any policy or configuration change.
4. Attach the new policy. In this case, the policy is named `oracle/wss10_saml_token_with_message_protection_service_policy`.

```
wls:/base_domain/domainRuntime> attachWebServicePolicy('/base_domain/soa
_server1/soa-infra','integration/services/TaskQueryService',
'web','WorkflowProvider','TaskQueryServicePortSAML','ora
cle/wss10_saml_token_with_message_protection_service_policy')
```

5. Restart the application to activate any policy or configuration change.
6. List the policy to validate.

```
wls:/base_domain/domainRuntime> listWebServicePolicies('/base_domain/soa
_server1/soa-infra','integration/services/TaskQueryService',
'web','WorkflowProvider','TaskQueryServicePortSAML')
    TaskQueryServicePortSAML :
    security :
oracle/wss10_saml_token_with_message_protection_service_policy,
    enabled=true
Attached policy or policies are valid; endpoint is secure.
```

7. Create a keystore, add the `orakey` alias, and run the Oracle Web Service Manager (OWSM) configuration to activate the SAML message-protected policy. For example:

```
keytool -genkeypair
        -keystore  domain_home/config/fmwconfig/default-keystore.jks
        -keyalg RSA
        -dname "cn=consumer,dc=example,dc=com"
        -alias clientalias
        -keypass  password
        -storepass password
        -validity 3600

keytool -exportcert
        -keystore domain_home/config/fmwconfig/default-keystore.jks
        -v
        -alias clientalias
        -storepass password
        -rfc
```

```
        -file domain_home/config/fmwconfig/certificate.cer

keytool -importcert
        -keystore domain_home/config/fmwconfig/default-keystore.jks
        -alias orakey
        -file domain_home/config/fmwconfig/certificate.cer
        -storepass password
         createCred(map="oracle.wsm.security", key="keystore-csf-key",
         user="owsm", password="welcome1", desc="Keystore key")
         createCred(map="oracle.wsm.security", key="enc-csf-key",
         user="clientalias", password="welcome1", desc="Encryption key")
         createCred(map="oracle.wsm.security", key="sign-csf-key",
         user="clientalias", password="welcome1", desc="Signing key")
```

8. Restart the servers.

5.4 Automatically Authenticating Oracle BPM Worklist and Oracle Business Process Management Users

This section describes how to authenticate Oracle BPM Worklist and Oracle Business Process Management users in different environments.

5.4.1 Automatically Authenticating Oracle BPM Worklist Users in SAML SSO Environments

To be automatically authenticated when accessing a second Oracle BPM Worklist from a first Oracle BPM Worklist in SAML SSO environments, you must perform the following steps. Otherwise, you are prompted to log in again when you access the second Oracle BPM Worklist. In these environments, the first Oracle BPM Worklist is configured as the SAML identity provider and the second Oracle BPM Worklist that you access is configured as the SAML service provider.

To automatically authenticate Oracle BPM Worklist users in SAML SSO environments:

1. Add `/integration/worklistapp/*` as the redirect URL for `worklistapp` to the SAML service provider site's `SAML2IdentityAsserter` configuration as follows.

   a. In the Oracle WebLogic Server Administration Console, select **Security Realms**.

   b. Click the realms for the service providers.

   c. Select the **Providers** tab, and then the **Authentication** subtab.

   d. From the provider list, select the provider with the description **SAML 2.0 Identity Assertion Provider.**

      If you do not see the SAML identity assertion provider configuration, follow the instructions in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

   e. Select the **Management** tab.

      Under the **Management** tab, you see a list of identity provider partners. These are hosts that have been configured as the SAML identity provider partners for this SAML identity service provider site. Remember that this configuration step is performed on the identity service provider site on which Oracle BPM Worklist is hosted.

f. Select the identity provider site where you want the user to perform the initial login.

g. Scroll down the page until you see the field **Redirect URIs**.

h. Add `/integration/worklistapp/*` to the list.

After performing this step, you can log in to Oracle BPM Worklist at the SAML identity provider site though the regular URL of `/integration/worklistapp`. If necessary, you can then navigate to the URL `/integration/worklistapp/ssologin` at the SAML service provider site, where you gain access to Oracle BPM Worklist and are automatically authenticated.

For more information about `SAML2IdentityAsserter` and configuring SSO with web browsers and HTTP clients, see *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

5.4.2 Automatically Authenticating Oracle BPM Worklist Users in Windows Native Authentication Environments

For Windows native authentication through Kerberos to work with Oracle BPM Worklist, you must use the `/integration/worklistapp/ssologin` protected URL. For example, after configuring Windows native authentication, you access Oracle BPM Worklist as follows:

```
http://host_name.domain_name:8001/integration/worklistapp/ssologin
```

For information on configuring SSO with Microsoft clients, see *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

5.4.3 Automatically Authenticating Oracle Business Process Management Process Composer Users in Windows Native Authentication Environments

For Windows native authentication through Kerberos to work with Oracle Business Process Management Process Composer, you must use the `bpm/composer/ssologin` protected URL. For example, after configuring Windows native authentication, you access Process Composer as follows:

```
http://host_name.domain_name:8001/bpm/composer/ssologin
```

For information on configuring SSO with Microsoft clients, see *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

5.5 Listing the Authentication Provider

This section describes how to set the first authentication provider.

5.5.1 Listing Oracle Internet Directory as the First Authentication Provider

The Oracle BPM Worklist and workflow services use Java Platform Security (JPS) and the User and Role API. For this reason, the Oracle Internet Directory authenticator must be the first provider listed when workflow is used with Oracle Internet Directory. If Oracle Internet Directory is not listed first (for example, it is listed below `DefaultAuthenticator`), login authentication fails.

For information about changing the order of authentication providers, see *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

5.5.2 Accessing Web-based Applications with the Default Authentication Provider

Logging in to web-based applications may fail when using Oracle Internet Directory authentication. This is caused when the Oracle WebLogic Server configuration is set to use the Oracle Internet Directory authentication before default authentication.

This may produce the following error:

```
"@ User "weblogic" is not found in configuration "jazn.com" Check if the user
 exists in the repository specified by the configurations. Check the error stack
 and fix the cause of the error. Contact oracle support if error is not fixable."
```

The order of the security providers should be:

1. Default authentication
2. Oracle Internet Directory/LDAP authentication

5.6 Configuring SSL

This section describes configuring secure socket layer (SSL) in Oracle SOA Suite and Oracle Business Process Management environments.

5.6.1 Recommendation to Configure Either All or No Managed Servers with SSL

As a best practice, Oracle recommends that you configure either *all* managed servers or *no* managed servers with SSL (SOA, BAM, and so on). Configuring some managed servers with SSL, while not configuring others, may lead to undesirable results in Oracle BPM Worklist and Oracle Web Services Manager (OWSM). For example, if there is an SSL-configured, managed server (bam_server), servers not configured with SSL are not used by OWSM. In cases in which an SSL-configured server is down, it causes OWSM to be in a down state, which in turn causes Oracle BPM Worklist to be in a down state.

5.6.2 Switching from Non-SSL to SSL Configurations with Oracle BPM Worklist

Switching from non-SSL to SSL configurations with Oracle BPM Worklist requires the **Frontend Host** and **Frontend HTTPS Port** fields to be set in Oracle WebLogic Server Administration Console. Not doing so results in exception errors when you attempt to create to-do tasks.

To switch from non-SSL to SSL configurations with Oracle BPM Worklist:

1. Log in to Oracle WebLogic Server Administration Console.
2. In the **Environment** section, select **Servers**.
3. Select the name of the managed server (for example, **soa_server1**).
4. Select **Protocols**, then select **HTTP**.
5. In the **Frontend Host** field, enter the hostname on which Oracle BPM Worklist is located.
6. In the **Frontend HTTPS Port** field, enter the SSL listener port.
7. Click **Save**.

5.6.3 Configuring SOA Composite Applications for Two-Way SSL Communication

Oracle SOA Suite uses both Oracle WebLogic Server and Sun Secure Socket Layer (SSL) stacks for two-way SSL configurations.

- For the inbound web service bindings, Oracle SOA Suite uses the Oracle WebLogic Server infrastructure and, therefore, the Oracle WebLogic Server libraries for SSL.
- For the outbound web service bindings, Oracle SOA Suite uses JRF HttpClient and, therefore, the Sun JDK libraries for SSL.

Due to this difference, start Oracle WebLogic Server with the following JVM option.

To configure SOA composite applications for two-way SSL communication:

1. Open the following file:
   - On UNIX operating systems, open `$MIDDLEWARE_HOME/user_projects/domains/`*`domain_name`*`/bin/setDomainEnv.sh`.
   - On Window operating systems, open `MIDDLEWARE_HOME\user_projects\domains\`*`domain_name`*`\bin\setDomainEnv.bat`.
2. Add the following lines in the `JAVA_OPTIONS` section, if the server is enabled for one-way SSL (server authorization only):

   ```
   -Djavax.net.ssl.trustStore=your_truststore_location
   ```

   For two-way SSL, the keystore information (location and password) is not required.

In addition, perform the following steps to enable two-way SSL for a SOA composite application to invoke another SOA composite application or another non-SOA application.

> **Note:** Both the server and client are assumed to have been configured for SSL with mutual authentication.

To enable two-way SSL for a SOA composite application to invoke another application:

1. On the client side, provide the keystore location.
   a. From the **SOA Infrastructure** menu, select **SOA Administration** > **Common Properties**.
   b. At the bottom of the page, click **More SOA Infra Advanced Configuration Properties**.
   c. Click **KeystoreLocation**.
   d. In the **Value** column, enter the keystore location.
   e. Click **Apply**.
   f. Click **Return**.
2. During design time in Oracle JDeveloper, update the reference section in the `composite.xml` file with the `oracle.soa.two.way.ssl.enabled` property.

   ```
   <reference name="Service1"
      ui:wsdlLocation=". . .">
      <interface.wsdl interface=". . ."/>
        <binding.ws port=". . .">
         <property name="oracle.soa.two.way.ssl.enabled">true</property>
     </binding.ws>
    </reference>
   ```

3. In Oracle Enterprise Manager Fusion Middleware Control, select **WebLogic Domain** > ***domain_name***.
4. Right-click ***domain_name*** and select **Security** > **Credentials**.
5. Click **Create Map**.
6. In the **Map Name** field, enter a name (for example, `SOA`), and click **OK**.
7. Click **Create Key**.
8. Enter the following details.

Field	Description
Select Map	Select the map created in Step 6 (for this example, **SOA**).
Key	Enter the key name (`KeystorePassword` is the default).
Type	Select **Password**.
User Name	Enter the keystore user name (`KeystorePassword` is the default).
Password	Enter the password that you created for the keystore.

Note: When you set up SSL in Oracle WebLogic Server, a key alias is required. You must enter `mykey` as the alias value. This value is required.

9. Set the keystore location in Oracle Enterprise Manager Fusion Middleware Control. See Step 1 for instructions.
10. Modify `composite.xml` to use `https` and *`sslport`* to invoke a SOA composite application. For example, change the syntax shown in bold:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Generated by Oracle SOA Modeler version 1.0 at [4/1/09 11:01 PM]. -->
<composite name="InvokeEchoBPELSync"
revision="1.0"
label="2009-04-01_23-01-53_994"
mode="active"
state="on"
xmlns="http://xmlns.oracle.com/sca/1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:orawsp="http://schemas.oracle.com/ws/2006/01/policy"
xmlns:ui="http://xmlns.oracle.com/soa/designer/">
<import
namespace="http://xmlns.oracle.com/CustomApps/InvokeEchoBPELSync/BPELProcess1"
  location="BPELProcess1.wsdl" importType="wsdl"/>
<import namespace="http://xmlns.oracle.com/CustomApps/EchoBPELSync/
BPELProcess1"location="http://hostname:port/soa-infra/services/default/EchoBPEL
Sync/BPELProcess1.wsdl"
importType="wsdl"/>
```

to use `https` and *`sslport`*:

```
location="https://hostname:sslport/soa-infra/services/default/EchoBPELSync
/BPELProcess1.wsdl"
```

5.6.4 Invoking References in One-Way SSL Environments in Oracle JDeveloper

When invoking a web service as an external reference from a SOA composite application in one-way SSL environments, ensure that the certificate name (CN) and the hostname of the server exactly match. This ensures a correct SSL handshake.

For example, if a web service is named `adfbc` and the certificate has a server name of `myhost05`, the following syntax results in an SSL handshake exception.

```
  <import namespace="/adfbc1/common/"

 location="https://myhost05.us.oracle.com:8002/CustomApps-adfbc1-context-root/Ap
pModuleService?WSDL"
          importType="wsdl"/>
 <import namespace="/adfbc1/common/" location="Service1.wsdl"
          importType="wsdl"/>
```

If you switch the order of `import`, the SSL handshake passes.

```
<import namespace="/adfbc1/common/" location="Service1.wsdl"
          importType="wsdl"/>
<import namespace="/adfbc1/common/"

location="https://myhost05.us.oracle.com:8002/CustomApps-adfbc1-context-root/Ap
pModuleService?WSDL"
          importType="wsdl"/>
```

Note the following restrictions around this issue:

- There are no options for ignoring hostname verification in Oracle JDeveloper as exist with the Oracle WebLogic Server Administration Console. This is because the SSL kit used by Oracle JDeveloper is different. Only the trust store can be configured from the command line. All other certificate arguments are not passed.
- In the WSDL file, `https://`*`hostname`* must match with that in the certificate, as described above. You cannot perform the same procedures as you can with a browser. For example, if the hostname is `myhost05.us.oracle.com` in the certificate's CN, then you can use `myhost05`, `myhost05.us.oracle.com`, or the IP address from a browser. In Oracle JDeveloper, always use the same name as in the certificate (that is, `myhost05.us.oracle.com`).

5.6.5 Configuring Oracle SOA Suite and Oracle HTTP Server for SSL Communication

Follow these steps to configure SSL communication between Oracle SOA Suite and Oracle HTTP Server.

5.6.5.1 Configuring Oracle HTTP Server for SSL Communication

To configure Oracle HTTP server for SSL communication:

1. Update `mod_ssl.conf` with the `<Location /integration/services>` location directive.

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

<IfModule mod_weblogic.c>
      WebLogicHost host.domain.com
      WLLogFile <logdir>/ohs_ssl.log
      Debug ALL
      DebugConfigInfo ON
      SecureProxy ON
      MatchExpression *.jsp
```

```
      WlSSLWallet <OHS_
HOME>/instances/instance1/config/OHS/ohs1/keystores/default
</IfModule>

<Location /soa-infra>
      WebLogicPort 8002
      SetHandler weblogic-handler
      ErrorPage http://host.domain.com:port/error.html
</Location>

<Location /b2bconsole>
      WebLogicPort 8002
      SetHandler weblogic-handler
      ErrorPage http://host.domain.com:port/error.html
</Location>

<Location /b2b>
      WebLogicPort 8002
      SetHandler weblogic-handler
       ErrorPage http://host.domain.com:port/error.html
</Location>

<Location /integration/worklistapp>
      WebLogicPort 8002
      SetHandler weblogic-handler
      ErrorPage http://host.domain.com:port/error.html
</Location>

<Location /integration/services>
      WebLogicPort 8002
      SetHandler weblogic-handler
      ErrorPage http://host.domain.com:port/error.html
</Location>

<Location /DefaultToDoTaskFlow>
      WebLogicPort 8002
      SetHandler weblogic-handler
      ErrorPage http://host.domain.com:port/error.html
</Location>

<Location /OracleBAM>
      WebLogicPort 9002
      SetHandler weblogic-handler
      ErrorPage http://host.domain.com:port/error.html
</Location>

<Location /OracleBAMWS>
      WebLogicPort 9002
      SetHandler weblogic-handler
      ErrorPage  http://host.domain.com:port/error.html
</Location>

<Location /sdpmessaging/userprefs-ui/>
      WebLogicPort 8002
      SetHandler weblogic-handler
      ErrorPage http://host.domain.com:port/error.html
</Location>
```

2. Start the Oracle WebLogic Servers as described in Section 5.6.3, "Configuring SOA Composite Applications for Two-Way SSL Communication."

5.6.5.2 Configuring Certificates for Oracle Client, Oracle HTTP Server, and Oracle WebLogic Server

To configure certificates for Oracle Client, Oracle HTTP Server, and Oracle WebLogic Server:

1. Export the user certificate from the Oracle HTTP Server wallet.

```
orapki wallet export -wallet . -cert cert.txt  -dn 'CN=\"Self-Signed
 Certificate for ohs1 \",OU=OAS,O=ORACLE,L=REDWOODSHORES,ST=CA,C=US'
```

2. Import the above certificate into the Oracle WebLogic Server trust store as a trusted certificate.

```
keytool -file cert.txt -importcert -trustcacerts -keystore DemoTrust.jks
```

3. Export the certificate from the Oracle WebLogic Server trust store.

```
keytool -keystore DemoTrust.jks -exportcert -alias wlscertgencab -rfc -file
certgencab.crt
```

4. Import the above certificate to the Oracle HTTP Server wallet as a trusted certificate.

```
orapki wallet add -wallet . -trusted_cert -cert certgencab.crt -auto_login_only
```

5. Restart Oracle HTTP Server.
6. Restart the Oracle WebLogic Servers as described in Section 5.6.3, "Configuring SOA Composite Applications for Two-Way SSL Communication."

5.6.6 Configuring SSL Between SOA Composite Application Instances and Oracle WebCache

The Test Web Service page in an Oracle WebCache and Oracle HTTP Server environment may require communication back through Oracle WebCache. Therefore, SSL must be configured between the SOA composite application instance and Oracle WebCache (that is, export the user certificate from the Oracle WebCache wallet and import it as a trusted certificate in the Oracle WebLogic Server trust store).

5.6.7 Using a Custom Trust Store for One-Way SSL During Design Time

To invoke a SOA composite application from another composite over HTTPS when using a custom trust store created with a tool such as `keytool` or `orapki`, perform the following actions in Oracle JDeveloper.

To use a custom trust store for one-way SSL during design time:

1. To fetch a WSDL file in the reference section, set the trust store information in **Tools** > **Preferences** > **Http Analyzer** > **HTTPS Setup** > **Client Trusted Certificate Keystore**.
2. During deployment to an SSL-enabled server, use the JSSE property at the command line:

```
jdev -J-Djavax.net.ssl.trustStore=your_trusted_location
```

5.6.8 Enabling an Asynchronous Process Deployed to an SSL-Enabled, Managed Server to Invoke Another Asynchronous Process Over HTTP

Assume you create the following environment:

- Asynchronous BPEL process A invokes asynchronous BPEL process B
- Asynchronous BPEL process A is deployed to a one-way SSL enabled, managed server
- All WSDL reference and bindings use plain HTTP

At runtime, the WSDL is looked for over HTTPS, and the callback message from asynchronous BPEL process B fails.

To resolve this issue, the `callbackServerURL` property must be passed at the reference binding level in the `composite.xml` file. This explicitly indicates the value of the callback URL for the given reference invocation. If the client composite is running in an SSL-managed server, then the callback defaults to SSL.

```
<reference name="Service1"
ui:wsdlLocation="http://localhost:8000/soa-infra/services/default/AsyncSecondB
PELMTOM/BPELProcess1.wsdl">
    <interface.wsdl
 interface="http://xmlns.oracle.com/Async/AsyncSecondBPELMTOM/BPELProcess1#wsdl
.interface(BPELProcess1)"
callbackInterface="http://xmlns.oracle.com/Async/AsyncSecondBPELMTOM/BPELProce
ss1#wsdl.interface(BPELProcess1Callback)"/>
    <binding.ws
port="http://xmlns.oracle.com/Async/AsyncSecondBPELMTOM/BPELProcess1#wsdl.endp
oint(bpelprocess1_client_ep/BPELProcess1_pt)"

location="http://localhost:8000/soa-infra/services/default/AsyncSecondBPELMTOM
/bpelprocess1_client_ep?WSDL">
         <wsp:PolicyReference URI="oracle/wss_username_token_client_policy"
                           orawsp:category="security"
orawsp:status="enabled"/>
      <wsp:PolicyReference URI="oracle/wsaddr_policy"
                           orawsp:category="addressing"
orawsp:status="enabled"/>
.
 <property name="callbackServerURL">http://localhost:8000/</property>
.
    </binding.ws>
.
  <callback>
      <binding.ws
port="http://xmlns.oracle.com/Async/AsyncSecondBPELMTOM/BPELProcess1#wsdl.endp
oint(bpelprocess1_client_ep/BPELProcess1Callback_pt)">
            <wsp:PolicyReference
URI="oracle/wss_username_token_service_policy"
                           orawsp:category="security"
orawsp:status="enabled"/>
 </binding.ws>
    </callback>

.
  </reference>
```

5.7 Configuring Security for Human Workflow WSDL Files

If the WSDL files for human workflow services are not exposed to external consumers, then set the flag that exposes the WSDL to `false` for each of the services:

```
<expose-wsdl>false</expose-wsdl>
```

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Web Services*.

6

Monitoring SOA Composite Applications

This chapter describes how to monitor recent instances and recent faults and rejected messages in SOA composite applications. It describes how to display details about BPEL process messages that require recovery and how to access the Recovery page of the BPEL process service engine to perform message recovery.

This chapter includes the following topic:

- Section 6.1, "Monitoring SOA Composite Application Recent Instances and Faults and Rejected Messages"

For more information, see Section 1.2.2, "Introduction to SOA Composite Applications."

6.1 Monitoring SOA Composite Application Recent Instances and Faults and Rejected Messages

You can monitor SOA composite application recent instances and faults and rejected messages from the SOA composite application Dashboard page. This page provides a high-level overview of the most recent state of the application.

To monitor SOA composite application recent instances and faults and rejected messages:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**.	1. Under **soa-infra**, expand the partition.
2. Select the **Deployed Composites** tab.	2. Select a specific SOA composite application.
3. In the **Composite** section, select a specific SOA composite application.	

2. Click **Dashboard** (if it is not selected).

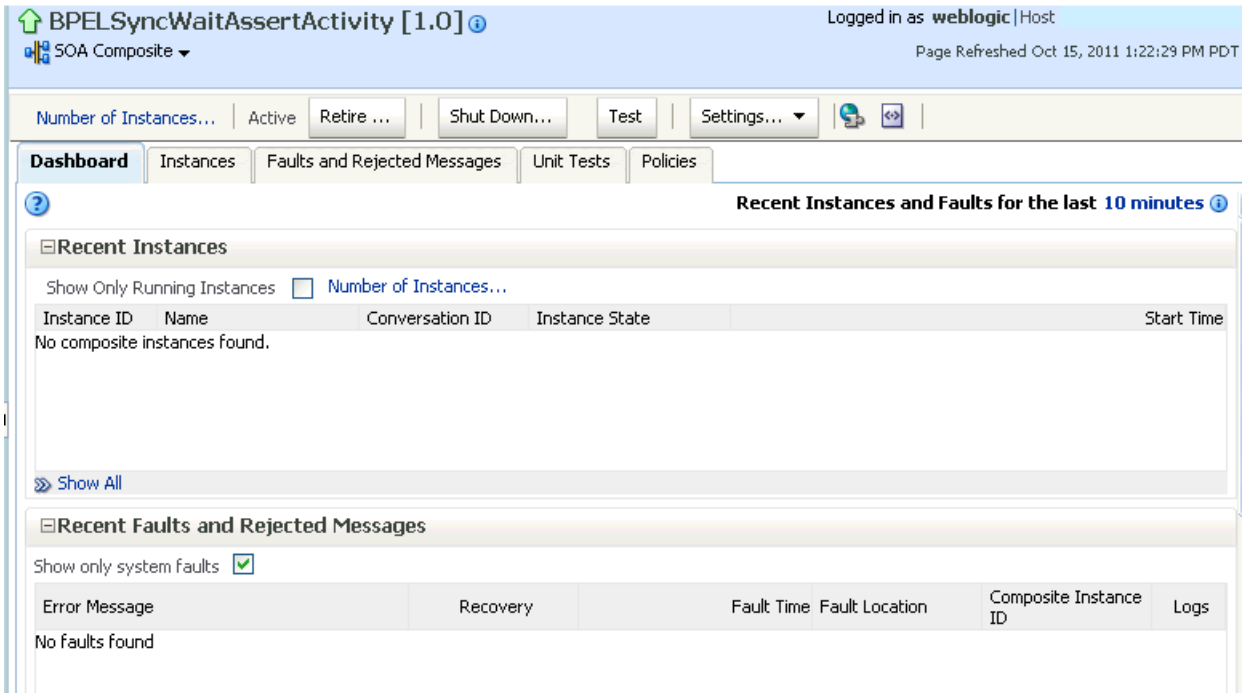
   The upper part of the Dashboard page displays the following details:

   - A summary of composite lifecycle states at the top of the Dashboard page, such as the number of running instances, total instances, and mode of the composite (active or retired).
   - A message indicating that the retrieval of recent instances and faults that are displayed on this page is restricted to the specified time period. This message is displayed if the **Restrict display of instances and faults to the last *time_period*** checkbox is selected on the SOA Infrastructure Common Properties

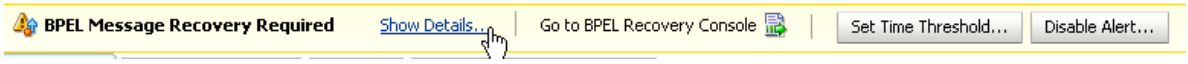
page (it is selected by default). The default time period value is 24 hours, but you can change this value. If this checkbox is not selected, all instances and faults (including count metrics) in the SOA Infrastructure since the last purging are displayed.

- Recent SOA composite application instances, including the instance ID, name, conversation ID, state (for example, faulted or completed), and start time.
- Recent faults and rejected messages, including the error message, whether you can recover from the fault, the time at which the fault occurred, the fault location (service, service component, or reference), the instance ID of the SOA composite application, and a link to log files describing the fault.

  If messages are awaiting recovery from the Recovery page of the BPEL process service engine, a message is displayed at the top of each SOA composite application home page.



3. If BPEL process messages require recovery from the Recovery page of the BPEL process service engine, the **BPEL Message Recovery Required** message is displayed at the top of each SOA Infrastructure home page.



The display of this message recovery information is controlled by the **bpelRecoveryStatus** and **excludeBpelMaxCreationTime** keys of the **AuditConfig** property in the System MBean Browser. This property is accessible from the **More SOA Infra Advanced Configuration Properties** section of the SOA Infrastructure Common Properties page. By default, the **bpelRecoveryStatus** key is set to **All** and the **excludeBpelMaxCreationTime** key is set to exclude the display of messages requiring recovery in the last five minutes.

a. Perform the following tasks:

Click This Link...	To...
Show Details	View a message showing the number of invoke, callback, and activity messages that require recovery from the Recovery page of the BPEL process service engine. Manual recovery is not required if automatic recovery of BPEL messages is enabled. In that case, only exhausted messages must be manually recovered. Click **Refresh** to recalculate the number of invoke, callback, and activity messages requiring recovery.
Go to BPEL Recovery Console	Access the Recovery page of the BPEL process service engine to perform message recovery. You can also access the Recovery page later by selecting **Service Engines** > **BPEL** from the **SOA Infrastructure** menu and clicking the **Recovery** tab in the resulting page. For more information, see Section 14.4, "Performing BPEL Process Service Engine Message Recovery."
Set Time Threshold	View a message showing the time period for excluding messages that require recovery. By default, this setting excludes the last five minutes of messages. This value is controlled by the **excludeBpelMaxCreationTime** key of the **AuditConfig** property on SOA Infrastructure Common Properties page. To change this value, click **Yes**.
Disable Alert	View a message enabling you to prevent the display of this message recovery information on the Dashboard page. If you click **Yes**, this message recovery information is not displayed. To display this information again on the Dashboard page, set the **bpelRecoveryStatus** key to **All** for the **AuditConfig** property in the **More SOA Infra Advanced Configuration Properties** section of the SOA Infrastructure Common Properties page. For more information, see Section 3.1, "Configuring SOA Infrastructure Properties."

4. In the **Recent Instances** section, perform the following tasks:

   a. Click the **Number of Instances** link to display a message showing the numbers of running and total instances for this SOA composite application.

      If you selected the **Restrict display of instances and faults to the last *time_period*** checkbox on the SOA Infrastructure Common Properties page and specified a time period or accepted the default value of 24 hours, the numbers of running and total instances for this SOA composite application for that time period are displayed. If you did not select this checkbox, all instances and faults for this SOA composite application since the last purging are displayed. Click **Recalculate** to recalculate the numbers.

   b. In the **Instance ID** column, click a specific instance ID to receive all instance details (flow trace and individual component audit trails) about the composite application. This displays the faults in the continuous context of a message flow from instance to instance.

      Note: If you disable the **Capture Composite Instance State** checkbox, the **Recent Instances** section does not show instances requiring fault recovery as running. However, these instances in need of recovery are still running and display in the **Recoverable** column of the **Component Metrics** section of this page, regardless of whether the instances state is captured or not.

c. Click **Show All** below the section to access the Instances page of the SOA composite application.

5. In the **Recent Faults and Rejected Messages** section, perform the following tasks:

   a. In the **Error Message** column, click an error message to display complete information about the fault. If the fault is identified as recoverable, click the **Recover Now** link to perform fault recovery.

   b. In the **Recovery** column, if a fault is identified as recoverable, click **Recover** to perform fault recovery at the component instance level.

   c. In the **Fault Location** column, click a specific location to access the Dashboard page for the service, service component, or reference.

   d. In the **Composite Instance ID** column, click a composite instance ID to access the flow trace of the message that contains the fault. This displays the faults in the continuous context of a message flow from instance to instance.

   e. In the **Logs** column, click a specific log to access the Log Messages page filtered for the specific faulted instance.

   f. Click **Show All** below the section to access the Faults and Rejected Messages page of the SOA composite application.

   The lower part of the Dashboard page displays the following details:

   - The name and type of service components used in this SOA composite application, the number of running and total instances, and the number of recoverable and nonrecoverable faulted instances for each service component.
   - The name and type of service (inbound) and reference (outbound) binding components used in this SOA composite application, the number of binding component faults, the total messages processed, and the average message processing time.

Component Metrics

Name	Component Type	Total Instances	Running Instances	Faulted Instances	
				Recoverable	Non Recoverable
Mediator1	Mediator	2	0	0	0
POProcessingInitiatorE	BPEL	2	0	0	0

Services and References

Name	Type	Usage	Faults	Total Messages	Average Processing Time (sec)
poprocessinginitiatorbpel_client_ep	Web Service	Service	10	8	0.111
Service1	Web Service	Reference	0	2	0.027
POApprovalFileOut	JCA Adapter	Reference	0	2	0.005

The **Faulted Instances** columns of the **Component Metrics** section count faults that are recoverable and nonrecoverable. Component instances associated with a recoverable fault are not considered faulted. These instances are considered to be running because they have not reached the end of the lifecycle. These instances can be recovered through a recovery option such as retry, rethrow, abort, and so on. This fault count can differ from the **Recent Instances** section of this page and the Faults and Rejected Messages page, which list faults without making a distinction between recoverable and nonrecoverable.

6. In the **Name** column of the **Component Metrics** section, click a service component. This displays its home page for viewing specific details about instances, faults, and policies.
7. In the **Name** column of the **Services and References** section, click a service or reference. This displays its home page for viewing specific details about instances, faults, policies, rejected messages, and message header configuration properties.

> **Note:** You can also go to the Instances page and the Faults and Rejected Messages page of the SOA Infrastructure to monitor instances and faults across all deployed composites, respectively. From there, you can click a specific composite for additional details.

For more information, see the following sections:

- Section 1.2.3, "Introduction to SOA Composite Application Instances"
- Section 1.2.4, "Introduction to Service Components and Service Component Instances"
- Section 1.2.5, "Introduction to Binding Components"
- Section 8.4, "Recovering from SOA Composite Application Faults at the SOA Infrastructure Level"
- Section 8.5, "Recovering from SOA Composite Application Faults in the Application Home Page"
- *Oracle Fusion Middleware Administrator's Guide* for details about viewing and searching log files

7

Deploying and Managing SOA Composite Applications

This chapter describes how to deploy and manage SOA composite applications, including managing the states of deployed composites; deploying, redeploying, and undeploying a SOA composite application from Oracle Enterprise Manager Fusion Middleware Control; automating testing of SOA composite applications; managing policies; exporting deployed composites; grouping composites into partitions; and managing BPEL and BPMN monitors.

This chapter includes the following topics:

- Section 7.1, "Deploying SOA Composite Applications"
- Section 7.2, "Redeploying SOA Composite Applications"
- Section 7.3, "Undeploying SOA Composite Applications"
- Section 7.4, "Managing the State of Deployed SOA Composite Applications"
- Section 7.5, "Automating the Testing of SOA Composite Applications"
- Section 7.6, "Managing SOA Composite Application Policies"
- Section 7.7, "Exporting a Deployed SOA Composite Application"
- Section 7.8, "Grouping SOA Composite Applications into Partitions"
- Section 7.9, "Disabling and Enabling BPEL and BPMN Business Monitors"

For information on the following:

- Creating SOA composite application archives and configuration plans in which you define the URLs and property values to use for test, development, and production environments, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*
- Deploying with `ant` scripts, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*
- Deploying with Oracle WebLogic Scripting Tool (WLST), see *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

> **Note:** If Oracle Enterprise Manager Fusion Middleware Control is run in a single sign-on (SSO)-enabled environment, you are again prompted to enter the user name and password credentials as part of the last step of the Deploy SOA Composite, Undeploy SOA Composite, and Redeploy SOA Composite wizards. This information is only requested once per Oracle Enterprise Manager Fusion Middleware Control session.

7.1 Deploying SOA Composite Applications

You can deploy SOA composite applications from Oracle Enterprise Manager Fusion Middleware Control with the Deploy SOA Composite wizard. You must first create a deployable archive in Oracle JDeveloper or with the `ant` or WLST command line tool. Use the Deploy SOA Composite wizard to deploy any of the following:

- A new SOA composite application for the first time.
- A new revision (for example, 2.0) alongside an older revision (for example, 1.0) without having an impact on the latter. The revision deployed last becomes the new default revision of that composite (unless you specify otherwise at a later step during deployment).
- A SOA bundle (ZIP file) containing multiple revisions (for example, revisions 2.0, 3.0, and 4.0) of a SOA composite application that has different revisions currently deployed (for example, 1.0). This option enables you to deploy revisions 1.0, 2.0, 3.0, and 4.0 at the same time. The bundle can also contain revisions of different composites. There is no restriction that all revisions must be of the same composite application. There should not be any cross references between the composites in the same bundle. For example, composite A revision 1.0 should not reference composite B revision 1.0.

Deployment extracts and activates the composite application in the SOA Infrastructure. After an application is deployed, you can perform administration tasks, such as creating instances, configuring properties, monitoring performance, managing instances, and managing policies and faults.

> **Note:** If you want to redeploy an *existing* revision of an application, do *not* use this wizard. Instead, use the Redeploy SOA Composite wizard.

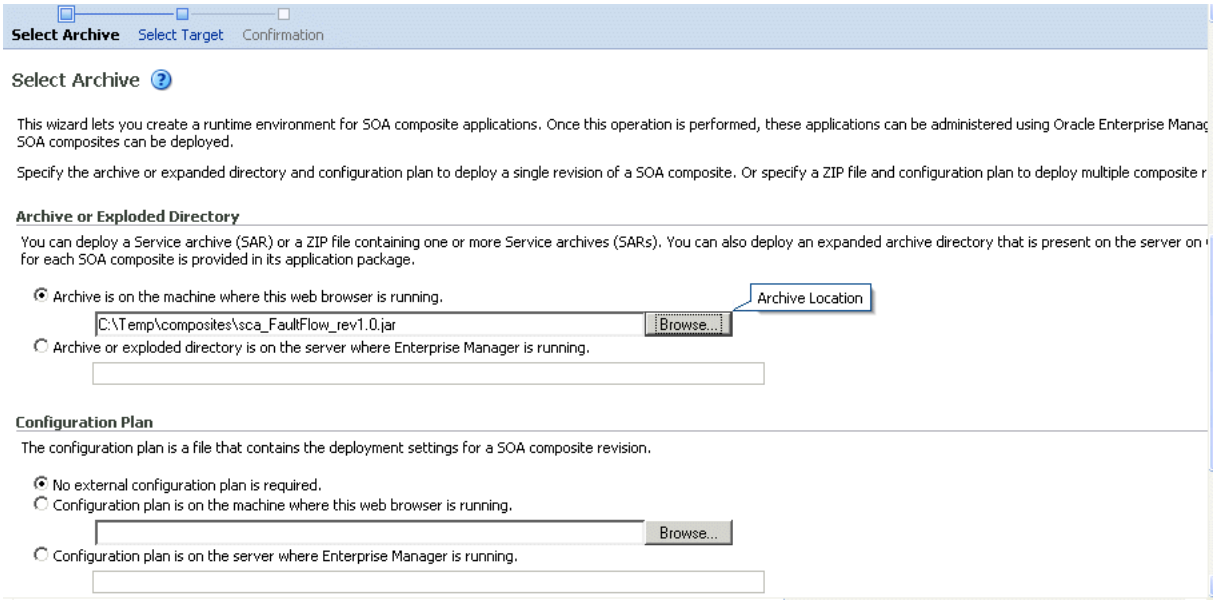
To deploy SOA composite applications:

1. Access the Deploy SOA Composite wizard through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...	From the SOA Infrastructure Home Page...	From the SOA Composite Menu...
1. Select **SOA Deployment** > **Deploy**.	1. Right-click **soa-infra**. 2. Select **SOA Deployment** > **Deploy**.	1. Click the **Deployed Composites** tab. 2. Above the **Composite** table, click **Deploy**.	1. Select **SOA Deployment** > **Deploy Another Composite.**

> **Note:** You can also access the Deploy SOA Composite wizard by selecting **Deploy To This Partition** from the **Deployment** dropdown list on the Manage Partitions page or home page of a specific partition, from the **SOA Partition** menu at the top of the home page of a specific partition, or by right-clicking a specific partition in the navigator.

The Select Archive page appears.



2. In the **Archive or Exploded Directory** section, specify the archive of the SOA composite application to deploy. The archive contains the project files of the composite to be deployed (for example, **HelloWorld_rev1.0.jar** for a single archive or **OrderBooking_rev1.0.zip** for multiple archives). This information is required.

3. In the **Configuration Plan** section, optionally specify the configuration plan to include with the archive. The configuration plan enables you to define the URL and property values to use in different environments. During process deployment, the configuration plan is used to search the SOA project for values that must be replaced to adapt the project to the next target environment.

4. Click **Next**.

   The Select Target page appears.

   This page lists the Oracle SOA Suite managed server or cluster to which to deploy the SOA composite application archive.

5. Select the partition into which to deploy this SOA composite application. Partitions enable you to logically group SOA composite applications into separate sections. Even if there is only one partition available, you must explicitly select it. Once deployed, a composite cannot be transferred to a different partition.

   If you want to deploy a SOA composite application to a partition that does not exist, exit the wizard and create the partition *before* deploying the composite. You create partitions in the Manage Partitions page, accessible from the **SOA Infrastructure** menu.

   If the server contains no partitions, you cannot deploy composite applications to that server. Also, if the server is not in a *running* state, you cannot deploy this

archive. By default, a partition named **default** is automatically included with Oracle SOA Suite. You can delete the default partition.

> **Note:** Human workflow artifacts such as task mapped attributes (previously known as flex field mappings) and rules (such as vacation rules) are defined based on the namespace of the task definition. Therefore, the following issues are true when the same SOA composite application with a human workflow task is deployed into multiple partitions:
>
> - For the same task definition type, mapped attributes defined in one partition are visible in another partition.
> - Rules defined on a task definition in one partition can apply to the same definition in another partition.

If you invoke the Deploy SOA Composite wizard by selecting **Deploy To This Partition** from the **Deployment** dropdown list on the Manage Partitions page or home page of a specific partition, the partition to which to deploy is selected. Therefore, the Select Target page is skipped.

6. Click **Next**.

   The Confirmation page appears.

7. Review your selections.
8. Select whether to deploy the SOA composite application as the default revision. The default revision is instantiated when a new request comes in.
9. Click **Deploy**.

   Processing messages are displayed.

   At this point, the deployment operation cannot be canceled. Deployment continues even if the browser page is closed.

10. When deployment has completed, the home page of the newly deployed composite revision is displayed automatically. A confirmation message at the top of the page tells you that the composite has been successfully deployed. In the case of a bundle deployment, the Deployed Composites page of the SOA Infrastructure is displayed.

For information about creating configuration plans and deploying applications from Oracle JDeveloper, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

7.1.1 Deploying SOA Composite Applications with Task Flows

When you deploy a SOA composite application with a task flow Enterprise Resource Archive (EAR) file from Oracle Enterprise Manager Fusion Middleware Control or Oracle WebLogic Server Administration Console to a multiple partition environment, you cannot specify partition details. To specify a partition, modify the `hwtaskflow.xml` file to include the partition name in the generated EAR file (the project version of the file remains unchanged). This file is located under the `TaskForm` project `adfmsrc` directory (for example, `HelpDeskRequestTaskFlow\adfmsrc\hwtaskflow.xml`).

```
<hwTaskFlows
 xmlns="http://xmlns.oracle.com/bpel/workflow/hwTaskFlowProperties">
   <ApplicationName>worklist</ApplicationName>
```

```
<LookupType>LOCAL</LookupType>
<TaskFlowDeploy>false</TaskFlowDeploy>
<PartitionName>partition2</PartitionName>
```

7.1.2 Deploying SOA Composite Applications with ant Scripts and the WLST Command Line Tool

You can also deploy SOA composite applications with `ant` scripts and the WLST command line tool.

- For information on deploying with `ant` scripts, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*
- For information on deploying with WLST, see *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

7.2 Redeploying SOA Composite Applications

You can redeploy SOA composite applications from Oracle Enterprise Manager Fusion Middleware Control with the Redeploy SOA Composite wizard. Using the Redeploy SOA Composite wizard has the following consequences:

- A new version of a revision of a currently deployed SOA composite application is redeployed on the same deployment target (for example, old version 1.0 is redeployed as new version 1.0).
- If the older, currently deployed version of this revision has running instances, you can select whether to change the state of those instances to stale. See Step 6 for a description of the **Running Instances** section of the Redeploy SOA Composite wizard and limitations on this option.

  The instance state is available in the instance listing, and you can access audit and flow trace details.

Notes:

- If you want to maintain multiple revisions of a deployed application (for example, revisions 1.0 and 2.0), do *not* use this wizard. Instead, use the Deploy SOA Composite wizard.
- Redeploying multiple SOA composite applications at once is not supported.

To redeploy applications:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...	From the SOA Infrastructure Home Page...	From the SOA Composite Menu...
1. Select **SOA Deployment** >**Redeploy**. The Select Composite page appears. 2. In the **SOA Composite Deployments** section, select the SOA composite application revision you want to redeploy, and click **Next**.	1. Right-click **soa-infra**. 2. Select **SOA Deployment** > **Redeploy**. The Select Composite page appears. 3. In the **SOA Composite Deployments** section, select the SOA composite application revision you want to redeploy, and click **Next**.	1. Click the **Deployed Composites** tab. 2. In the **Composite** table, select a specific SOA composite application. Only one application can be redeployed at a time. 3. Above the **Composite** table, click **Redeploy**.	1. Select **SOA Deployment** > **Redeploy.**

Note: You can also access the Redeploy SOA Composite wizard by right-clicking a partition and selecting **SOA Deployment** > **Redeploy**.

The Select Archive page appears.

2. In the **Archive or Exploded Directory** section, select the location of the SOA composite application revision you want to redeploy.
3. In the **Configuration Plan** section, optionally specify the configuration plan to include with the archive.
4. Click **Next**.

   The Confirmation page appears.
5. In the **Default Revision** section, select whether to redeploy the SOA composite application as the default revision.
6. In the **Running Instances** section, select whether to continue running the current instances of a redeployed SOA composite application.
   - **Change states of running instances to stale**:
     - Select to change the states of currently running instances to stale after redeployment of the SOA composite application.

     Note: Even if the state of a SOA composite application instance becomes stale, the instance states of BPEL process service components included in this composite remain as completed and not as stale. This is done for performance reasons, and is the expected behavior.
   - **Continue instances on redeploy (current instance states will not be changed)**:

Note: This option is displayed if Oracle BPM Suite is installed in the SOA Infrastructure, and only supported for the deployment of BPM composites. Do not select this option if you are deploying:

- A SOA composite application from a SOA Infrastructure environment in which Oracle BPM Suite is also installed.
- A BPM composite that includes a durable BPEL process, regardless of whether that process has been modified. Durable BPEL processes are those that take time to complete execution. Examples of durable BPEL processes are asynchronous processes (which are always durable) and synchronous processes that include a durable activity such as a wait activity.

  If you select this option and attempt to redeploy a durable BPEL process, then deployment fails.

– Select to continue running instances after redeployment of the BPM composite application. This prevents these instance states from being changed to stale.

  Instances of different service components behave differently after redeployment. Ensure that you understand the following details:

For...	Description
Oracle Business Process Management Notation (BPMN) instances	You must manually migrate instances. BPMN service component instances are displayed as running in Oracle Enterprise Manager Fusion Middleware Control after redeployment. However, to ensure that your redeployed application is running correctly, search for instances with the pending migration state in Oracle BPM WorkSpace and manually migrate these instances to the new component definition.

For...	Description
Oracle BPEL process, decision service (business rule), Oracle Mediator, and human workflow instances	Instances are automatically migrated. Oracle Enterprise Manager Fusion Middleware Control attempts to automatically migrate the instances of these components. If the migration is successful, your SOA composite application is redeployed. However, if some of the following component instances cannot be migrated, redeployment of the entire composite fails. ■ BPEL instances: This capability is not supported if the composite includes a BPEL process service component that has changed from the previous revision. ■ Oracle Mediator: Checks the new component definition for compatibility. If it is incompatible with the existing component definition, the redeployment of the composite revision fails. However, if at the time of redeployment all Oracle Mediator processes within the running instances of this composite have completed, the definition change has no impact and redeployment is successful. Because Oracle Mediator instances take a short time to complete, their migration does not pose a high risk of redeployment failure. ■ Decision service: If the component definition has changed and is incompatible with the existing component definition, this may cause the redeployment of the composite revision to fail. However, if at the time of redeployment all decision service processes within the running instances of this composite have completed, the definition change has no impact and redeployment is successful. Because decision service instances take a short time to complete, their migration does not pose a high risk of redeployment failure. ■ Human workflow: Changes in human workflow do not impact any existing user task that has been created. Newly created human tasks after redeployment can use a new routing slip.

7. Click **Redeploy**.

   Processing messages are displayed.

   At this point, the deployment operation cannot be canceled. Deployment continues even if the browser page is closed.

8. When redeployment has completed, click **Close**.

   When redeployment has completed, the home page of the newly redeployed composite revision is displayed. A confirmation message at the top of the page tells you that the composite has been successfully redeployed.

7.3 Undeploying SOA Composite Applications

You can undeploy SOA composite applications from Oracle Enterprise Manager Fusion Middleware Control with the Undeploy SOA Composite wizard. Using the Undeploy SOA Composite wizard has the following consequences:

- You can no longer configure and monitor this revision of the application.
- You can no longer process instances of this revision of the application.
- The state of currently running instances is changed to stale and no new messages sent to this composite are processed.
- The instance state of the undeployed composite application is set to stale. The instance state is available in the instance listing, and you can access audit trail and flow trace details.

- If you undeploy the default revision of the SOA composite application (for example, 2.0), the next active, available revision of the application is automatically designated as the new default (for example, 1.0).
- A warning message is displayed at the end of this wizard when you undeploy the default composite revision.

  If no active revision is available and the default revision is undeployed, your composite may be unable to process new incoming requests. It is recommended that you have at least one active revision of this composite deployed before you undeploy the default revision.

  If you undeploy this revision and no active revisions of this composite are found, a retired revision is automatically designated as the new default revision. A warning message is displayed after this wizard closes. Although all currently executing instances complete normally in retired composites, they cannot process any incoming requests. To process new incoming requests for this composite after the current default revision is undeployed, you must deploy a new revision or reactivate a previously retired revision.

> **Note:** If you want to undeploy and then redeploy an existing revision of this application, do *not* use this wizard. Instead, use the Redeploy SOA Composite wizard. The Redeploy SOA Composite wizard enables you to redeploy an existing revision of a SOA composite application and remove (overwrite) the older, currently deployed version of the revision.

To undeploy applications:

> **Note:** You can undeploy multiple SOA composite applications together if they are located in the same partition. For information, see Section 7.8, "Grouping SOA Composite Applications into Partitions."

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...	From the SOA Infrastructure Home Page...	From the SOA Composite Menu...
1. Select **SOA Deployment** >**Undeploy**. The Select Composite page appears. 2. In the **SOA Composite Deployments** section, select a specific SOA composite application to undeploy, and click **Next**.	1. Right-click **soa-infra**. 2. Select **SOA Deployment** > **Undeploy**. The Select Composite page appears. 3. In the **SOA Composite Deployments** section, select a specific SOA composite application to undeploy, and click **Next**.	1. Click the **Deployed Composites** tab. 2. In the **Composite** table, select a specific SOA composite application. Only one application can be undeployed at a time. 3. Above the **Composite** table, click **Undeploy**.	1. Select **SOA Deployment** > **Undeploy.**

> **Note:** You can also access the Undeploy SOA Composite wizard through these additional partition options:
>
> - Right-clicking a partition and selecting **SOA Deployment** > **Undeploy All From This Partition**
> - Selecting **Deployment** > **Undeploy All From This Partition** on the partition home page
> - Selecting **Deployment** > **Undeploy All From This Partition** for the selected partition from the Manage Partitions page

The Confirmation page appears.

2. If you are satisfied, click **Undeploy**. You are warned if you are about to undeploy the last remaining revision of a deployed composite application.

   Processing messages are displayed.

   At this point, the undeploy operation cannot be canceled. Undeployment continues even if the browser page is closed.

3. When undeployment has completed, the SOA Infrastructure Deployed Composites page is displayed automatically. A confirmation message at the top of the page tells you that the composite has been successfully undeployed.

> **Note:** When a partition is deleted, all SOA composite applications in it are automatically undeployed. A message is displayed indicating that all the applications in that partition are to be undeployed.

7.4 Managing the State of Deployed SOA Composite Applications

You can manage the lifecycle state of deployed SOA composite applications from either of two pages:

- From the Deployed Composites page of the SOA Infrastructure, which lists all SOA composite applications deployed to the SOA Infrastructure
- From the application home page of a specific SOA composite application (all tabs)

The management tasks that you can perform are based on the page you are on. Table 7–1 provides details.

Table 7–1 Application State Actions

Action	Perform on the Deployed Composites Page of the SOA Infrastructure?	Perform on the Application Home Page (All Tabs)?
Shut Down and **Start Up**	Yes	Yes
Retire and **Activate**	Yes	Yes

Table 7–1 (Cont.) Application State Actions

Action	Perform on the Deployed Composites Page of the SOA Infrastructure?	Perform on the Application Home Page (All Tabs)?
Set as Default	Yes	▪ No: If only one version of the composite application is set as the default. ▪ Yes: If there are multiple versions of the same composite application, this option is visible for all other versions of the same composite expect the one that is the default.
Deploy	Yes	Yes (through the **Composite** menu by selecting **SOA Deployment** > **Deploy Another Composite**)
Undeploy	Yes	Yes (through the **Composite** menu by selecting **SOA Deployment** > **Undeploy**)
Redeploy	Yes	Yes (through the **Composite** menu by selecting **SOA Deployment** > **Redeploy**)
Test	No	Yes
Composite Audit Level	No	Yes
Payload Validation	No	Yes
Enable/Disable Business Monitoring	No	Yes
Show WSDL and Endpoint URI (icon)	No	Yes
Show XML Definition (icon)	No	Yes

See the following section based on the action you want to perform:

- Section 7.4.1, "Managing the State of All Applications at the SOA Infrastructure Level"
- Section 7.4.2, "Managing the State of an Application from the SOA Composite Application Home Page"

For more information, see Section 1.2.2, "Introduction to SOA Composite Applications."

7.4.1 Managing the State of All Applications at the SOA Infrastructure Level

You can manage the state of *all* SOA composite applications from the Deployed Composites page at the SOA Infrastructure level.

To manage the state of all applications at the SOA Infrastructure level:

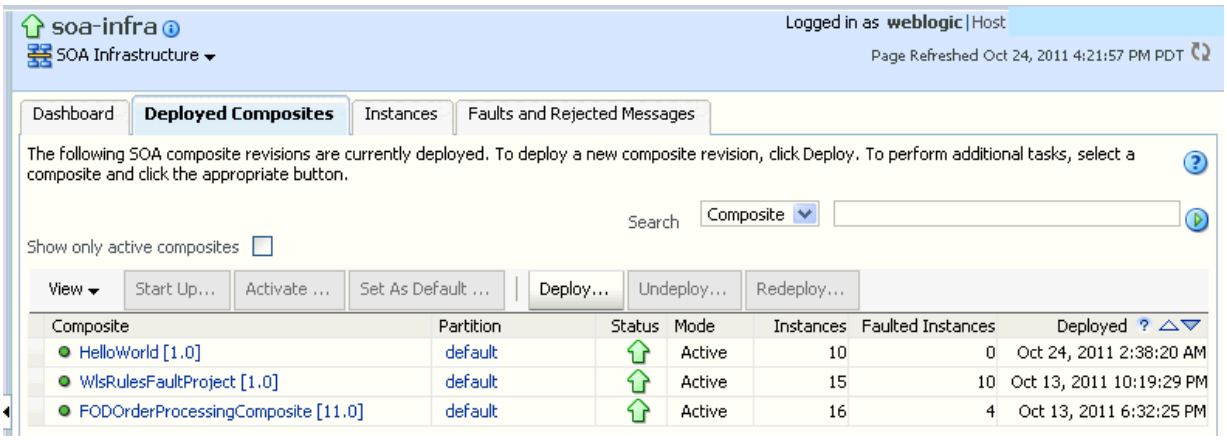
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...	From the SOA Composite Menu...
1. Select **Home**.	1. Click **soa-infra**.	1. Select **SOA Infrastructure**.

2. Click the **Deployed Composites** tab.

   The Deployed Composites page displays the following details:

   - A utility for searching for a specific SOA composite application by specifying a full or partial composite name and clicking **Search**. You can also search for SOA composite applications by partition.
   - A list of all SOA composite applications deployed in the SOA Infrastructure, including the partition in which they are deployed, current mode (active or retired), number of instances, number of faulted instances, and last modification date (deployment time, redeployment time, or any composite configuration change). The green dot to the left of the composite name indicates that this is the default revision of the application.



> **Note:** To always see the latest details about deployed SOA composite applications, click the **Refresh** icon in the upper right corner or navigate away from this page and return to it.

3. Click **Deploy** to deploy a new application. For all other options listed above the **Composite** section, first select the composite application by clicking the column to the left of the name, then select a specific option to perform.

   The following table describes the available options:

Action	Description
Shut Down	Shuts down a running SOA composite application revision. Any request (initiating or a callback) to the composite is rejected if the composite is shut down. New incoming requests cannot be processed. All existing instances are allowed to complete as usual (the same as when a composite is retired). **Note:** The behavior differs based on which binding component is used. For example, if it is a web service request, it is rejected back to the caller. A JCA adapter binding component may do something else in this case (for example, put the request in a rejected table). This option is displayed when the composite application has been started.
Start Up	Restarts a composite application revision that was shut down. This action enables new requests to be processed (and not be rejected). No recovery of messages occurs. This option is displayed when the composite application has been stopped.
Retire	Retires the selected composite revision. If the process lifecycle is retired, you cannot create a new instance. Existing instances are allowed to complete normally. An initiating request to the composite application is rejected back to the client. The behavior of different binding components during rejection is as described for the shut down option. A callback to an initiated composite application instance is delivered properly. This option is displayed when the composite application is active. Note the following details when you attempt to retire the default composite revision, or have already retired a default composite revision. A warning page is also displayed with these details. ■ When you attempt to retire the default composite revision, if another active revision of the composite is found, it is designated as the new default revision. If there are multiple active revisions, the active composite that was most recently the default revision (based on the time stamp) is designated as the default revision. If you then re-activate the retired revision, it does not automatically become the default revision again. You must explicitly make it the default revision again. ■ If you retire the default composite revision and no active revision of this composite is found, a new default revision is not designated and a warning message is displayed. The retired revision remains the default revision. However, this composite can no longer process any incoming requests. To process new incoming requests for this composite, you must deploy a new revision or re-activate one of the previously retired revisions.
Activate	Activates the retired composite application revision. Note the following behavior with this option: ■ All composite applications are automatically active when deployed. ■ Other revisions of a newly deployed composite application remain active (that is, they are not automatically retired). If you want, you must explicitly retire them. This option is displayed when the application is retired.

Action	Description
Set As Default	Sets the selected composite application revision to be the default. Default revisions are indicated by a green dot in the **Composite** table. If a new request comes in for a specific composite application revision, that composite application revision is invoked. If a new request comes in without specifying a revision, the default revision is invoked. The default revision can change when a composite application is retired. The change is based on whether there is another active revision of the composite. For details, see the description for the Retire action in this table. The default revision is changed automatically when a default composite application revision is undeployed. The default composite revision also changes automatically when you redeploy a composite application. The newly redeployed revision automatically becomes the default revision, unless at the time of redeployment, you specify to keep the previous default revision unchanged. For details, see the description of the Undeploy action in this table. Inbound adapters are activated only on the default revision.
Deploy	Deploys a revision. Deployment activates the composite application in the SOA Infrastructure. Use this selection when you want to deploy: ▪ A new SOA composite application for the first time. ▪ A new revision (for example, 2.0) of a SOA composite application that has a different revision that is currently deployed (for example, 1.0). This option enables both revisions 1.0 and 2.0 to be deployed at the same time. If you specify a revision that exists, you receive an error. You must change this revision outside of the Deploy SOA Composite wizard. For more information, see Section 7.1, "Deploying SOA Composite Applications" and Section 7.8, "Grouping SOA Composite Applications into Partitions."

Action	Description
Undeploy	Undeploys the selected composite application revision. The consequences of this action are as follows: ▪ You can no longer configure and monitor this revision of the composite application. ▪ You can no longer process instances of this revision of the composite application. ▪ You cannot view previously completed processes. ▪ The state of currently running instances is changed to stale and no new messages sent to this composite application are processed. ▪ If you undeploy the default revision of the composite application (for example, 2.0), the next available, active revision of the composite application becomes the default (for example, 1.0). If no active revision is available and the old default revision is undeployed, your composite may be unable to process new incoming requests. It is recommended that you have at least one active revision of this composite deployed before you undeploy the default revision. If you undeploy the default revision and no active revisions of this composite are found, a retired revision is automatically designated as the new default revision. A warning message is displayed after this wizard closes. Although all currently executing instances complete normally in retired composites, they cannot process any incoming requests. To process new incoming requests for this composite after the current default revision is undeployed, you must deploy a new revision or reactivate a previously retired revision. **Note:** Undeploying multiple SOA composite applications at the same time is supported if they are in the same partition. For more information, see Section 7.3, "Undeploying SOA Composite Applications" and Section 7.8, "Grouping SOA Composite Applications into Partitions."
Redeploy	Redeploys an existing revision of a SOA composite application. The consequences of this action are as follows: ▪ A new version of a revision of a currently deployed SOA composite application is redeployed (for example, old version 1.0 is redeployed as new version 1.0). ▪ The older, currently deployed version of this revision is removed (overwritten). ▪ If the older, currently deployed version of this revision has running instances, you can select whether to change the state of those instances to stale. For more information, see Section 7.2, "Redeploying SOA Composite Applications."

For more information, see Section 1.4.3.3, "Introduction to the Lifecycle State of SOA Composite Applications."

7.4.2 Managing the State of an Application from the SOA Composite Application Home Page

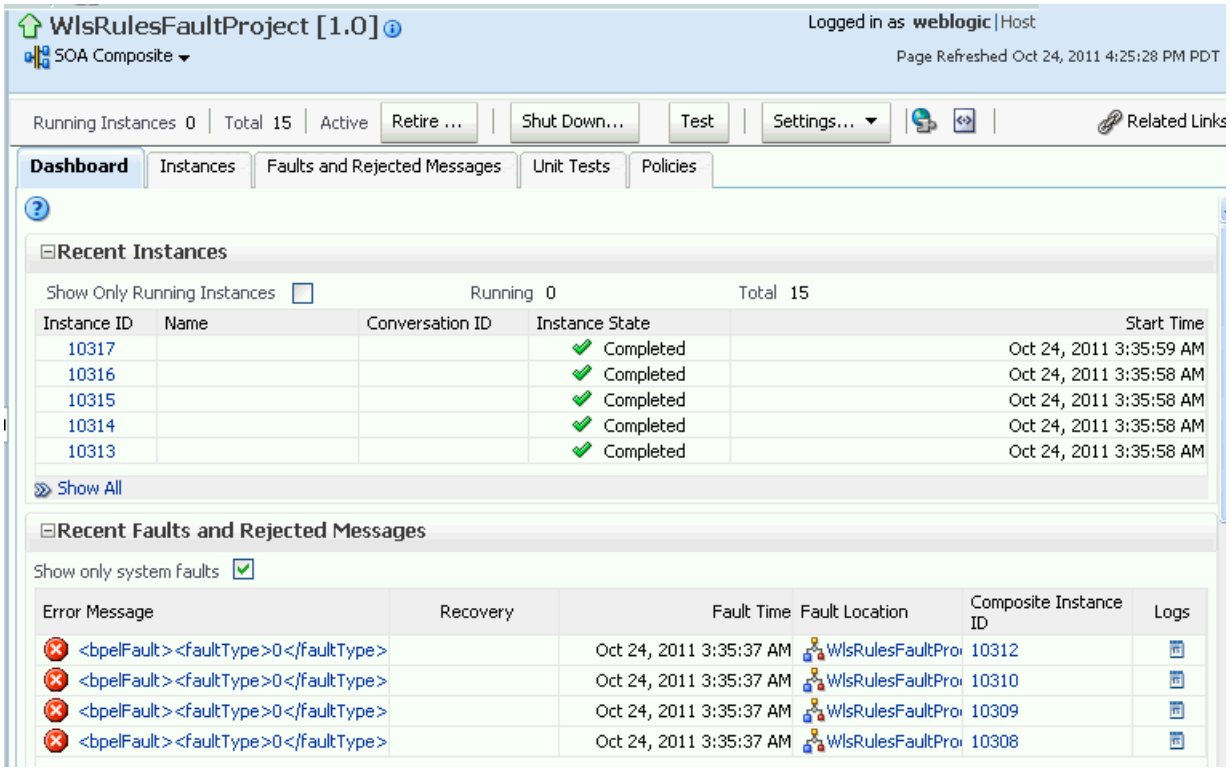
You can manage the state of an individual SOA composite application from the application's home page.

To manage the state of an application from the SOA composite application home page:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**.	1. Under **soa-infra**, expand the partition.
2. Select the **Deployed Composites** tab.	2. Select a specific SOA composite application.
3. Select a specific SOA composite application.	

The Dashboard page of the selected SOA composite application is displayed.



Notes:

- The **Total** field of the **Recent Instances** section sometimes does not display the correct number of total instances despite instances having completed successfully. In these cases, click the **Refresh** icon in the upper right corner to view the actual number of total instances.
- When the **Capture Composite Instance State** checkbox is enabled on the SOA Infrastructure Common Properties page, created instances are displayed immediately even if you have defined a constraint that appears to prevent an instance from being displayed immediately (for example, you have defined a flush delay of 10 minutes or specified a batch size of 100 records to write to a database). This is because instance tracking is moved to the immediate mode since the state of the composites must be captured.
- After the SOA Infrastructure is started, it may not be completely initialized to administer incoming requests until all deployed composites are loaded. During SOA Infrastructure initialization, a warning message is displayed at the top of the SOA composite application home page. Do not perform operations such as composite deployment, composite undeployment, and others while this message is displayed. For more information, see Section 3.2.1, "Waiting for SOA Infrastructure Startup Initialization to Complete."

2. From the list of options at the top of the page, select a specific action to perform. These options are also displayed at the top of the Instances, Faults and Rejected Messages, Unit Tests, and Policies pages of the SOA composite application.

Action	Description
Shut Down	See the table under Step 3 of Section 7.4.1, "Managing the State of All Applications at the SOA Infrastructure Level" for a description of this option.
Start Up	See the table under Step 3 of Section 7.4.1, "Managing the State of All Applications at the SOA Infrastructure Level" for a description of this option.
Retire	See the table under Step 3 of Section 7.4.1, "Managing the State of All Applications at the SOA Infrastructure Level" for a description of this option.
Activate	See the table under Step 3 of Section 7.4.1, "Managing the State of All Applications at the SOA Infrastructure Level" for a description of this option.
Test	Enables you to initiate a test instance from the Test Web Service page. **Note:** This button is disabled when the SOA composite application is stopped or retired. This is because you cannot create an instance for a stopped or retired application. This button is also disabled when there are no web services available for the application. Only composite applications having services with web service bindings can be tested from this page. For more information, see Section 8.1, "Initiating a SOA Composite Application Test Instance."

Action	Description
Settings: Composite Audit Level	Sets the level of audit tracking to perform at the SOA composite application level. This setting *can* override the audit level defined at the SOA Infrastructure level. By default, the value is **Inherit**, which does not override the SOA Infrastructure level setting. If you select to set the audit tracking level, the following options are available: ■ **Inherit**: Logging matches the SOA Infrastructure audit level that you set on the SOA Infrastructure Common Properties page. This is the default setting. ■ **Production**: Minimal information for SOA composite application instances is collected. For example, the BPEL process and Oracle Mediator service engines do not capture the payload. Therefore, the payload details are not available in the flow audit trails. The BPEL process service engine collects payload details for all activities except assign activities. This level is optimal for most standard operations and testing. ■ **Development**: Complete information for SOA composite application instances is collected. This option allows both composite instance tracking and payload tracking. This setting may have an impact on performance because the payload is stored at each step in the message flow. This setting is useful for debugging purposes. ■ **Off**: No logging is performed. Composite instance tracking information and payload tracking information are not collected. Setting audit level tracking at the SOA composite application level overrides the same tracking set at the SOA Infrastructure level. By default, the settings are the same at the SOA composite application and SOA Infrastructure levels. SOA composite application settings are automatically changed when the global SOA Infrastructure settings are changed. By choosing any other setting at the SOA composite application level, you are overriding the inherited settings. One form of overriding is when you explicitly select the same local composite value that happens to be the current global value. If the SOA Infrastructure setting is then changed, this specific composite application does not inherit the new value. For example, assume the SOA Infrastructure setting is **Off**. Therefore, all composite applications have their audit tracking set to **Off**. Then, you explicitly set composite application XYZ to **Off**. Then, go to the SOA Infrastructure and change the setting to **Production**. The tracking levels for all composite applications are now **Production**; except for XYZ, which is still set to **Off**. Note the following impact of instance tracking changes on message flows that span several SOA composite applications (for example, a composite application invoking another composite application through a reference binding component or an event published in one composite application and subscribed to in another composite application). ■ If an intermediate composite application has disabled instance tracking, then a single message flow across multiple composite application instances appears as separate, unconnected flows. For example, assume a message flows through composite applications C1, C2, and C3. C1 and C3 have enabled instance tracking, while C2 has disabled it. Two separate flows for C1 and C3 are displayed in Oracle Enterprise Manager Fusion Middleware Control. ■ Sources or targets of events or messages may not be displayed. For example, assume you have two composite applications: C1 and C2. If C1 has disabled instance tracking, the flow trace does not show the origin of the message flow and makes it appear as if C2 were directly invoked.

Action	Description
Settings: Payload Validation	Validates the XML schema-based payload at the inbound and outbound points of the composite application revision. If you enable payload validation and there is an invalid payload (that does not follow the schema), a fault is generated for that message. The exception to this is the response message of a synchronous service. That message is not validated, even with payload validation enabled. The inbound message is still validated; only the outbound message is not.
Settings: Enable/Disable Business Monitoring	Select an option to invoke a confirmation dialog that displays the current status of the sensors. ■ **Disable**: Select to disable BPEL sensors for all BPEL components in this SOA composite application. ■ **Enable**: Select to enable BPEL sensors for all BPEL components in this SOA composite application. The **Enable/Disable Business Monitoring** selection is only displayed for composites that have a BPEL service component, regardless of whether that component includes sensors. When BPEL sensors are disabled at the service engine level, you cannot enable or disable BPEL sensors at the SOA composite application level. You can enable or disable BPEL monitors and sensors at the service engine level in the BPEL Service Engine Properties page. For more information, see Section 7.9, "Disabling and Enabling BPEL and BPMN Business Monitors" and Section 12.1, "Configuring BPEL Process Service Engine Properties."
Show WSDL and endpoint URI (icon)	Click to display the end point addresses and WSDLs of all external services for this SOA composite application. **Note**: If you are using the Safari Browser to view this information, see Section B.8.1, "Limitation on Using the Safari Browser to View WSDL File Content."
Show Composite XML Definition (... icon)	Click to show the XML definition of the SOA composite application.

For more information, see the following sections:

- Section 1.4.3.3, "Introduction to the Lifecycle State of SOA Composite Applications"
- Section 3.1, "Configuring SOA Infrastructure Properties"

7.4.3 Starting and Stopping a Managed Oracle WebLogic Server on Which the SOA Infrastructure is Deployed in the Middle of BPEL Processing

If you start and stop a managed Oracle WebLogic Server on which the SOA Infrastructure is deployed in the middle of BPEL processing in a SOA composite application, note the following issues:

- For synchronous BPEL processes

  The whole scenario is synchronous and the instances that are in a running state (after server restart) are pending in the BPEL wait activity. Therefore, the flow thread ends with the server (while sleeping in the wait activity). When the server is restarted, the same instance is not restarted because the flow is synchronous.

Therefore, these instances always remain in a running state because no processing can happen on them after server restart.

- For asynchronous BPEL processes

  If server shutdown occurred in the middle of a BPEL invoke activity, the messages received by BPEL are not handled. BPEL does not automatically recover these messages during restart; they must be recovered manually using Facade API calls. For more information about the Facade API, see Chapter 10, "Programmatically Managing SOA Composite Applications."

7.4.4 Setting the Composite Instance Name

You can set the instance name of a SOA composite application during design time for Oracle Mediator and Oracle BPEL Process Manager. For more information, see Section "Setting the Composite Instance Name at Design Time" of *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

7.5 Automating the Testing of SOA Composite Applications

You can create, deploy, and run test cases that automate the testing of SOA composite applications. Test cases enable you to simulate the interaction between a SOA composite application and its web service partners before deployment in a production environment. This helps to ensure that a process interacts with web service partners as expected by the time it is ready for deployment to a production environment. You create test cases in Oracle JDeveloper and include them in a SOA composite application that is then deployed and administered from Oracle Enterprise Manager Fusion Middleware Control. You can also create BPEL process service component test cases in the SOA composite application test case.

To automate the testing of SOA composite applications:

Note: Before testing SOA composite applications or BPEL process service components from Oracle Enterprise Manager Fusion Middleware Control, see Chapter "Automating Testing of SOA Composite Applications" of *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite* for instructions on creating test cases.

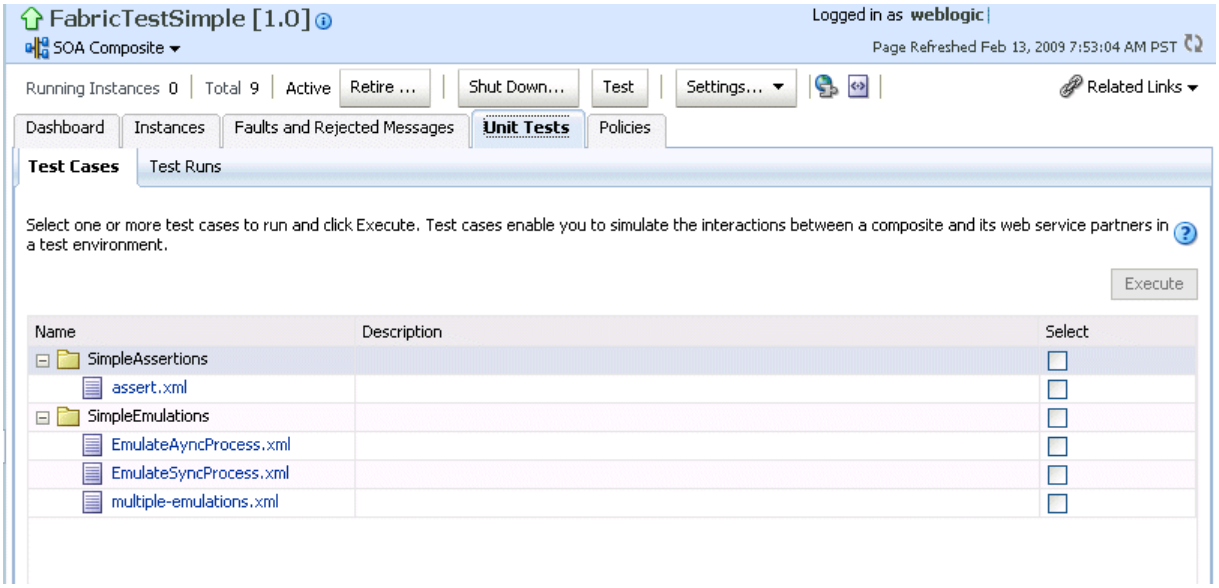
---

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...	From the SOA Composite Menu...
1. Select **Home**. 2. Select **Deployed Composites**. 3. In the **Composite** section, select a specific SOA composite application. 4. Click the **Unit Tests** tab.	1. Under **soa-infra**, expand the partition. 2. Select a specific SOA composite application. 3. Click the **Unit Tests** tab.	1. Select **Unit Test**.

   The test cases that are displayed were designed in Oracle JDeveloper and included in a deployed SOA composite application.

2. Select the entire test suite or individual tests of a suite to run, and click **Execute**.



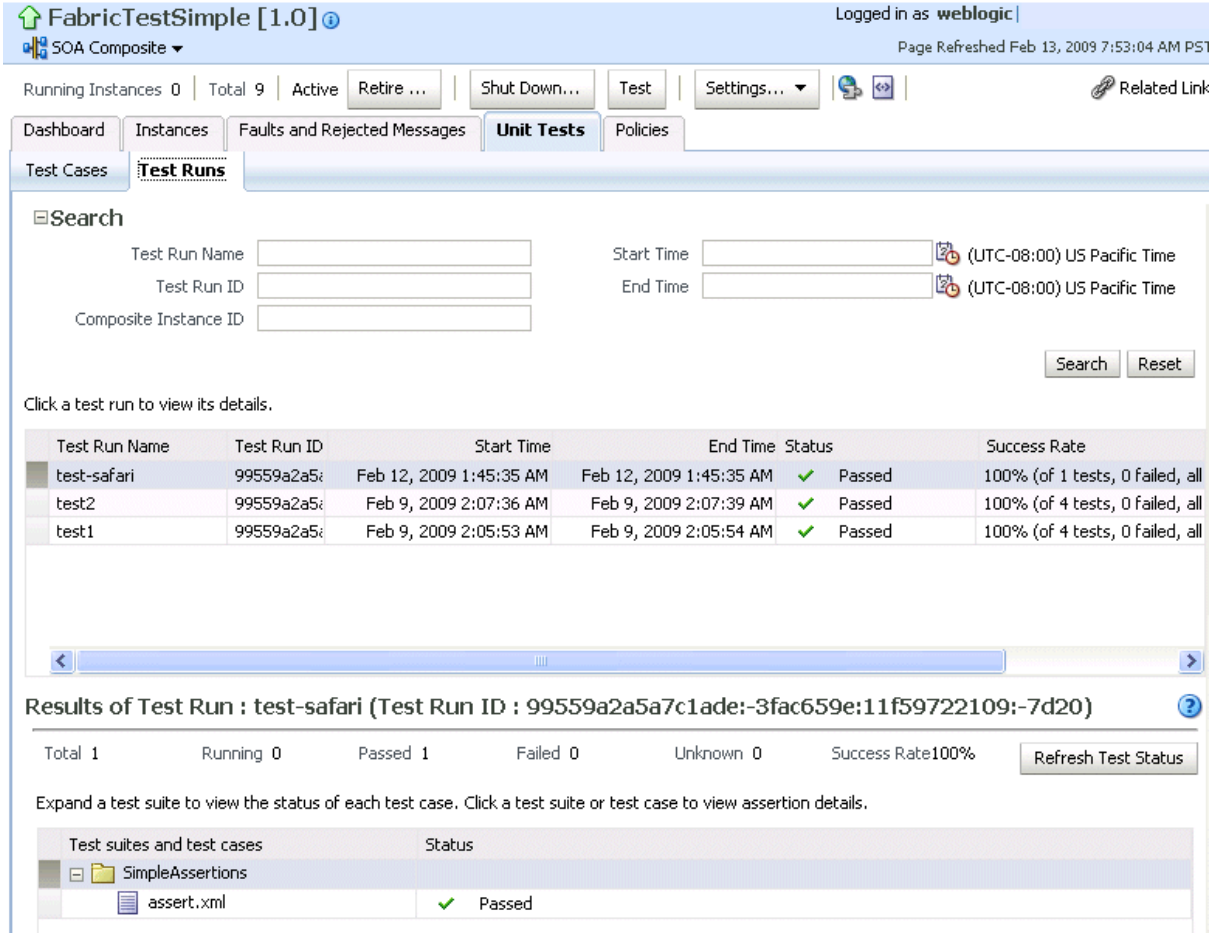
You are prompted to create a test.

3. Enter the following values, and click **OK**.

Field	Description
Test Run Name	Enter a name for the test instance. When testing is complete, report details are captured under this name.
Timeout	Enter a value in seconds in which to complete this test. If the test does not complete within this time limit, then testing is terminated.
Number of Concurrent Test Instances	Enter the number of test instances to create.

The Test Runs page is automatically displayed for tracking the running tests.

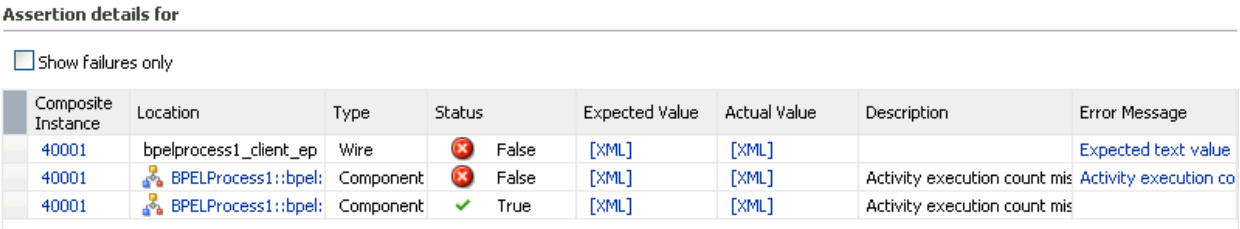
The Test Runs page enables you to track running test cases and view test results. Test suites consist of a logical collection of one or more test cases. Each test case contains a set of commands to perform as the test instance is executed. The execution of a test suite is known as a test run.



4. In the **Test Run Name** column, click a specific test run to display details in the **Results of Test Run** section. If you want to create more test runs, you can switch back to the Test Cases page at any time.

   The **Results of Test Run** section displays details about the executed test run, such as a test summary and the success rate. Click the **Help** icon for additional details.

5. View assertion details at the bottom of the page. Assertions enable you to verify variable data or process flow.



6. Click a composite instance number to view specific test details.

   The composite instances created by executing unit test runs are displayed with a yellow square next to the instance ID in the Instances page of a SOA composite application and in the **Recent Instances** tables of the SOA Infrastructure and SOA composite application. This yellow box distinguishes these instances from test instances created on the Test Web Service page or automatically created by external consumers of the application.

For more information, see the following documentation:

- Section 1.4.3.4, "Introduction to SOA Composite Application Automated Testing"
- Chapter "Automating Testing of SOA Composite Applications" of *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite* for instructions on creating test cases for SOA composite applications and BPEL process service components in Oracle JDeveloper

7.6 Managing SOA Composite Application Policies

You can attach or detach security policies to and from currently deployed SOA composite applications. Policies apply security to the delivery of messages.

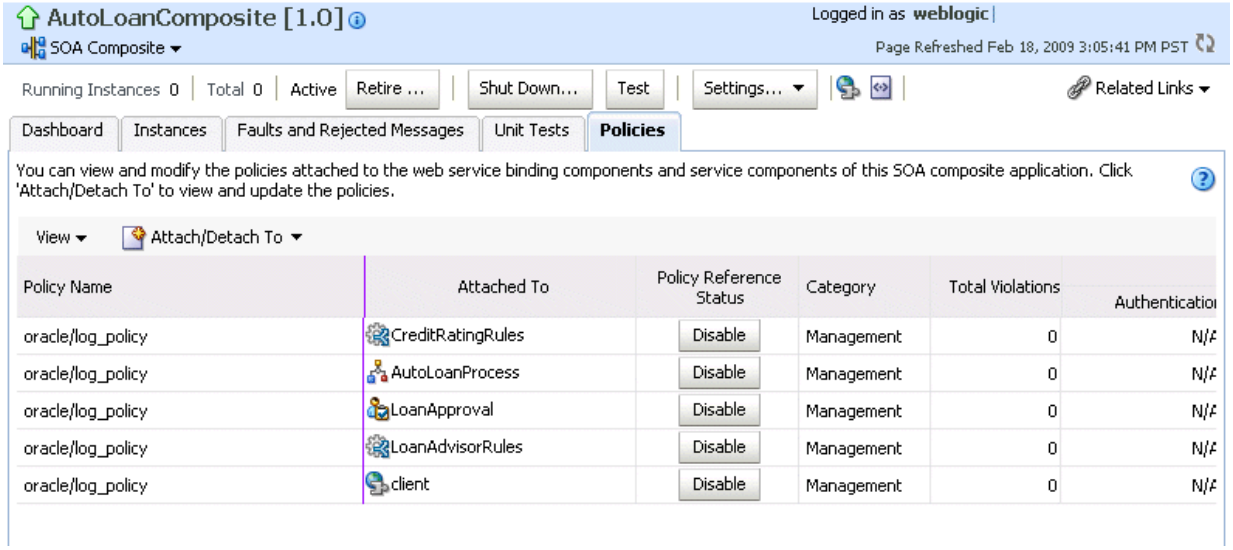
> **Note:** Before attaching policies, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for definitions of available policies and details about which ones to use in your environment.

To manage SOA composite application policies:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...	From the SOA Composite Menu...
1. Select **Home**. 2. Select **Deployed Composites**. 3. In the **Composite** section, select a specific SOA composite application. 4. Click the **Policies** tab.	1. Under **soa-infra**, expand the partition. 2. Select a specific SOA composite application. 3. Click the **Policies** tab.	1. Select **Policies**.

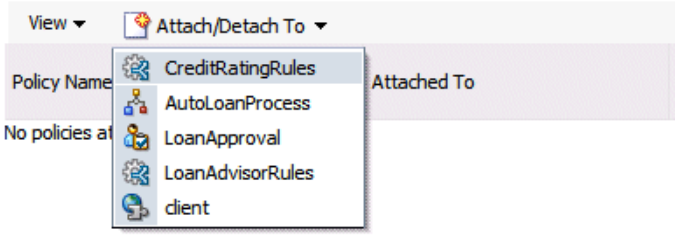
The Policies page enables you to attach and detach policies to and from SOA composite applications. The policies table displays the attached policy name, the component to which the policy is attached, the policy reference status (enabled or disabled) that you can toggle, the category (Management, Reliable Messaging, MTOM Attachment, Security, or WS-Addressing), the violations, and the authentication, authorization, confidentiality, and integrity failures since the SOA Infrastructure was last restarted.



2. Click **Attach/Detach To**.

   If multiple services or components are available, you are prompted to select the service or component for which to perform the attachment or detachment.

3. Select the component to or from which to attach or detach a policy.



   This invokes a dialog for attaching or detaching policies.

   Currently attached policies appear in the **Attached Policies** section. Additional policies available for attachment appear in the **Available Policies** section.

Attach/Detach Policies(AutoLoanComposite/1.0/Service/client/WSBinding) OK Validate Cancel

Attached Policies

Policy Name	Category	Enabled	Description	View Full Description
oracle/log_policy	Management	✓	This policy causes the req...	

Attach Detach

Available Policies

Search Policy Category

Policy Name	Category	Enabled	Description	View Full Description
oracle/wsaddr_policy	WS-Addressin	✓	This policy causes the pla...	
oracle/wsmtom_policy	MTOM Attachr	✓	This Message Transmission ...	
oracle/binding_authorization_denyall_policy	Security	✓	This policy is a special c...	
oracle/binding_authorization_permitall_policy	Security	✓	This policy is a special c...	
oracle/binding_permission_authorization_policy	Security	✓	This policy is a special c...	
oracle/wss10_message_protection_service_policy	Security	✓	This policy enforces messa...	
oracle/wss10_saml_hok_token_with_message_protection_service_policy	Security	✓	This policy enforces messa...	
oracle/wss10_saml_token_service_policy	Security	✓	This policy authenticates ...	
oracle/wss10_saml_token_with_message_integrity_service_policy	Security	✓	This policy enforces messa...	
oracle/wss10_saml_token_with_message_protection_service_policy	Security	✓	This policy enforces messa...	
oracle/wss10_saml_token_with_message_protection_ski_basic256_service_policy	Security	✓	This policy enforces messa...	
oracle/wss10_username_id_propagation_with_msg_protection_service_policy	Security	✓	This policy enforces messa...	
oracle/wss10_username_token_with_message_protection_service_policy	Security	✓	This policy enforces messa...	

4. Select policies to attach that are appropriate to your environment.
5. Click **Attach**.

   The attached policy appears in the **Attached Policies** section.

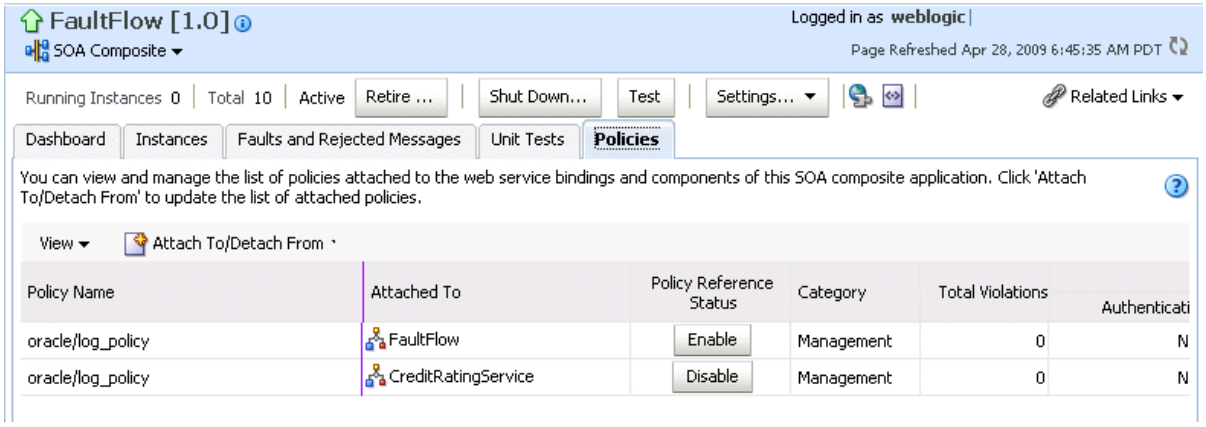
Attach/Detach Policies(AutoLoanComposite/1.0/Service/client/WSBinding) OK Validate Cancel

Attached Policies

Policy Name	Category	Enabled	Description	View Full Description
oracle/log_policy	Management	✓	This policy causes the req...	
oracle/wss10_saml_token_service_policy	Security	✓	This policy authenticates ...	

6. Attach additional policies as needed.
7. When you are finished attaching policies, click **Validate**.
8. If an error message appears, make the necessary corrections until you no longer have any validation errors.
9. Click **OK**.

   The attached policy is displayed in the policies table.



For more information about policies, see the following documentation:

- Section 1.4.3.2, "Introduction to Policies"
- *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for definitions of available policies and details about which ones to use for your environment

7.6.1 WS-RM Sessions

Multiple requests from Oracle SOA Suite in a single WS-RM session are not currently supported. Each request is in an individual WS-RM session.

7.6.2 Policy Attachments and Local Optimization in Composite-to-Composite Invocations

OWSM supports an Oracle SOA Suite local optimization feature for composite-to-composite invocations in which the reference of one composite specifies a web service binding to a second composite. Local optimization enables you to bypass the HTTP stack and SOAP/normalized message conversions during runtime. Local optimization is not used if the composites are in different containers. If a policy is attached to the web service binding, the policy may not be invoked if local optimization is used.

By default, an OWSM security policy includes a `local-optimization` property that identifies if the policy supports local optimization. You can view the setting for a policy in Oracle Enterprise Manager Fusion Middleware Control.

To view the local optimization setting for policies:

1. In the navigator, expand the **WebLogic Domain** folder.
2. Right-click **WLS_SOAWC**, and select **Web Services** > **Policies**.
3. Select a policy and click **Export to File**.
4. Open the file with a text editor and search for `local-optimization` to identify the value. This property supports the following values:
   - `on`: Local optimization is used in the attached policy, and the policy is not applied at runtime.
   - `off`: Local optimization is not used in the attached policy, and the policy is applied at runtime.

- `check-identity`: If a JAAS subject exists in the current thread, local optimization is used. Otherwise, local optimization is not used.

For information on the default local optimization settings for security policies, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

You can override the local optimization setting for a policy by adding the `oracle.webservices.local.optimization` property in the binding section of the `composite.xml` file. The following values are supported:

- `true` (default value): Local optimization is used, and the policy is applied if it is applicable to optimized calls (details are defined in the individual policy file).
- `false`: Local optimization is not used, regardless of the default setting for the `local-optimization` property at the OWSM policy level. This setting forces the policy to be applied.

For example, the following setting of `false` causes `oracle/wss_username_token_client_policy` to be applied.

```
 <binding.ws
port="http://xmlns.oracle.com/CalledBPELProcessApp_
jws/CalledBPELProcess/CalledBPELProcess#wsdl.endpoint(calledbpelprocess_client_
ep/CalledBPELProcess_pt)"

location="http://sta00634.us.oracle.com:8001/soa-infra/services/default/CalledBPEL
Process!1.0/calledbpelprocess_client_ep?WSDL">
      <wsp:PolicyReference URI="oracle/wss_username_token_client_policy"
                           orawsp:category="security"
orawsp:status="enabled"/>
      <wsp:PolicyReference URI="oracle/log_policy"
orawsp:category="management"
                           orawsp:status="enabled"/>
                            <property
name="oracle.webservices.local.optimization">false</property>
    </binding.ws>
```

For more information about local optimization, see Section 3.7, "Configuring Local Optimization."

7.7 Exporting a Deployed SOA Composite Application

You can export the contents of a deployed SOA composite application to an archive JAR file. The file can include some or all of the following data:

- The original design-time composite
- Postdeployment changes in the rules dictionary and domain value maps (DVMs)
- Postdeployment property changes such as binding component properties, composite properties such as audit level settings and payload validation status, and policy attachments

---

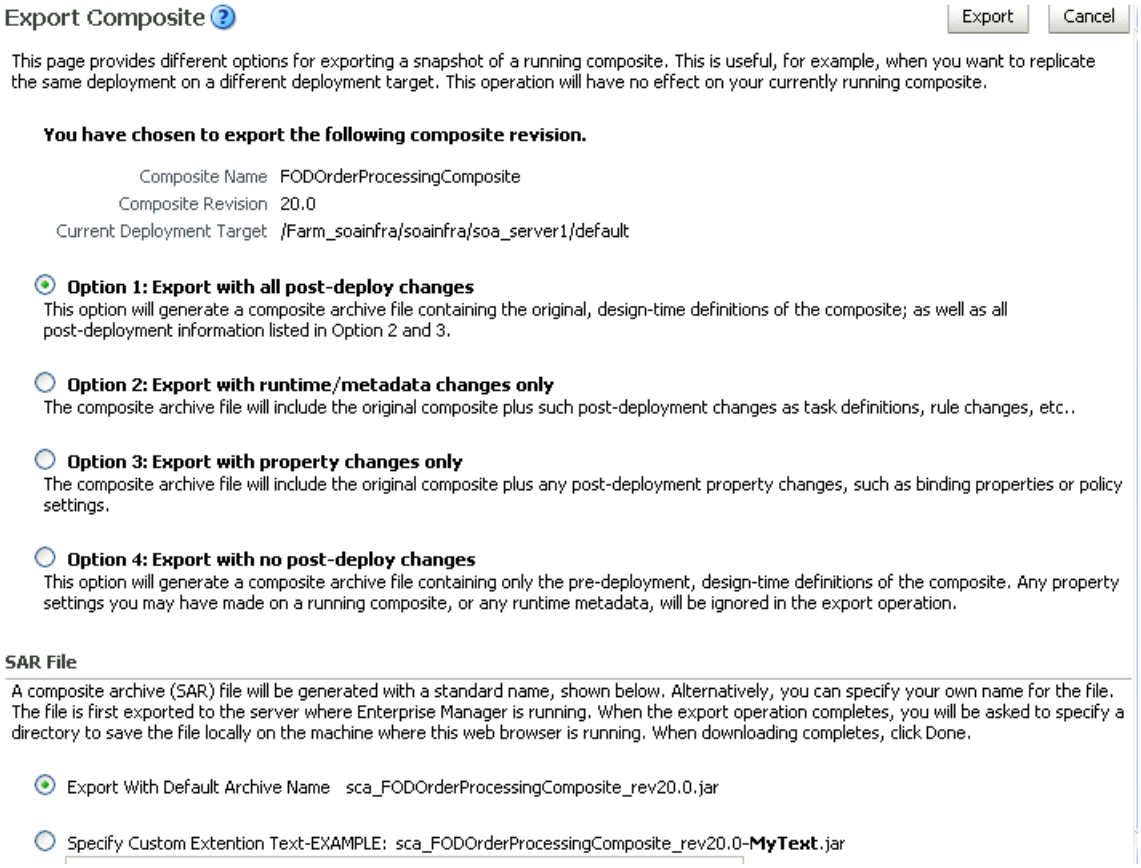
Notes:

- SOA composite application exporting is currently only allowed at the individual SOA composite level.
- Shared metadata is not exported as part of the composite export SOA archive (SAR).

To export a running SOA composite application:

1. Go to the home page of the SOA composite application to export.
2. From the **SOA Composite** menu, select **Export**.

   The Export Composite page appears.



3. Select an option.
   - **Option 1**: Generates an archive file containing the original design-time composite and the postdeployment details described in **Option 2** and **Option 3**.
   - **Option 2**: Includes the original design-time composite and postdeployment changes in the rules dictionary and DVMs.
   - **Option 3**: Includes the original design-time composite and postdeployment property changes such as binding component properties, composite properties such as audit level settings and payload validation status, and policy attachments.
   - **Option 4**: Generates an archive file containing only the original design-time composite. Options 2 and 3 are not included.
4. If you want to append an additional name to the existing file, select **Specify Custom Extension Text**. For example, entering `MyText` to a file named `sca_OrderBookingComposite_rev1.0.jar` names the exported file as `sca_OrderBookingComposite_rev1.0-MyText.jar`.
5. Click **Export**.

The Processing: Export Composite dialog displays the progress of archive file generation. When generation completes, you are prompted to save the file.

6. Click **Save File**.

   A dialog appears for either opening or saving the file to a directory on your local host.

   > **Note:** It is important that you click the **Save File** button. Do not simply close this dialog. Although the composite is exported, you cannot retrieve the actual exported file.

7. Specify the local directory in which to save the JAR file.
8. In the upper right of the Processing: Export Composite dialog, click the **x** icon to close the dialog.
9. On the Export Composite page, note that the Cancel button has changed to **Done**.
10. Click **Done**.

    The Export Composite is closed and you are returned to the SOA composite application home page.

7.8 Grouping SOA Composite Applications into Partitions

You can deploy SOA composite applications into separate sections of the SOA Infrastructure known as partitions. Deploying to partitions enables you to logically group SOA composites and perform bulk lifecycle management tasks on all SOA composite applications within a specific partition. Partitions are similar to the domain feature that was part of 10.1.*x* releases of Oracle BPEL Process Manager. However, you *cannot* perform specific configuration tasks on partitions, such as restricting login access to a specific partition or configuring partitions (such as configuring threading).

At least one partition is required for deploying SOA composite applications. A default partition named **default** is automatically included with Oracle SOA Suite.

You can manage partitioning from either of two pages:

- From the Manage Partitions page of the SOA Infrastructure, which lets you create partitions, delete partitions, and perform bulk lifecycle management tasks on all SOA composite applications in a specific partition
- From the partition home page, which also enables you to perform bulk lifecycle management tasks on all SOA composite applications in a specific partition

---

Note: If SOA composite applications using the same inbound resource are deployed to different partitions, it cannot be guaranteed which partition picks up the message for processing.

For example, assume you are using the file adapter and `/home/Directory1` is the inbound directory for the composite **SOAComposite1**. If this composite is deployed to both **Partition1** and **Partition2**, when a file is placed in `/home/Directory1`, either the composite in **Partition1** or **Partition2** may pick up the file.

With the socket adapter, however, there is a limitation that does not permit you to deploy any composite that uses the same inbound port. In that case, an exception is thrown indicating that the inbound port is in use.

Table 7–2 provides more specific details on the tasks you can perform from both pages.

Table 7–2 Partition Management Actions

Action	Perform on the Manage Partitions Page?	Perform on the Partition Home Page?
Create a partition	Yes	No
Delete a partition	Yes	Yes. Select the **SOA Partition** menu, then select **Delete This Partition**. **Note**: You can also delete a partition by right-clicking it in the navigator and selecting **Delete This Partition**.
Perform bulk lifecycle management tasks on all composites deployed to a specific partition: ■ Start all ■ Shut down all ■ Retire all ■ Activate all ■ Undeploy all	Yes	Yes

Notes:

- Partitions are *not* associated with a particular state such as started, stopped, activated, or retired. Only the composites within the partition are associated with a particular state. Therefore, you *cannot* start, stop, activate, or retire a partition.
- After the SOA Infrastructure is started, it may not be completely initialized to administer incoming requests until all deployed composites are loaded. During SOA Infrastructure initialization, a warning message is displayed at the top of the Manage Partitions and Partitions home pages. Do not perform operations such as composite deployment, composite undeployment, and others while this message is displayed. For more information, see Section 3.2.1, "Waiting for SOA Infrastructure Startup Initialization to Complete."

See the following section based on the tasks you want to perform:

- Section 7.8.1, "Creating and Deleting Partitions"
- Section 7.8.2, "Performing Bulk Lifecycle Management Tasks on Composites in Partitions"

For more information about partitions, see Section 1.4.3.5, "Introduction to Partitioning of the SOA Infrastructure."

7.8.1 Creating and Deleting Partitions

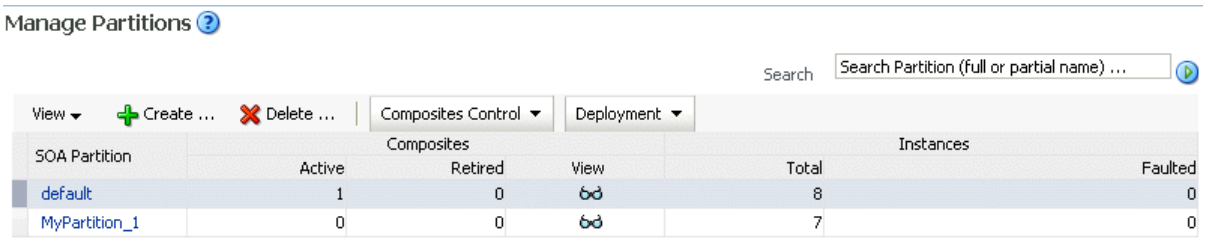
You can create and delete partitions on the Manage Partitions page. A default partition named **default** is automatically included with Oracle SOA Suite. You can delete the **default** partition. You cannot rename existing partitions; only creation and deletion of partitions is supported.

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the Home Page of a Specific Partition...
1. Select **Manage Partitions**.	1. From the **Related Links** list of a specific partition, select **Manage Partitions**.

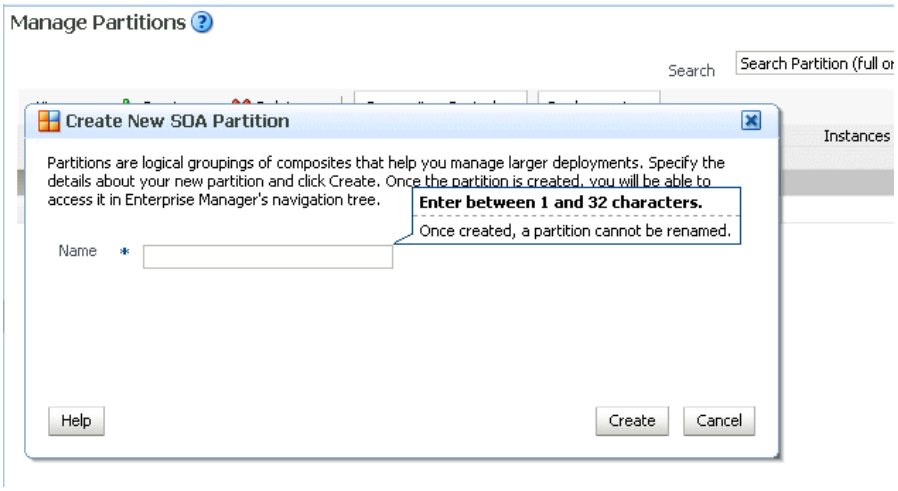
The Manage Partitions page displays the following details:

- The name of each partition, the number of active and retired SOA composite application revisions in each partition, the name of the composites contained in each partition (under the **View** link), and the total number of running and faulted instances in each partition.
- A utility for searching for a specific partition. Enter a full or partial partition name and click the **Search** icon or press the **Return** key. The search is not case-sensitive.



2. To add a partition, click **Create**.

   The Create New SOA Partition dialog is displayed.



   a. In the **Name** field, enter a partition name, and click **Create**.

   Note: The name must conform to the following conventions:

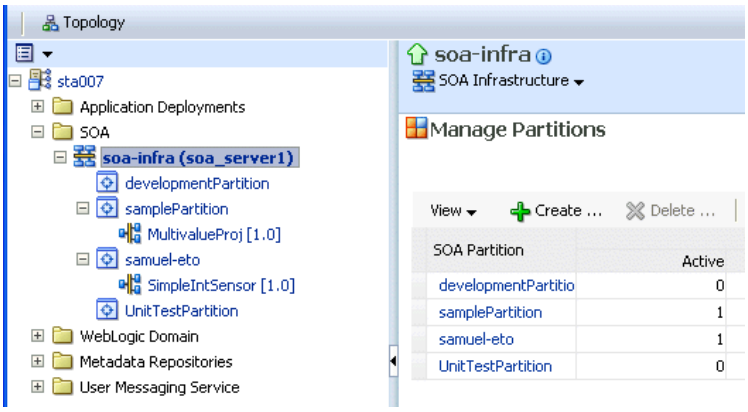
   - ASCII letters and numbers are permitted.
   - Underscores (_) are permitted.
   - Hyphens (-) are permitted (except as the first character).
   - Non-ASCII letters are permitted.
   - Spaces are *not* permitted.

   Examples of valid names are `mypartition`, `partition2`, `dept-a`, `customer_services`, and `22`. Examples of invalid names are `-part2`, `/partition`, and null or empty names.

   You cannot rename an existing partition or later transfer the composite applications you deployed to it to a different partition.

   The new partition is displayed in both the navigator under **soa-infra** and the **SOA Partition** column of the Manage Partitions page. You can now deploy composites to this partition by selecting **Deploy to This Partition** from the **Deployment** dropdown list or right-clicking a specific partition in the navigator and clicking **Deploy to This Partition**.

   When a composite is deployed to a partition, it is displayed beneath the partition in the navigator. Once deployed, a composite cannot be transferred to a different partition.

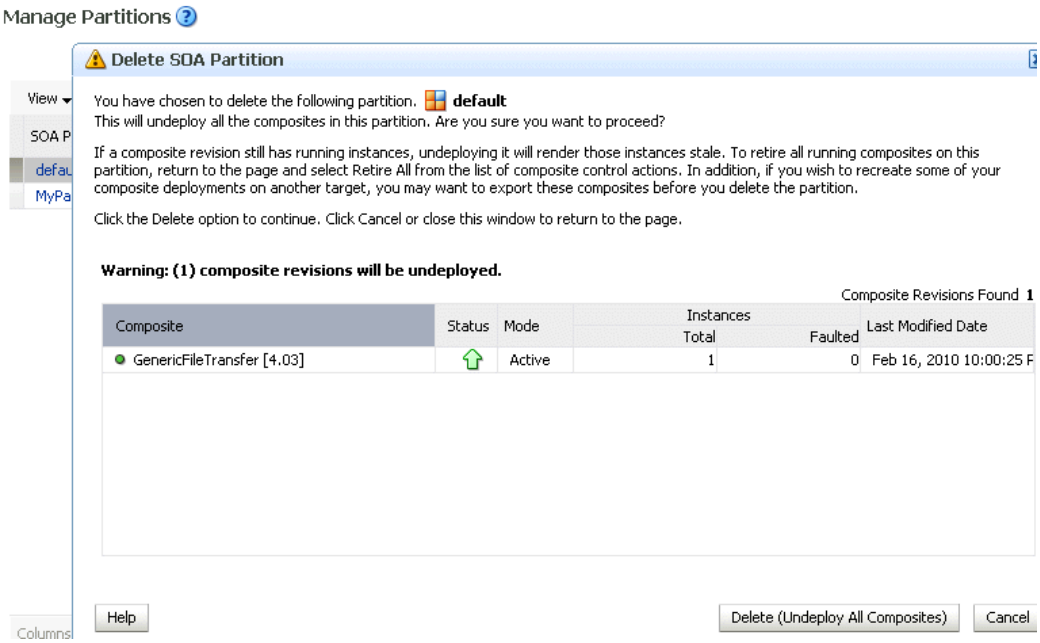


3. To delete a partition, select a specific partition and click **Delete**. You can also right-click a specific partition in the navigator and click **Delete This Partition**.

   The Delete SOA Partition dialog is displayed. Note the following:

   - If you want to re-create some of your composite deployments in another partition, you can export those composites to a JAR file *before* you delete this partition.
   - Before deleting the selected partition, all SOA composite application revisions in the partition are undeployed. The states of all undeployed instances of these revisions become stale.

   Note: You must have at least one partition. If you delete all partitions, you cannot deploy a SOA composite application.



   a. Click **Delete (Undeploy All Composites)**.

      All composites that were deployed in the partition are undeployed and no longer appear in the navigator. The partition is then deleted from both the

navigator under **soa-infra** and the **SOA Partition** column of the Manage Partitions page.

For information about performing bulk lifecycle management tasks from the **Composites Control** and **Deployment** lists, see Section 7.8.2, "Performing Bulk Lifecycle Management Tasks on Composites in Partitions."

You can also create partitions with the Oracle WebLogic Scripting Tool (WLST) and `ant` commands. For information, see *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference* and *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

7.8.2 Performing Bulk Lifecycle Management Tasks on Composites in Partitions

You can perform bulk lifecycle management tasks on all SOA composite applications in a specific partition on the Manage Partitions page, on the home page of a specific partition, and from the menu that is displayed when you right-click a partition in the navigator.

Bulk lifecycle management tasks impact not one, but many, composites at once. If a composite has running instances and a lifecycle changing operation is performed on the composite, the instances may not complete. For information about how different lifecycle operations impact the composite instances, see Step 3 of Section 7.4.1, "Managing the State of All Applications at the SOA Infrastructure Level."

To perform bulk lifecycle management tasks on all SOA composite applications in a specific partition:

1. Access either page through one of the following options:

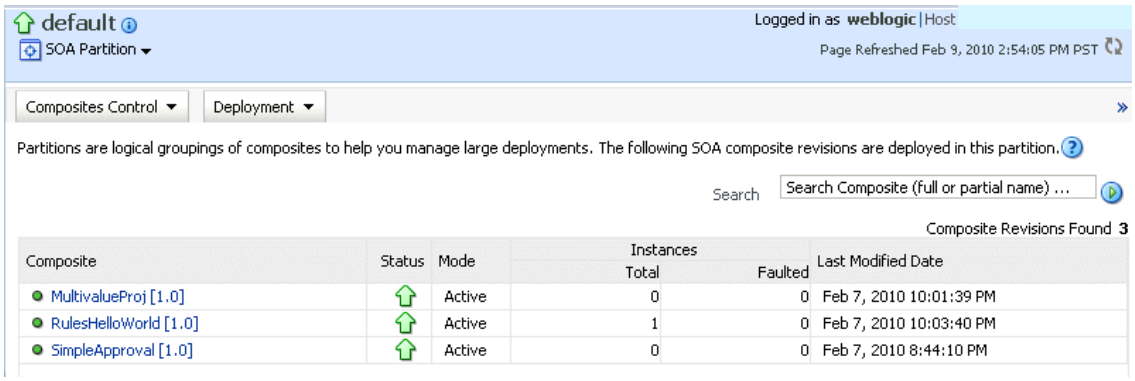
From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Manage Partitions**.	1. Under **soa-infra**, select a specific partition.

Note: As a shortcut, you can also right-click a specific partition in the navigator to display a menu for selecting the bulk lifecycle management actions described in this section. For more information about this menu, see Step 3 of Section 2.2.3, "Navigating Through the Partition Home Page and Menu."

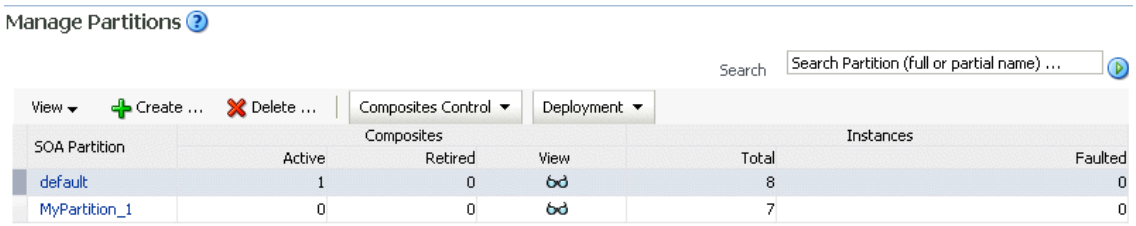
Two dropdown lists that are displayed on either page enable you to perform bulk lifecycle management actions:

- **Composites Control** list
- **Deployment** list

On the home page of a specific partition, these lists are displayed at the top of the page.

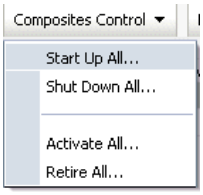


On the Manage Partitions page, these lists are displayed above the **SOA Partition** table:



Note: You can also select to deploy composites to a partition and perform bulk lifecycle management tasks by selecting the **SOA Partition** menu at the top of the partition home page.

2. To perform one of the following bulk lifecycle management tasks for all SOA composite applications contained in the selected partition, select the **Composites Control** list:
   - Start all composites.
   - Shut down all composites.
   - Activate all composites.
   - Retire all composites.



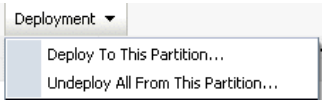
   a. Select an operation to perform.

      A dialog is displayed that prompts you to confirm your selection. When the operation completes, a confirmation message is displayed at the top of the page.

3. To perform one of the following management tasks, select the **Deployment** list:
   - Specify a composite to deploy to this partition. This selection invokes the Deploy SOA Composite wizard where you specify a composite revision to deploy.

- Undeploy all composites in this partition.

  A dialog is displayed that prompts you to confirm your selection. When the operation completes, a confirmation message is displayed at the top of the page.



7.9 Disabling and Enabling BPEL and BPMN Business Monitors

The term business monitoring comprises different types of sensors that can be defined for some types of SOA components, such as the following:

- BPEL sensors: Enable you to create sensors in BPEL faults, activities, and variables.
- BPEL Monitors: Enable you to capture BPEL process metrics that are sent to Oracle BAM Server, and then used for analysis and graphic display.
- BPMN measurements: Enable you to measure a business indicator at a certain point in the process or in a section of the process.

At the SOA composite application level, you set the same status for all sensors defined for all types of service components comprising the selected composite. You cannot selectively enable or disable sensors defined for a specific type of service component for just one composite. However, you can globally disable service component-type specific sensors for all composites in the respective BPEL Service Engine Properties page or BPMN Service Engine Properties page.

By default, BPEL and BPMN sensors defined in SOA composite applications are enabled. Disabling sensors means that sensor values are not captured during runtime. For example, this results in the values not being displayed in the **Sensor Values** section of the BPEL audit trail.

To disable sensors at the service engine level:

1. Access the BPEL Service Engine Properties page by following the steps in Section 12.1, "Configuring BPEL Process Service Engine Properties."
2. Select the **Disable BPEL Monitors and Sensors** checkbox.
3. Click **Apply**.
4. Access the BPMN Service Engine Properties page by following the steps in Section 37.1, "Configuring BPMN Process Service Engine Properties."

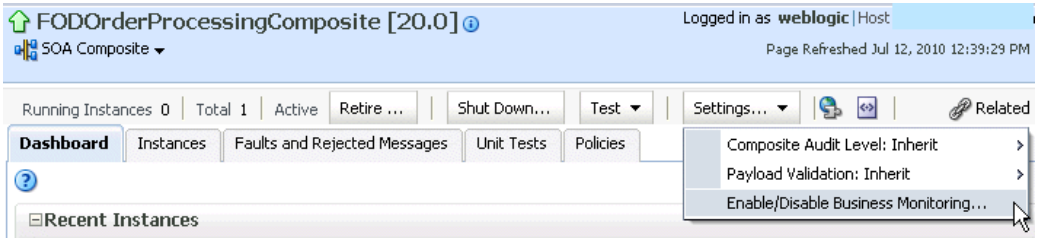
   Note: The BPMN Service Engine Properties page is only displayed if Oracle BPM Suite is installed.

5. Select the **Disable BPMN Measurements** checkbox.
6. Click **Apply**.

To disable or enable sensors at the SOA composite application level:

1. Go to the home page of the SOA composite application in which you want to disable or enable sensors.

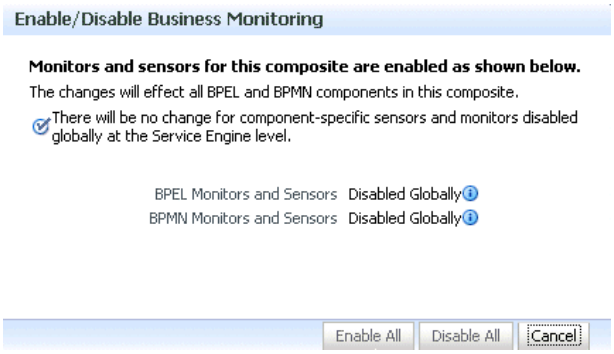
2. From the **Settings** menu, select **Enable/Disable BPEL Business Monitoring**. This selection is only displayed for composites that have at least one BPEL or BPMN service component, regardless of whether those components include sensors.



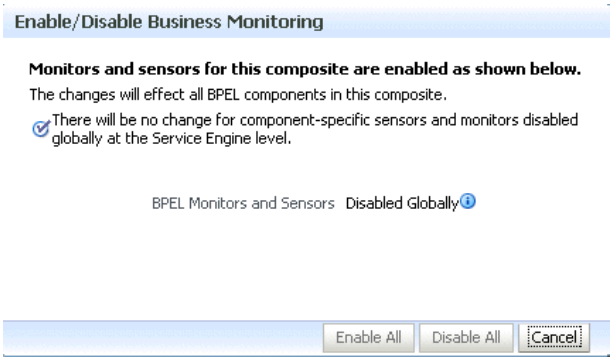
A dialog is invoked that displays the current status of sensors and enables you to change that status. The dialog only displays the options applicable to the component types present in the selected composite. For example, if the composite contains only BPEL components and not BPMN components, you see only the option to set the status of BPEL sensors.

The following steps describe the types of dialogs that can be displayed and the available actions.

a. If sensors are disabled at both service engine levels, the message **Disabled Globally** is displayed for each. You cannot select **Enable All** or **Disable All** in this dialog. Both buttons are disabled.

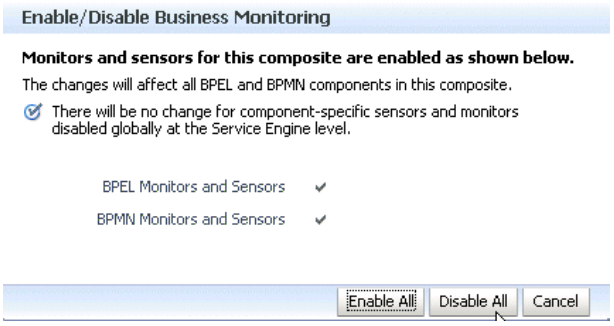


In addition, if sensors are disabled at the BPEL service engine level and the BPMN service engine does not appear because Oracle BPM Suite is not installed, you cannot select **Enable All** or **Disable All** in this dialog. Both buttons are disabled.



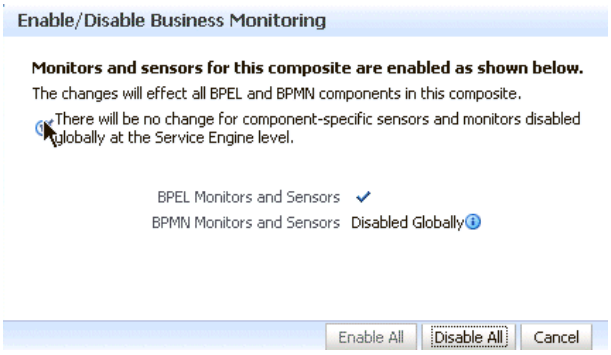
b. If sensors are not disabled at the composite level, checkmarks are displayed. If sensors are also not disabled at both the BPEL and BPMN service engine levels, the message **Disabled Globally** does *not* display.

Click **Disable All** to disable all types of sensors defined for service components that comprise the selected composite. (If sensors are disabled at the service engine level, they remain disabled.)



c. If sensors are disabled at a specific service engine level, the sensor status you set for those types of sensors at the composite application level only takes effect when the corresponding **Disable BPEL Monitors and Sensors** or **Disable BPMN Measurements** checkbox in the service engine Properties page is deselected.

For example, if sensors are disabled at the BPMN service engine level (as shown below), and you select **Enable All** for all sensors at the selected composite level, that status is only applied to other types of sensors, such as BPEL. BPMN sensors and monitors remain disabled. However, if you later change the BPMN service engine setting, BPMN sensors are automatically enabled in this composite.



d. If sensors are disabled at the composite level, no checkmark is displayed. Click **Enable All** to enable all types of sensors defined for service components that comprise the selected composite. (Sensors disabled at the service engine level remain disabled until you change the service engine level setting.) Because the composite does not include BPMN service components, BPMN is not displayed.

After you select an action, an inline message is displayed in the page confirming that sensors were enabled or disabled.

For more information about BPEL sensors and monitors, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

For more information about BPMN measurements, see *Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management*.

8

Managing SOA Composite Application Instances

This chapter describes how to manage SOA composite application instances, including initiating a test instance of an application, monitoring and deleting instances, recovering from faults, and deleting rejected messages.

This chapter includes the following topics:

- Section 8.1, "Initiating a SOA Composite Application Test Instance"
- Section 8.2, "Monitoring and Deleting SOA Composite Application Instances from the Application Home Page"
- Section 8.3, "Monitoring and Deleting SOA Composite Application Instances at the SOA Infrastructure Level"
- Section 8.4, "Recovering from SOA Composite Application Faults at the SOA Infrastructure Level"
- Section 8.5, "Recovering from SOA Composite Application Faults in the Application Home Page"
- Section 8.6, "Deleting Rejected Messages at the SOA Infrastructure Level"
- Section 8.7, "Deleting Rejected Messages from the Application Home Page"

> **Note:** The procedures in this guide describe how to access Oracle Enterprise Manager Fusion Middleware Control pages from the **SOA Infrastructure** menu, **soa-infra** icon in the navigator, **SOA Composite** menu, and **SOA Partition** menu. You can also access many pages from the Farm home page. For more information, see Section 2.2.6, "Navigating to the SOA Infrastructure or SOA Composite Application Home Page from the Farm Home Page."

8.1 Initiating a SOA Composite Application Test Instance

This section describes how to initiate a test instance of a deployed SOA composite application.

To initiate a SOA composite application test instance:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...	From the Composite Menu...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application. 4. At the top of the page, click **Test**.	1. Under **soa-infra**, expand the partition. 2. Select a specific SOA composite application. 3. At the top of the page, click **Test**.	1. Select **Test Service** > **client**.

Note: The **Test** button is disabled in the following situations:

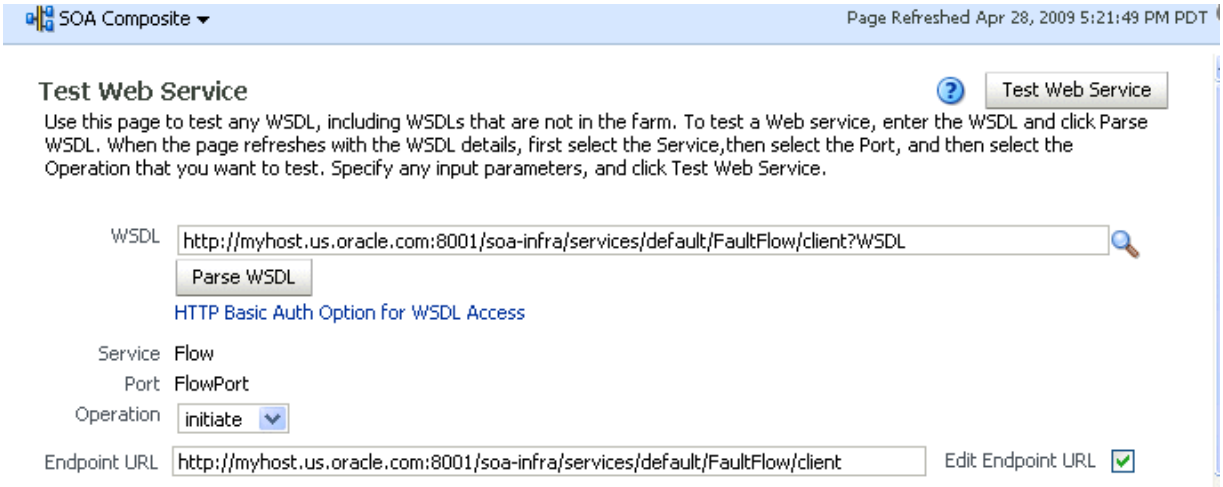
- The SOA composite application revision is stopped or retired.
- There are no web services available for the application. Only composites having services with web service bindings can be tested from this page.

2. If the composite includes multiple services, the **Test** button has a drop-down list to select the service to test.

   The Test Web Service page for initiating an instance appears.

   This page provides many options for initiating an instance. At a minimum, you must specify the XML payload data to use in the **Input Arguments** section.

   The WSDL file and endpoint URL are populated automatically based on the service you selected to test. The endpoint URL is derived from the WSDL and can be overridden to invoke that service at a different location. If the service selected has multiple ports, a drop-down list is displayed. Otherwise, the port of the current service is displayed.



3. Accept the default values for these fields or provide values appropriate to your test environment.

4. If you change the WSDL file, click **Parse WSDL** to reload the WSDL file.

   If the WSDL URL does not contain the revision number, it is processed by the default composite application. For example, if there are two revisions of a

composite application named `HelloWorld`, then the following endpoints are exposed by them:

- `http://host:port/soa-infra/services/default/HelloWorld!1.0/client`
- `http://host:port/soa-infra/services/default/HelloWorld!2.0/client`

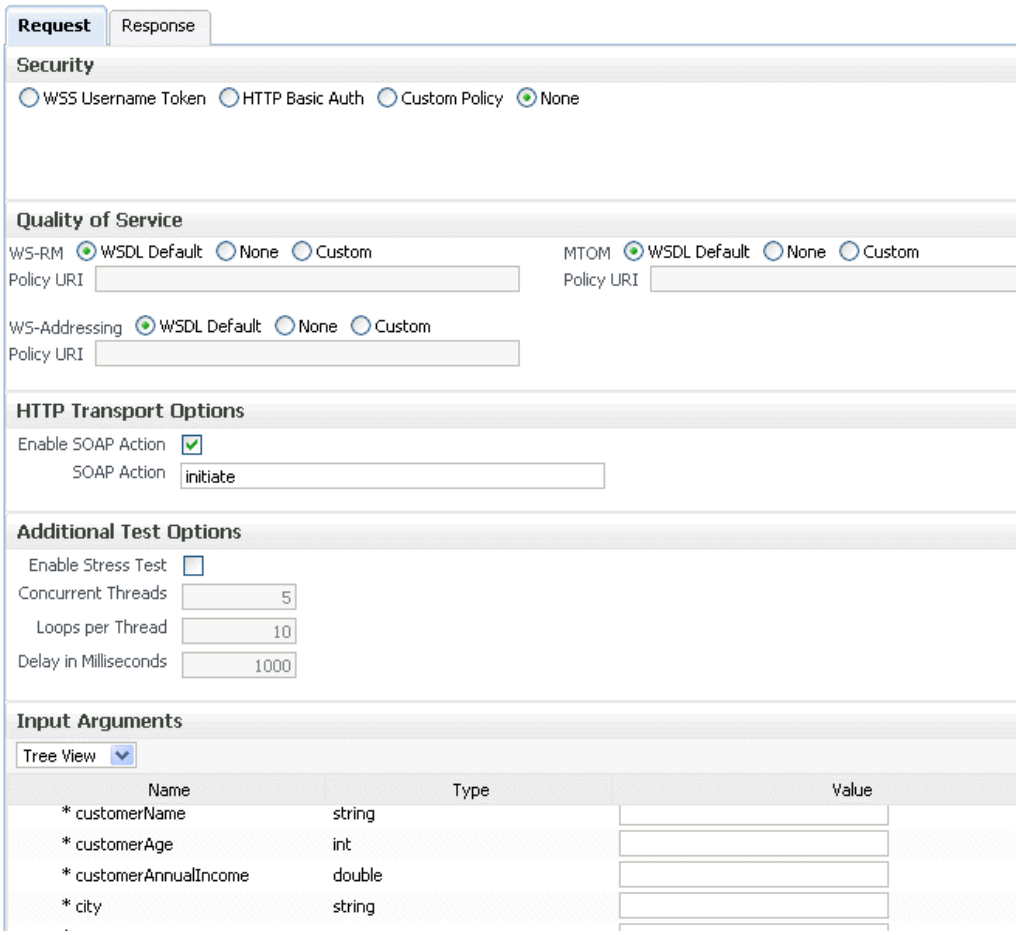
However, if the WSDL specified for web service invocation does not contain the revision details (for example, `http://host:port/soa-infra/services/default/HelloWorld/client`), it is processed by the composite revision that is set as default.

5. Select the operation that you want to test from the **Operation** menu. The available operations are determined from the WSDL.

   To test a RESTful web service, select the **GET** or **POST** service port operation.

6. If you want to edit the endpoint URL, click **Edit Endpoint URL** and make appropriate changes.

   The lower part of the Test Web Service page consists of the **Request** tab. This tab enables you to specify security, quality of service, HTTP transport, stress testing options, and XML input arguments:



The **Security** section includes the following fields for passing security properties with messages.:

Field	Description
WSS Username Token	Inserts a WS-Security SOAP header. The **Username** field is required, and the **Password** field is optional.
HTTP Basic Auth	Inserts the username and password credentials in the HTTP transport header. Both the **Username** and **Password** fields are required.
Custom Policy	Uses a custom policy to authenticate the user (specifies the URI for the custom policy). The **Username** and **Password** fields are optional.
None	Select to not specify security credentials. This is the default selection.

When testing RESTful Web services, because the SOAP protocol is not used, the only security options are **HTTP Basic Authentication** or **None**.

7. Accept the default values for these fields or provide values appropriate to your test environment.

   The **Quality of Service** section includes the following fields. Oracle Fusion Middleware uses a policy-based model to manage web services. A policy applies behavior requirements to the delivery of messages. This section is not available when testing RESTful web services. For additional details about using the Test Web Service page, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

Field	Description
WS-RM	Select one of the following options for testing WS-Reliable Messaging (RM) protocol policies. Reliable messaging policies support this protocol, which guarantees the end-to-end delivery of messages. ■ **WSDL Default**: Executes the default behavior of the WSDL. For example, if the WSDL contains a reference to a WS-RM policy, then the policy is enforced. If the WSDL does not contain a reference to a WS-RM policy, then reliable messaging is not tested. ■ **None**: No policy for WS-RM is tested even if the WSDL contains a reference to a policy. ■ **Custom**: Enforces a custom policy. Specify the URI of the custom policy in the **Policy URI** field. If a WS-RM policy is referenced in the WSDL, it is ignored, and the policy specified in the **Policy URI** field is used instead.
MTOM	Select one of the following options for testing Message Transmission Optimization Mechanism (MTOM) policies. MTOM policies ensure that attachments are in MTOM format, a format for efficiently sending binary data to and from web services. ■ **WSDL Default**: Executes the default behavior of the WSDL. For example, if the WSDL contains a reference to an MTOM policy, then the policy is enforced. If the WSDL does not contain a reference to an MTOM policy, then MTOM is not tested. ■ **None**: No policy for MTOM is tested, even if the WSDL contains a reference to a policy. ■ **Custom**: Enforces a custom policy. Specify the URI of the custom policy in the **Policy URI** field. If an MTOM policy is referenced in the WSDL, it is ignored, and the policy specified in the **Policy URI** field is used instead.

Field	Description
WS-Addressing	Select one of the following options for testing WS-Addressing policies. WS-Addressing policies verify that SOAP messages include WS-Addressing headers in conformance with the WS-Addressing specification. ■ **WSDL Default**: Executes the default behavior of the WSDL. For example, if the WSDL contains a reference to a WS-Addressing policy, then the policy is enforced. If the WSDL does not contain a reference to a WS-Addressing policy, then WS-Addressing is not tested. ■ **None**: No policy for WS-Addressing is tested even if the WSDL contains a reference to a policy. ■ **Custom**: Enforces a custom policy. Specify the URI of the custom policy in the **Policy URI** field. If a WS-Addressing policy is referenced in the WSDL, it is ignored, and the policy specified in the **Policy URI** field is used instead.

8. Accept the default values for these fields or provide values appropriate to your test environment.

   The **HTTP Transport Options** section includes the following fields:

Field	Description
Enable SOAP Action	Specifies whether the WSDL `soap:operation` has a `soapAction` attribute. This flag is enabled if a `soapAction` attribute exists. If you do not want to send a request with the SOAP action HTTP header, then clear the checkbox.
SOAP Action	Displays the `soapAction` attribute of the WSDL `soap:operation`, if one exists. You may specify a different SOAP action in this text box.

   This section is not available when testing RESTful web services.

9. Accept the default values for these fields or provide values appropriate to your test environment.

   The **Additional Test Options** section includes the following fields. This section provides a simple stress test that simultaneously invokes multiple instances.

> **Note:** This is *not* a real stress test tool. Therefore, do not enter huge values for both concurrent threads and the number of times to invoke the operation. Doing so can result in errors.

Field	Description
Enable Stress Test	Click **Enable** to create a simple stress test. With this enabled, no conversation ID is displayed.
Concurrent Threads	Enter the number of concurrent threads on which to send the invocations. The default is `5` threads.
Loops per Thread	Enter the number of times to invoke the operation. The default is `10` times.
Delay in Milliseconds	Specify the delay of milliseconds to wait between operation invocations. The default is `1000` milliseconds (`1` second).

10. Accept the default values for these fields or provide values appropriate to your test environment.

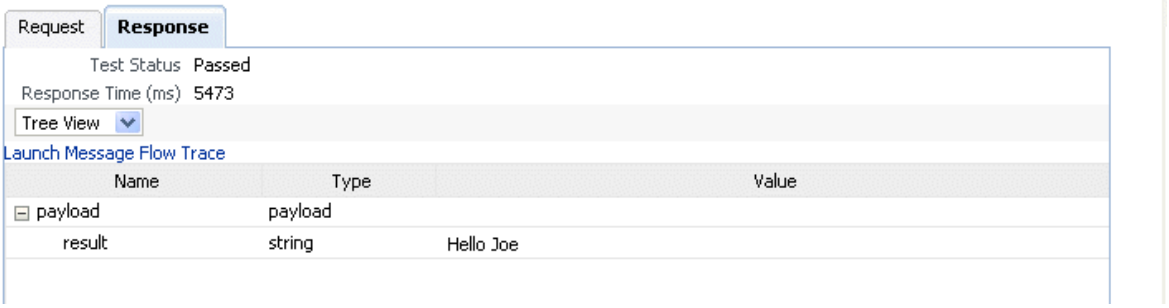
    The **Input Arguments** section includes the following fields for entering XML payload data.

Field	Description
Tree View	Displays a graphical interface of text fields in which to enter information. This field automatically generates the required headers and XML structure.
XML View	Displays the XML file format for inserting values. You can paste the raw XML payload of your message into this field.

> **Note:** If you are using Oracle Enterprise Manager Grid Control, you can save the payload you enter. This feature is not available with Oracle Enterprise Manager Fusion Middleware Control.

11. Click **Test Web Service**.

    The test results appear in the **Response** tab upon completion.



> **Note:** The **Response** tab does not display payload data if you are performing a stress test or are testing an asynchronous service.

12. Click **Launch Message Flow Trace** to access the flow trace of the instance.
13. To return to the composite home page, click the name of the composite that appears at the top of the page or select **Home** from the composite target menu.
14. Return to the Dashboard page of the SOA composite application.

    The **Recent Instances** table lists recent SOA composite application instances. Each created instance has its own unique ID.

For more information, see the following sections:

- Section 1.2.3, "Introduction to SOA Composite Application Instances" for conceptual details about instances
- Section 1.4.3.2, "Introduction to Policies" for an overview of policies
- *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for specific details about policies and testing web services from the Test Web Service page

8.1.1 Specifying RPC/Literal-Style WSDL Files on the Test Web Service Page

If you are specifying an RPC/literal-style WSDL file with a message defined by `"element="` in the Test Web Service page in Oracle Enterprise Manager Fusion Middleware Control, use the **XML View** option of the **Input Arguments** section to modify the SOAP message. The SOAP body should look as follows:

```
<soap:Body>
    <ns:initiate>
        <payload>
          <value xmlns="...">3</value>
        </payload>
    </ns:initiate>
</soap:Body>
```

where `initiate` is the operation name, `payload` is the part name, and `value` is the element defined in the WSDL message/part.

8.2 Monitoring and Deleting SOA Composite Application Instances from the Application Home Page

Section 7.4, "Managing the State of Deployed SOA Composite Applications" describes how to manage the lifecycle state of SOA composite applications. You can also monitor and delete specific SOA composite application instances from the Instances page of the application home page.

To monitor and delete SOA composite application instances from the application home page:

1. Access this page through one of the following options:

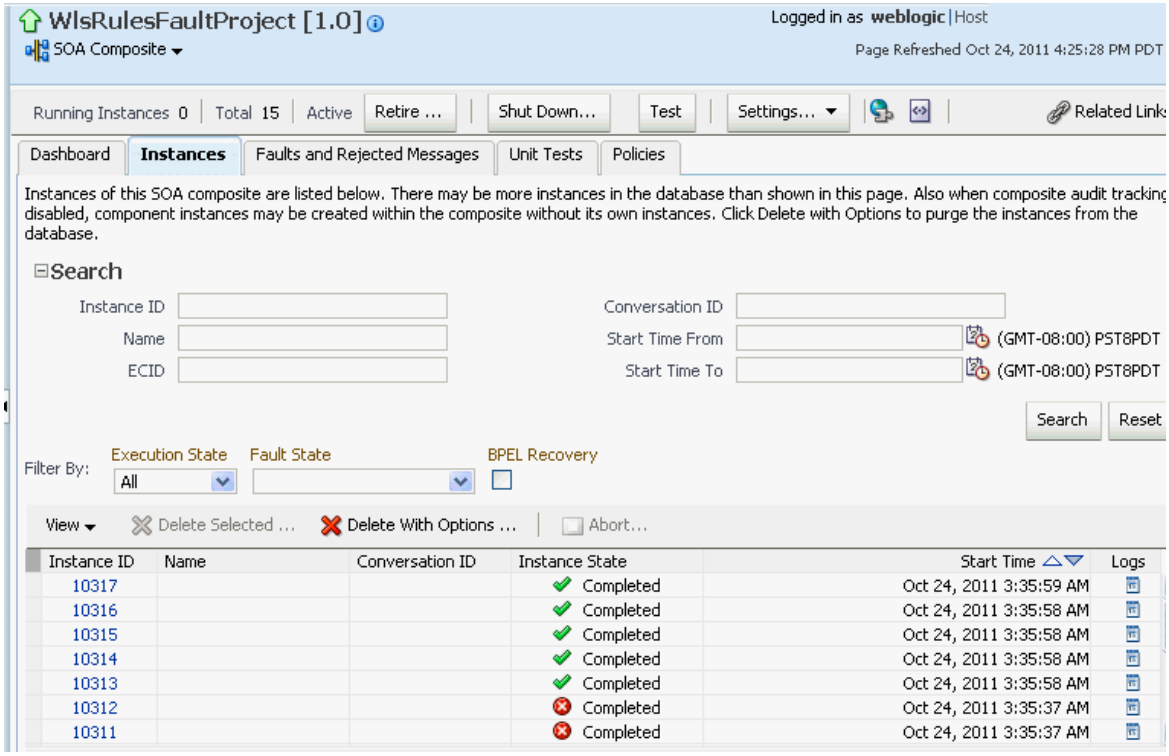
From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. Select a specific SOA composite application.	1. Under **soa-infra**, expand the partition. 2. Select a specific SOA composite application.

2. Click the **Instances** tab.

   The Instances page displays the following details:

   - A utility for searching for a specific instance by specifying criteria and clicking **Search**.
   - SOA composite application instance ID, name, conversation ID, most recent known state of each instance since the last data refresh of the page (for example, completed successfully, running, unknown, and so on), instance start time, and a log file describing any faults. A unique instance ID is created whenever a new instance of a SOA composite application is initiated either automatically by an external consumer of the application, or manually by an administrator from the Test Web Service page.

     If a **?** icon is displayed, the **Capture Composite Instance State** checkbox was not enabled on the SOA Infrastructure Common Properties dialog. Therefore, the instance state was not evaluated. Determining the composite instance state requires evaluating the states of the underlying component, Therefore, this can be disabled to improve performance.



Note: It is possible to generate orphaned service component instances. These instances are generated without any associated composite application instances. The orphaned component instances are generated under the following circumstances:

- The SOA Infrastructure audit level is set to **Off** or the composite audit level is set to **Off**. Even in such cases, the BPEL process service engine can generate instance data for the service components that are included in the SOA composite application.
- The SOA Infrastructure audit level is set to **Off**. However, the BPEL process or Oracle Mediator service engine audit level is set to a value other than **Off**.
- All the audit levels are set to **Off**, but some faults are generated in one of the service engines. In these cases, the component instance gets generated.

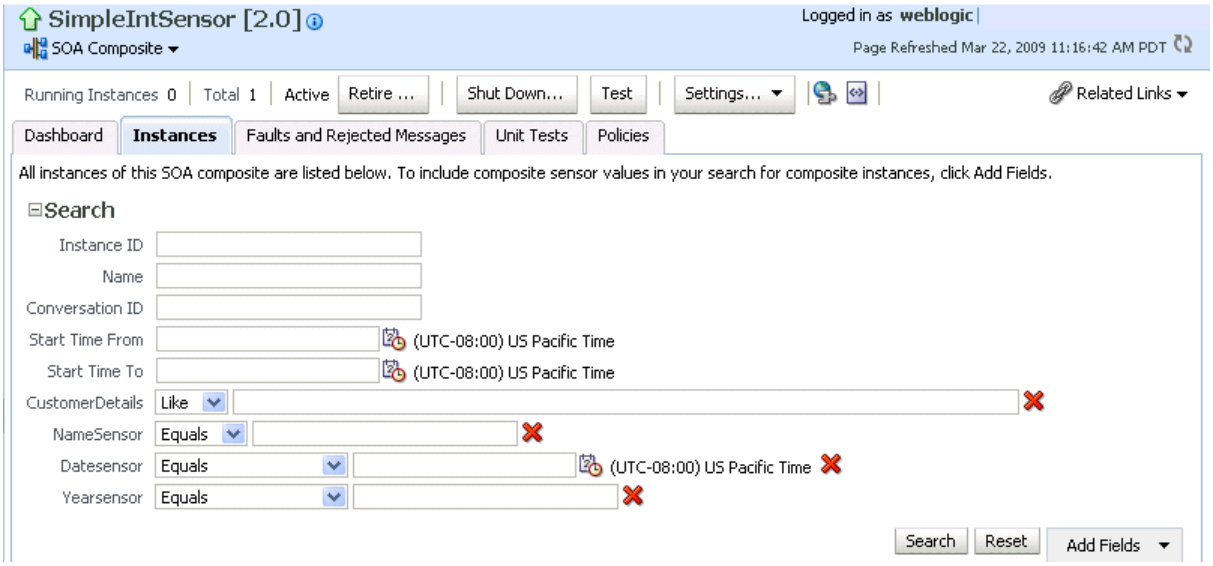
To delete orphaned instances or large numbers of instances, use the purge script described in Section 9.3, "Deleting Large Numbers of Instances with the Purge Scripts." Selecting the **Delete All Instance** options in the **Delete with Options** dialog does *not* delete orphaned component instances.

If composite sensors are included in your SOA composite application, the **Instances** tab has the following differences:

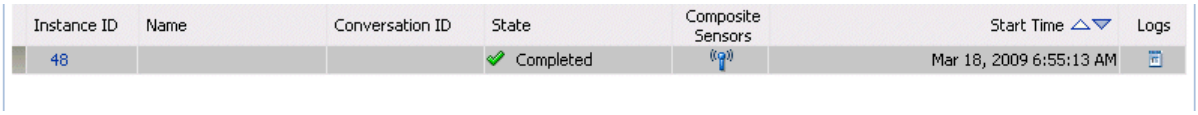
- The **Add Fields** button appears next to **Search** and **Reset** in the search utility. This button enables you to add sensor values to your search criteria.

- A **Composite Sensors** column appears in the **Instances** table. Click the sensor icon in that column to display the details about sensor values available in a given instance of the composite.

3. From the **Add Fields** list, select composite sensors to add to the search criteria. In this example, four have been selected (**CustomerDetails**, **NameSensor**, **Datesensor**, and **Yearsensor**).
4. Input specific values by which each sensor searches. Only the composite instances in which the sensor values match your specified criteria are returned.



The **Composite Sensors** column indicates that this SOA composite application includes composite sensors.



5. Click **Reset** to remove all composite sensor fields from the search criteria or click the **Remove** icon to the right of the field to remove an individual sensor.
6. Select a specific instance to delete by clicking a row in the **Instances** table. To select multiple instances, press Ctrl-Click or Shift-Click for the rows you want to select.
7. Select a specific action to perform.

Action	Description
Filter By	Specify criteria for displaying composite instance states: ▪ **Execution State** Filter the display of instances by execution state (running, completed, terminated, or stale). ▪ **Fault State** Filter the display of instances by fault state (with or without faults). You can further customize the faulted state by selecting to display faults requiring recovery or nonrecoverable faults. It you select **Stale** from the **Execute State** list, the **Fault State** list is disabled. ▪ **BPEL Recovery** Filter the display of instances by whether a recovery action is required. By default, this filter excludes all messages and instances created in the last five minutes, and displays the rest. You can control the number of minutes with the **excludeBpelMaxCreationTime** key of the **AuditConfig** property in the System MBean Browser. This property is available in the **More SOA Infra Advanced Configuration Properties** section of the SOA Infrastructure Common Properties page. For more information, see Section 3.1, "Configuring SOA Infrastructure Properties."
Delete Selected	Deletes the selected instance. After deleting an instance, instance details are no longer available for review.
Delete With Options	Prompts you to first specify criteria for deleting the selected instance directly from the database. Use this option to delete running, rolled back instances. However, this option does not delete the associated invoke messages that are awaiting recovery. As a result, there are orphaned messages pending in BPEL message recovery. To delete these messages, go to the Recovery page of the BPEL process service engine. ▪ **Common Delete Options**: Select a preset range of instances to delete from a list (for example, older than 24 hours). ▪ **Delete All Instances Of This Composite**: Select to delete all instances of the composite. This option deletes the rejected messages associated and all component, service, and reference instances associated with the composite, including those not associated with any composite instance ID. **Note:** If this composite has thousands of instances to delete, do not use this option. Instead, use the purge script described in Section 9.3, "Deleting Large Numbers of Instances with the Purge Scripts." ▪ **Delete All Instances That Match These Criteria**: Specify criteria for deleting instances, including the start and stop times, and instance state. Any selections you may have made in the Instances page (such as specifying and executing a search criteria) are ignored for this operation. To monitor the progress of instance deletion, you must check the log files. For information about log files, see Section 3.4, "Configuring Log Files."
Abort	Terminates the selected instance. However, instance details are still available for review.

8. From the **View** list, select **Columns** > **Partition** to display the partition in which the instance of the SOA composite application revision is contained.
9. From the **View** list, select **Columns** > **ECID** to display execution context IDs (ECIDs). An ECID enables you to track a message flow that crosses instances of different composites.
10. In the **Instances** table, perform the following additional tasks:
    a. In the **Instance ID** column, click a specific instance ID to show the message flow through the various service components and binding components. If an instance ID is listed as unavailable, you can click the **Unavailable** link for details.
    b. In the **State** column, if an instance state is marked as **Unknown**, click it to display more details.
    c. If the **Composite Sensors** column is available, click a sensor icon to display details about composite sensors included in the instance, such as name, location, and value.
    d. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

    Note: Multiple revisions of a SOA composite application that includes inbound JCA adapters are displayed as running. However, only the most recent revision (the default version) is considered active. All previous revisions are not considered active. This is because for inbound JCA adapters, there can only be one active revision of a SOA composite application at any time. The JCA adapter endpoints in all previous revisions are de-activated.

For more information, see the following sections:

- Section 1.2.3, "Introduction to SOA Composite Application Instances"
- Section 1.4.3.3, "Introduction to the Lifecycle State of SOA Composite Applications"
- Section 8.1, "Initiating a SOA Composite Application Test Instance"
- *Oracle Fusion Middleware Administrator's Guide* for details about viewing and searching log files

8.2.1 Mismatch Between the Number of SOA Composite Application Instances and Service Component Instances

The number of SOA composite application instances may not always match the number of service component instances displayed in Oracle Enterprise Manager Fusion Middleware Control.

A SOA composite application instance is first created when the composite is invoked. When the service components within the composite receive a subsequent invocation, a corresponding service component instance is created that refers to the composite instance ID previously created.

There can be scenarios under which the composite instance is created, but the underlining service component instance is not created. For example:

- The composite instance is created, but the invocation has not yet reached the service component due to a system failure.
- The composite instance is created, but the invocation fails payload validation and is rejected. In this case, invocation does not reach the underlining service components.

You can also have orphaned service component instances for which no SOA composite application instance has been created.

8.2.2 Instance States of Service Components and SOA Composite Applications

Assume you have a SOA composite application with multiple service components (for example, two BPEL process service components). If these service components are marked with the following instance states:

- Instance state of one BPEL process is marked as completed.
- Instance state of the other BPEL process is marked as faulted.

This results in the overall composite instance state being marked as faulted. This behavior differs from 11*g* Release 1 (11.1.1.2), in which the same scenario resulted in the overall composite instance state being marked as completed.

Assume you have a parent SOA composite application that calls a child SOA composite application, and a fault occurs in the child composite (and is handled by the parent composite). This results in the following instance states:

- The instance state of the child composite is marked as faulted.
- The instance state of the parent composite is marked as completed.

8.3 Monitoring and Deleting SOA Composite Application Instances at the SOA Infrastructure Level

Section 7.4, "Managing the State of Deployed SOA Composite Applications" described how to manage the lifecycle state of all instances of a specific SOA composite application. You can also monitor and delete any number of instances *across* all deployed SOA composite applications by using the Instances page of the SOA Infrastructure home page. This page lists all SOA composite application instances deployed to the SOA Infrastructure.

To monitor and delete SOA composite application instances at the SOA infrastructure level:

1. Access this page through one of the following options:

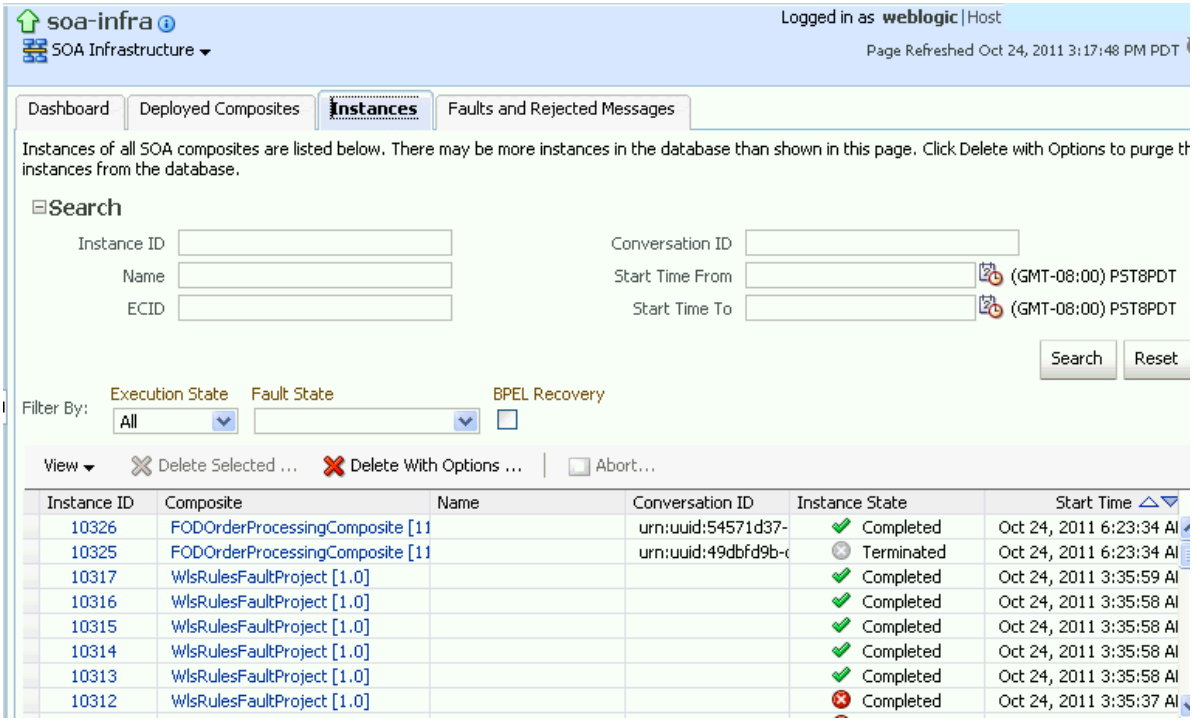
From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...	From the SOA Composite Menu...
1. Select **Home**.	1. Click **soa-infra**.	1. Select **SOA infrastructure**.

2. Click the **Instances** tab.

   The Instances page displays the following details:

   - A utility for searching for a specific instance by specifying criteria and clicking **Search**.

- All SOA composite application instances in the SOA Infrastructure, including instance and conversation IDs, composite name and revision, SOA composite application instance state, and instance start time.



You can also terminate and delete instances from this page.

3. Select a specific instance by clicking a row in the **Instances** table. To select multiple instances, press Ctrl-Click or Shift-Click for the rows you want to select.
4. Select a specific action to perform.

Action	Description
Filter By	Specify criteria for displaying composite instance states: ▪ **Execution State** Filter the display of instances by execution state (running, completed, terminated, or stale). ▪ **Fault State** Filter the display of instances by fault state (with or without faults). You can further customize the faulted states by selecting to display faults requiring recovery or nonrecoverable faults. It you select **Stale** from the **Execute State** list, the **Fault State** list is disabled. ▪ **BPEL Recovery** Filter the display of instances by whether a recovery action is required. By default, this filter excludes all messages and instances created in the last five minutes, and displays the rest. You can control the number of minutes with the **excludeBpelMaxCreationTime** key of the **AuditConfig** property in the System MBean Browser. This property is available in the **More SOA Infra Advanced Configuration Properties** section of the SOA Infrastructure Common Properties page. For more information, see Section 3.1, "Configuring SOA Infrastructure Properties."

Action	Description
Delete Selected	Deletes the selected instance.
Delete With Options	Prompts you to first specify criteria for deleting the selected instance directly from the database. Use this option to delete running, rolled back instances. However, this option does not delete the associated invoke messages that are awaiting recovery. As a result, there are orphaned messages pending in BPEL message recovery. To delete these messages, go to the Recovery page of the BPEL process service engine. ▪ **Common Delete Options**: Select a preset range of instances to delete from a list (for example, older than 24 hours). ▪ **Delete All Instances That Match These Criteria**: Specify criteria for deleting instances, including the start and stop times, and instance state. Any instance state selections you made at the top of the Instances page are ignored for this operation. To monitor the progress of instance deletion, you must check the log files. For information about log files, see Section 3.4, "Configuring Log Files." **Notes:** ▪ If this composite has thousands of instances to delete, do not use this option. Instead, use the purge script described in Section 9.3, "Deleting Large Numbers of Instances with the Purge Scripts." ▪ If you delete an instance with faults, those faults are no longer displayed in the Faults and Rejected Messages page. ▪ This option does not purge BPEL process service engine recoverable invoke and callback messages. To completely purge those messages, use the purge scripts described in Section 9.3, "Deleting Large Numbers of Instances with the Purge Scripts."
Abort	Terminates the selected instance. However, instance details are still available for review. **Note:** If you delete an instance with faults, those faults are no longer displayed in the Faults and Rejected Messages page. In addition, if a terminated instance (shown as aborted) had a fault, it is not added to the fault count.

5. In the instances table, perform the following tasks:
   a. From the **View** list, select **Columns** > **Partition** to display the partition in which the instance of the SOA composite application revision is contained.
   b. From the **View** list, select **Columns** > **ECID** to display execution context IDs (ECIDs). An ECID enables you to track a message flow that crosses instances of different composites.
   c. In the **Instance ID** column, click a specific instance ID to show the message flow through the various service components and binding components. If the instance ID is unavailable, the message flow cannot be accessed. However, you can still click the link for details.
   d. In the **Composite** column, click a specific SOA composite application to access its home page.
   e. In the **Instance State** column, click the **Recovery** icon to access the Faults and Rejected Messages page with faults filtered based on the composite instance ID. There can be multiple faults for a composite instance ID.

f. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

8.4 Recovering from SOA Composite Application Faults at the SOA Infrastructure Level

You can monitor and perform individual and bulk fault recoveries for BPEL process and Oracle Mediator service components across any number of SOA composite applications. For BPEL process faults to be identified as recoverable, there must be a fault policy defined that is bound to the fault (through the `fault-bindings.xml` file) and which triggers the action `ora-human-intervention`. However, without defining any fault policies, the fault takes its standard course as either a recoverable or nonrecoverable fault. Examples of performing both individual and bulk recovery are provided in this section. Human task service component or human workflow service engine faults are recovered from Oracle BPM Worklist.

You can also perform a manual recovery of undelivered BPEL process invoke or callback messages. For more information, see Section 14.4, "Performing BPEL Process Service Engine Message Recovery."

To recover from SOA composite application faults at the SOA Infrastructure level:

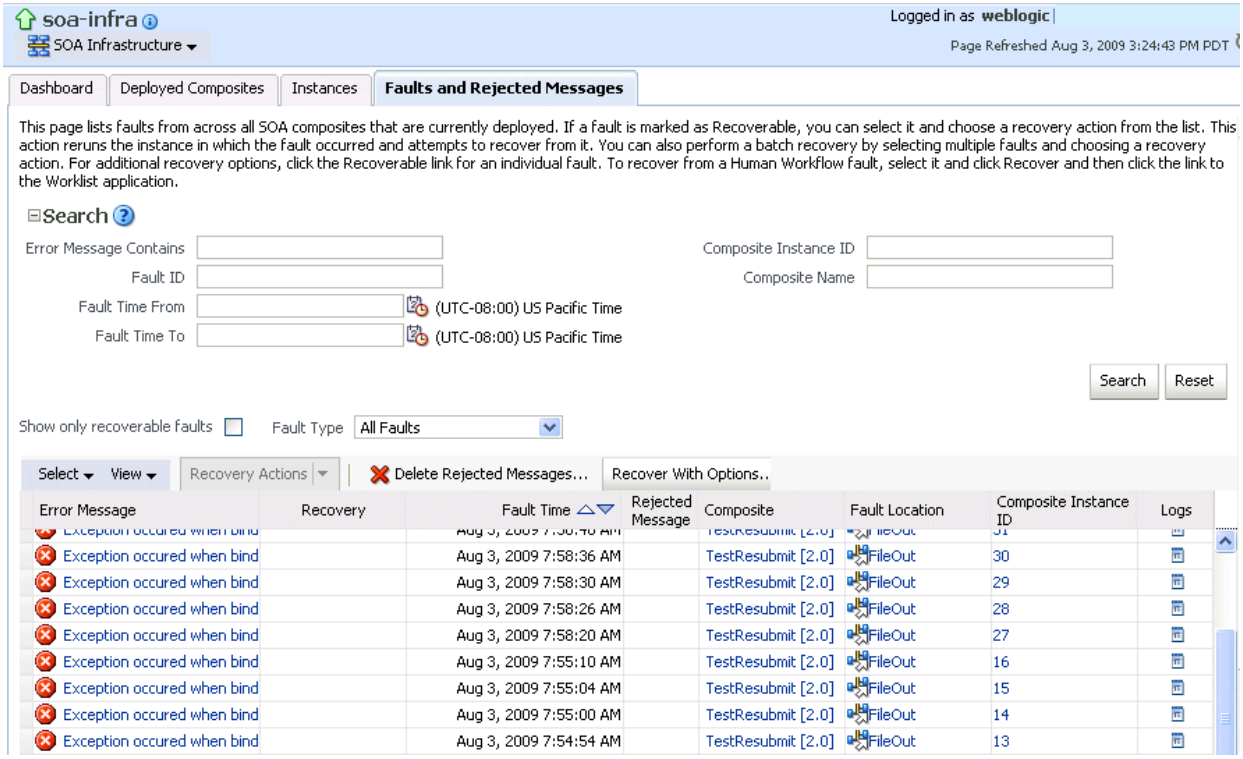
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...	From the SOA Composite Menu...
1. Select **Home**.	1. Click **soa-infra**.	1. Select **SOA Infrastructure**.

2. Click the **Faults and Rejected Messages** tab.

   The Faults and Rejected Messages page displays the following details for all SOA composite application faults:

   - A utility for searching for a specific fault by specifying criteria and clicking **Search**. Click the **Help** icon for details.
   - Options for selecting instance recovery actions (for example, retry, abort, replay, and others), deleting rejected messages, and performing bulk message recovery.
   - Faults and rejected messages, including the error message, whether you can recover from the fault, the time of the fault, if the fault message is classified as a rejected message (if so, a checkmark is displayed), the SOA composite application in which the fault occurred, the fault location, the instance ID, and a link to log files describing the fault.



Note: You cannot search for human workflow error messages by entering details in the **Error Message Contains** field because these faults do not persist in the dehydration store.

Faults identified as recoverable can be recovered.

3. Select faults for recovery using one of the following options. Fault recovery selection at the SOA Infrastructure level matches the SOA composite application level and BPEL process and Oracle Mediator service component levels.

For...	Then...
Single fault recovery	There are three options from which to choose for single-fault recovery: 1. Click the row of the fault that has been identified as recoverable. With the row highlighted, select a specific action from the **Recovery Action** list, as described in Step 4. 2. In the **Recovery** column, click the **Recover** link to access the Faults page of the instance audit trail to perform fault recovery. 3. In the **Error Message** column, click the message of a fault that has been identified as recoverable. This displays complete fault details, including the fault ID, fault time, fault location, fault type, and error message text. A **Recover Now** option is displayed for recoverable faults. Click **Recover Now** to access the Faults page of the instance audit trail to perform fault recovery.

For...	Then...
Bulk fault recovery	There are two options from which to choose for bulk-fault recovery: **1.** Use Shift+Click or Control+Click to select specific faults in the rows. or **2.** From the **Select** menu, choose **Select All Recoverable**. Then use Shift+Click or Control+Click to deselect the faults to *not* include in the recovery operation. Then: **3.** Select an action from the **Recovery Action** list, as described in Step 4. **Note:** Only the actions applicable to all selected faults are available.
Recovery of all faults	**1.** From the **Select** menu, choose **Select All Recoverable**. **2.** Select an action from the **Recovery Action** list, as described in Step 4. **Note:** Only the actions applicable to all selected faults are available.

4. Select an action from the **Recovery Action** list.

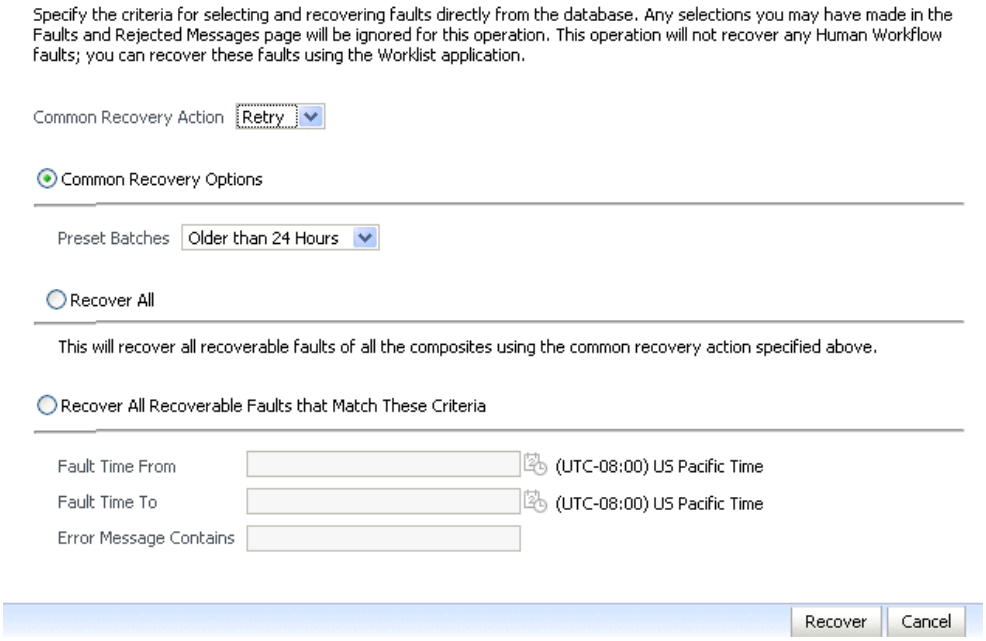
Action	Description	Action is Available for...
Retry	Retries the instance directly. An example of a scenario in which to use this recovery action is when the fault occurred because the service provider was not reachable due to a network error. The network error is now resolved.	BPEL process and Oracle Mediator
Abort	Terminates the entire instance.	BPEL process and Oracle Mediator
Replay	Replays the entire scope again in which the fault occurred.	BPEL process
Rethrow	Rethrows the current fault. BPEL fault handlers (catch branches) are used to handle the fault. By default, all exceptions are caught by the fault management framework unless an explicit rethrow fault policy is provided.	BPEL process
Continue	Ignores the fault and continues processing (marks the faulted activity as a success).	BPEL process

Note: In most cases, fault policy actions are automatically executed. The only exception is if you defined a fault policy that uses the action `ora-human-intervention`. This action creates a recoverable fault that can be recovered from Oracle Enterprise Manager Fusion Middleware Control.

5. If you want to delete rejected messages for all composites in the SOA Infrastructure, see Section 8.6, "Deleting Rejected Messages at the SOA Infrastructure Level."

6. If you want to perform a bulk recovery of messages, click **Recover with Options**.

This displays the Recover with Options dialog for specifying criteria for recovering BPEL and Oracle Mediator messages of all composites directly from the database. Human workflow messages can be recovered manually from Oracle BPM Worklist. Business event and business rule messages cannot be recovered.



7. Specify criteria. **Retry** and **Abort** are the only recovery actions permitted.

Note: For bulk fault recovery at the SOA Infrastructure level, a check of the state of composites cannot be performed. If the state of a composite is set to off, a recovery of its faults cannot be performed. However, no error or warning message is displayed. Upon submission of the bulk fault recovery request, the server checks if the originating composite's state is set to off. That fact is then noted in the log, and the fault is skipped.

You are also not notified when a fault has been skipped during recovery for any other reason (for example, an unsupported service engine, an unrecoverable fault, and so on).

---

8. Click **Recover**. Depending upon the number of messages, recovery can take some time.
9. Perform the following additional tasks from within the faults table:
   a. From the **View** list, select **Columns** > **Fault ID** to display the fault IDs for each error message. The fault ID is automatically generated and uniquely identifies a fault. The fault ID is also displayed when you click an error message.
   b. In the **Composite** column, click a specific SOA composite application to access its home page.
   c. In the **Fault Location** column, click a specific location to access the faults page for the location of the fault. The location can be a service, service component, or reference.

d. In the **Composite Instance ID** column, click a specific ID to access the flow trace of the instance.

e. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

10. See the following sections for examples of single and bulk fault recovery with BPEL processes and Oracle Mediator.

- Section 8.4.1, "Examples of Fault Recovery for BPEL Processes"
- Section 8.4.2, "Examples of Fault Recovery for BPMN Processes"
- Section 8.4.3, "Examples of Fault Recovery for Oracle Mediator"

For more information about concepts and instructions on designing a fault policy, see the following documentation:

- Section 1.4.3.1, "Introduction to Fault Recovery"
- Section "Using the Fault Management Framework" of *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

8.4.1 Examples of Fault Recovery for BPEL Processes

This section provides examples of how to define a fault policy that enables human intervention on a BPEL process fault and perform single and bulk fault recovery on a BPEL process service component.

- Section 8.4.1.1, "Example: Single Fault Recovery for BPEL Processes"
- Section 8.4.1.2, "Example: Bulk Fault Recovery for BPEL Processes"

In this example, you define a fault policy by specifying that a fault can be manually recovered through human intervention. If an invalid social security number is submitted from a loan broker BPEL process to a credit rating service, the credit rating service returns a negative credit fault. This human intervention action is defined with the `ora-human-intervention` action in the `fault-policies.xml` file. Without fault policies, BPEL instances do not generate recoverable faults (instead they are nonrecoverable); the `ora-human-intervention` action makes the fault recoverable.

```
<faultPolicies xmlns="http://schemas.oracle.com/bpel/faultpolicy">
<faultPolicy version="2.0.1"
           id="CRM_ServiceFaults"
           xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://schemas.oracle.com/bpel/faultpolicy"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <Conditions>
               <faultName xmlns:credit="http://services.otn.com"
               name="credit:NegativeCredit">
               <!-- get this fault when SSN starts with 0-->
                  <condition>
                     <test>$fault.payload="Bankruptcy Report"</test>
                     <action ref="ora-human-intervention"/>
                  </condition>
               </faultName>
            </Conditions>
</faultPolicy>
</faultPolicies>
```

The `fault-bindings.xml` file associates the fault policies defined in the `fault-policies.xml` file with the `CRM_ServiceFaults` composite application.

```
<faultPolicyBindings version="2.0.1"
 xmlns="http://schemas.oracle.com/bpel/faultpolicy"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <composite faultPolicy="CRM_ServiceFaults"/>
</faultPolicyBindings>
```

Because human intervention is defined as an action, you perform BPEL process fault recovery in Oracle Enterprise Manager Fusion Middleware Control.

For more information about creating and designing `fault-policies.xml` and `fault-bindings.xml` files, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

For information about BPEL process message recovery, see Section 14.4, "Performing BPEL Process Service Engine Message Recovery."

8.4.1.1 Example: Single Fault Recovery for BPEL Processes

This example assumes the following:

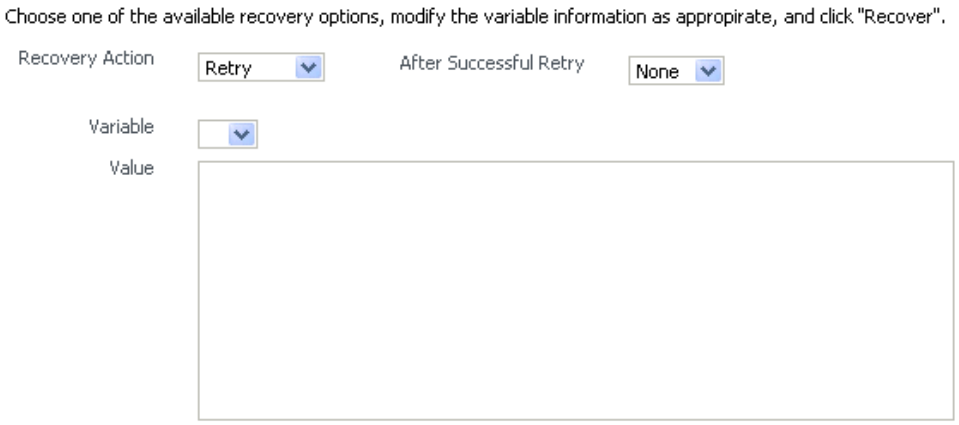
- An instance was initiated on the Test Web Service page shown in Section 8.1, "Initiating a SOA Composite Application Test Instance."
- An invalid social security number that begins with `0` was entered.

To perform single fault recovery for BPEL processes:

1. From the **SOA Infrastructure** menu, select **Home**.
2. Click the **Faults and Rejected Messages** tab.
3. In the faults table, locate the fault that has been identified as recoverable. You can use the search utility to locate the specific fault.
4. In the **Recovery** column, click **Recover**. If you first want to see details about the fault, click the error message. Then, click **Recover Now.**

   The Faults page for that BPEL process instance is displayed.
5. In the Recovery column, click **Recoverable**.

   The page refreshes to display the fault recovery section at the bottom of the page.



6. From the **Recovery Action** list, select **Retry**.
7. Select **None** from the **Chain Action Upon Successful Retry** list. This list enables you to select Java callout recovery actions. For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

8. Select a variable from the **Variable** list. The content of this variable is displayed in the **Value** field. For this example, the variable **crInput** is selected. This variable is used in an invoke activity and contains an incorrect social security number value.
9. Enter the correct value in the **Value** field. For this example, the social security number is edited to begin with `1`:

```
<ssn xmlns="http://service.otn.com">123456789</ssn>
```

10. Click **Set Value**, and click **Yes** when prompted to continue.
11. Click **Recover** to recover from the fault, and then click **Yes** when prompted to continue.

    The page refreshes to indicate that no faults occurred.

8.4.1.2 Example: Bulk Fault Recovery for BPEL Processes

For the social security number example, selecting **Retry** is not an option for performing a bulk recovery, because the value for the social security number is incorrect and requires correction. An example of performing a bulk recovery with the **Retry** option is if the social security number is correct, but the system providing the credit rating service was temporarily unavailable and caused a composite reference fault. This prevents the messages from being delivered. Once the credit rating service is available again, selecting **Retry** attempts the invocation to the credit rating service through the composite reference again.

To perform bulk fault recovery for BPEL processes:

1. Perform Step 1 and Step 2 of Section 8.4.1.1, "Example: Single Fault Recovery for BPEL Processes."
2. In the search utility, enter criteria based on known fault parameters (for example, the time range, composite name, component type (BPEL process), and so on).
3. If the search returns too many results, limit it by selecting the **Show only recoverable faults** checkbox.
4. From the Select list, choose **Select All Recoverable**.
5. From the **Recovery Action** list, select **Abort**.

   All selected faults are manually terminated.

8.4.2 Examples of Fault Recovery for BPMN Processes

This section provides examples of how to define a fault policy that enables human intervention on a BPMN process fault and perform single and bulk fault recovery on a BPMN process service component.

> **Note:** When a multi-instance process has met the conditions for its completion, it raises a nonrecoverable system fault (to cancel remaining instances). Although this fault appears in the Oracle Enterprise Manager Fusion Middleware Control, you do not need to take any action. It appears simply to notify you that the multi-instance process was finalized because the condition was completed.

- Section 8.4.2.1, "Example: Single Fault Recovery for BPMN Processes"
- Section 8.4.2.2, "Example: Bulk Fault Recovery for BPMN Processes"

In this example, you define a fault policy specifying that a fault be manually recovered through human intervention. If an invalid social security number is submitted from a loan broker BPMN process to a credit rating service, the credit rating service returns a negative credit fault. This human intervention action is defined with the `ora-human-intervention` action in the `fault-policies.xml` file. Without fault policies, BPMN instances do not generate recoverable faults (instead they are nonrecoverable); the `ora-human-intervention` action makes the fault recoverable.

```
<faultPolicies xmlns="http://schemas.oracle.com/bpmn/faultpolicy">
<faultPolicy version="2.0.1"
           id="CRM_ServiceFaults"
           xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://schemas.oracle.com/bpmn/faultpolicy"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <Conditions>
               <faultName xmlns:credit="http://services.otn.com"
               name="credit:NegativeCredit">
               <!-- get this fault when SSN starts with 0-->
                  <condition>
                     <test>$fault.payload="Bankruptcy Report"</test>
                     <action ref="ora-human-intervention"/>
                  </condition>
               </faultName>
            </Conditions>
</faultPolicy>
</faultPolicies>
```

The `fault-bindings.xml` file associates the fault policies defined in the `fault-policies.xml` file with the `CRM_ServiceFaults` composite.

```
<faultPolicyBindings version="2.0.1"
 xmlns="http://schemas.oracle.com/bpmn/faultpolicy"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <composite faultPolicy="CRM_ServiceFaults"/>
</faultPolicyBindings>
```

Because human intervention is defined as an action, you perform BPMN process fault recovery in Oracle Enterprise Manager Fusion Middleware Control.

For more information about creating and designing `fault-policies.xml` and `fault-bindings.xml` files, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

8.4.2.1 Example: Single Fault Recovery for BPMN Processes

This example assumes the following:

- An instance was initiated on the Test Web Service page shown in Section 8.1, "Initiating a SOA Composite Application Test Instance."
- An invalid social security number that begins with `0` was entered.

To perform single fault recovery for BPMN processes:

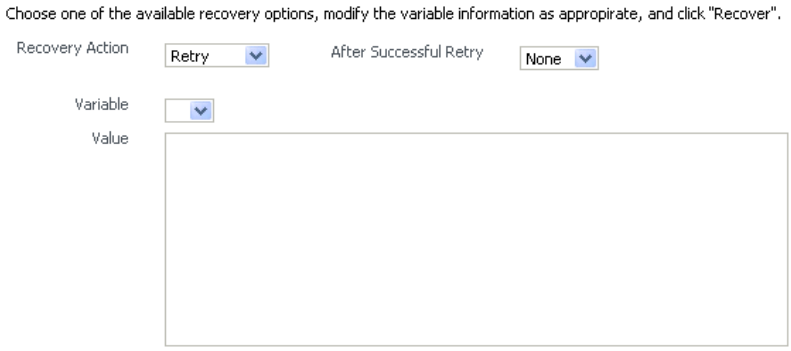
1. From the **SOA Infrastructure** menu, select **Home**.
2. Click the **Faults and Rejected Messages** tab.
3. In the faults table, locate the fault that has been identified as recoverable. You can use the search utility to locate the specific fault.

4. In the **Recovery** column, click **Recover**. If you first want to see details about the fault, click the error message. Then, click **Recover Now.**

   The Faults page for that BPMN process instance is displayed.

5. In the Recovery column, click **Recoverable**.

   The page refreshes to display the fault recovery section at the bottom of the page.



6. From the **Recovery Action** list, select **Retry**.
7. From the **Chain Action Upon Successful Retry** list, select **None**. This list enables you to select Java callout recovery actions. For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.
8. From the **Variable** list, select a variable. The content of this variable is displayed in the **Value** field. For this example, the variable **crInput** is selected. This variable is used in an invoke activity and contains an incorrect social security number value.
9. In the **Value** field, enter the correct value. For this example, the social security number is edited to begin with `1`:

   ```
   <ssn xmlns="http://service.otn.com">123456789</ssn>
   ```

10. Click **Set Value**, and click **Yes** when prompted to continue.
11. Click **Recover** to recover from the fault, then click **Yes** when prompted to continue.

    The page refreshes to indicate that no faults occurred.

8.4.2.2 Example: Bulk Fault Recovery for BPMN Processes

For the social security number example, selecting **Retry** is not an option for performing a bulk recovery because the value for the social security number is incorrect and requires correction. An example of performing a bulk recovery with the **Retry** option is if the social security number is correct, but the system providing the credit rating service was temporarily unavailable and caused a composite reference fault. This prevents the messages from being delivered. Once the credit rating service is available again, selecting **Retry** re-attempts the invocation to the credit rating service through the composite reference.

To perform bulk fault recovery for BPMN processes:

1. Perform Steps 1 through 2 of Section 8.4.2.1, "Example: Single Fault Recovery for BPMN Processes."
2. In the search utility, enter criteria based on known fault parameters (for example, the time range, composite name, component type (BPMN process), and so on).

3. If the search returns too many results, limit it by selecting the **Show only recoverable faults** checkbox.
4. From the Select list, choose **Select All Recoverable**.
5. From the **Recovery Action** list, select **Abort**.

   All selected faults are manually terminated.

8.4.3 Examples of Fault Recovery for Oracle Mediator

This section provides an example of how to perform single and bulk fault recovery on an Oracle Mediator service component.

- Section 8.4.3.1, "Example: Single Fault Recovery for Oracle Mediator"
- Section 8.4.3.2, "Example: Bulk Fault Recovery for Oracle Mediator"

In this example, a service binding component for an inbound Siebel adapter submits a payload message to Oracle Mediator for transformation. The processed payload message is then delivered to a reference binding component for an outbound file adapter. However, the outbound directory into which to write the payload message is not configured with write permissions. This causes a fault to occur. The fault policy defined during design time specifies that the fault be manually recovered through human intervention. Three retries are attempted, as defined with the `retryCount` attribute. The condition and action are defined as follows in the `fault-policies.xml` file.

Recoverable Oracle Mediator faults do not require a fault policy (though it is one way to make faults recoverable, as described through an `ora-human-intervention` action). Any parallel routing rule that receives a remote fault from the outbound endpoint also creates a recoverable fault (in this specific example, the fault policy is not required if the Oracle Mediator uses a parallel routing rule to invoke the outbound file adapter).

```
<faultPolicies xmlns="http://schemas.oracle.com/bpel/faultpolicy">
<faultPolicy version="2.0.1"
           id="ConnectionFaults"
           xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://schemas.oracle.com/bpel/faultpolicy"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
              <Conditions>
                <faultName xmlns:medns="http://schemas.oracle.com/mediator/faults"
                name="medns:mediatorFault">
                   <condition>
                      <test>contains($fault.mediatorErrorCode, "TYPE_FATAL_
                         MESH")</test>
                      <action ref="ora-retry"/>
                   </condition>
                </faultName>
              </Conditions>
. . .
. . .
      <Action id="ora-retry">
        <retry>
          <retryCount>3</retryCount>
          <retryInterval>5</retryInterval>
          <retryFailureAction ref="ora-human-intervention"/>
          <retrySuccessAction ref="ora-terminate"/>
        </retry>
```

```
      </Action>
   </Actions>
</faultPolicy>
</faultPolicies>
```

Processing is set to retry `3` times before terminating.

The fault policies are associated with the `ConnectionFaults` composite application in the `fault-bindings.xml` file:

```
<faultPolicyBindings version="2.0.1" xmlns="http://schemas.oracle.com/bpel/fault
policy" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <composite faultPolicy="ConnectionFaults"/>
</faultPolicyBindings>
```

8.4.3.1 Example: Single Fault Recovery for Oracle Mediator

For this example, the `sap` output directory is made read-only. An inbound file adapter retrieves the `sender.xml` file from the `siebel` directory and the message is routed through Oracle Mediator to an outbound file adapter reference for placing a file in the `sap` directory.

To perform single fault recovery for Oracle Mediator:

1. Change the directory permissions at the operating system command prompt.

   ```
   chmod 000 sap
   cp sender.xml siebel/
   ```

2. From the **SOA Infrastructure** menu, select **Home**.
3. Click the **Faults and Rejected Messages** tab.

   Three faults appear, based on three retries being attempted. In this case, you see three retries only because the fault policy on the Oracle Mediator interaction with the outbound file adapter defines three retries. Without the fault policy, there is only one fault (no automated retries).

4. Click the specific instance ID in the **Composite Instance ID** column.

   The Flow Trace appears. The faults table at the top of the page displays the fault messages. If you want to see where the faulted Oracle Mediator instance is located in the overall message flow, select the fault in the faults table. This highlights the associated instance in the trace table. You can then click the instance to access its audit trail to see more details about the faulted flow.

   > **Note:** Steps 4 through 10 represent one way to recover this single fault. The fault can also be recovered directly from the Oracle Mediator faults page through the **Recovery Action** list.

5. Locate the Oracle Mediator component instance fault you want to recover in the **Faults** table and click **Recover** in the **Recovery** column.
6. Select **Sender** from the **Payload Part** list.

   The payload is automatically displayed in the **Payload** field. If necessary, payload modifications can be performed in this field. For this example, payload modification is not necessary.

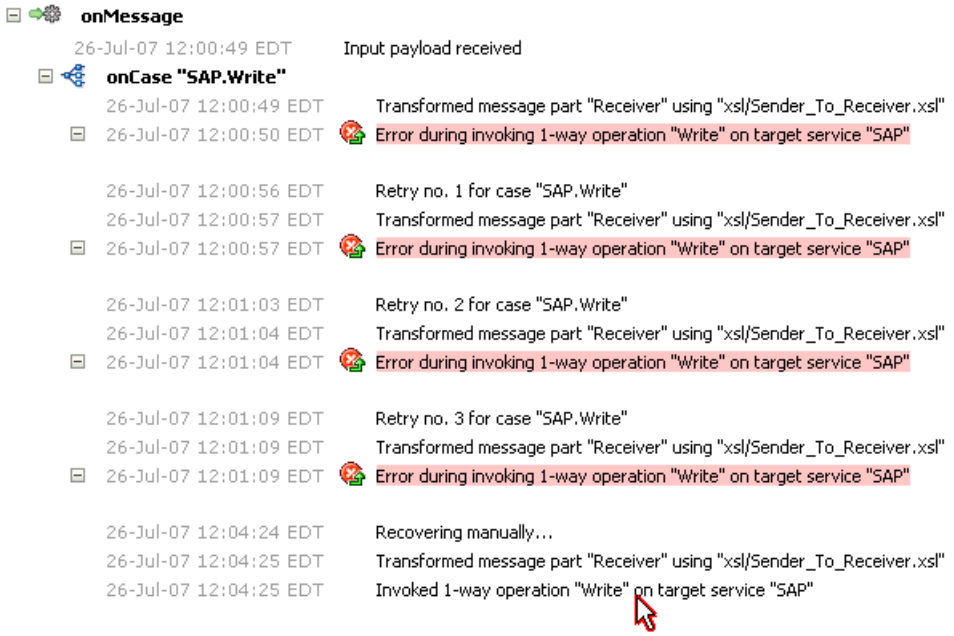
7. Change the `sap` directory to be writable at the operating system command prompt.

```
chmod 777 sap
```

8. Return to the **Faults** tab and click the **Refresh** icon in the upper right corner of the page.
9. Click **Retry**.
10. Click **Yes** when prompted to resubmit the selected fault for recovery.

    The page refreshes to indicate that no faults occurred.
11. Click the **Audit Trail** tab.

    The final message indicates that manual recovery was successful and the message payload was written to the `sap` directory.



8.4.3.2 Example: Bulk Fault Recovery for Oracle Mediator

Assume the `sap` directory to which to write the `sender.xml` payload message is again configured with read-only permissions at the operating system command prompt. Three copies of the `sender.xml` file are placed in the `siebel` directory of the service binding component for the inbound Siebel adapter. This creates three instances.

```
chmod 000 sap
cp sender.xml siebel/
cp sender.xml siebel/
cp sender.xml siebel/
```

To perform bulk fault recovery for Oracle Mediator:

1. Change the `sap` directory to be writable.
2. From the **SOA Infrastructure** menu, select **Home**.
3. Click the **Faults and Rejected Messages** tab.
4. In the search utility, enter criteria based on known fault parameters (for example, the time range, composite name, and so on).

5. If the search returns too many results, limit it by selecting the **Show only recoverable faults** checkbox.
6. Change the `sap` directory to be writable at the operating system command prompt.

   ```
   chmod 777 sap
   ```

7. Select all the faults to be recovered.
8. Select **Retry** from the **Recovery Action** list.
9. Select **Yes** when prompted to perform fault recovery.
10. Click the **Audit Trail** tab.

    The final message indicates that manual recovery was successful and the message payload was successfully written to the `sap` directory.

8.5 Recovering from SOA Composite Application Faults in the Application Home Page

You can monitor and perform individual and bulk fault recoveries in a SOA composite application. For BPEL process faults to be identified as recoverable, there must be a fault policy defined that is bound to the fault (through the `fault-bindings.xml` file) and which triggers the action `ora-human-intervention`. However, without defining any fault policies, the fault takes its standard course as either a recoverable or nonrecoverable fault. Human workflow faults can also be recovered, but not directly from Oracle Enterprise Manager Fusion Middleware Control. Instead, the audit trail provides a link to the Oracle BPM Worklist, from which the fault can be addressed.

To recover from SOA composite application faults in the application home page:

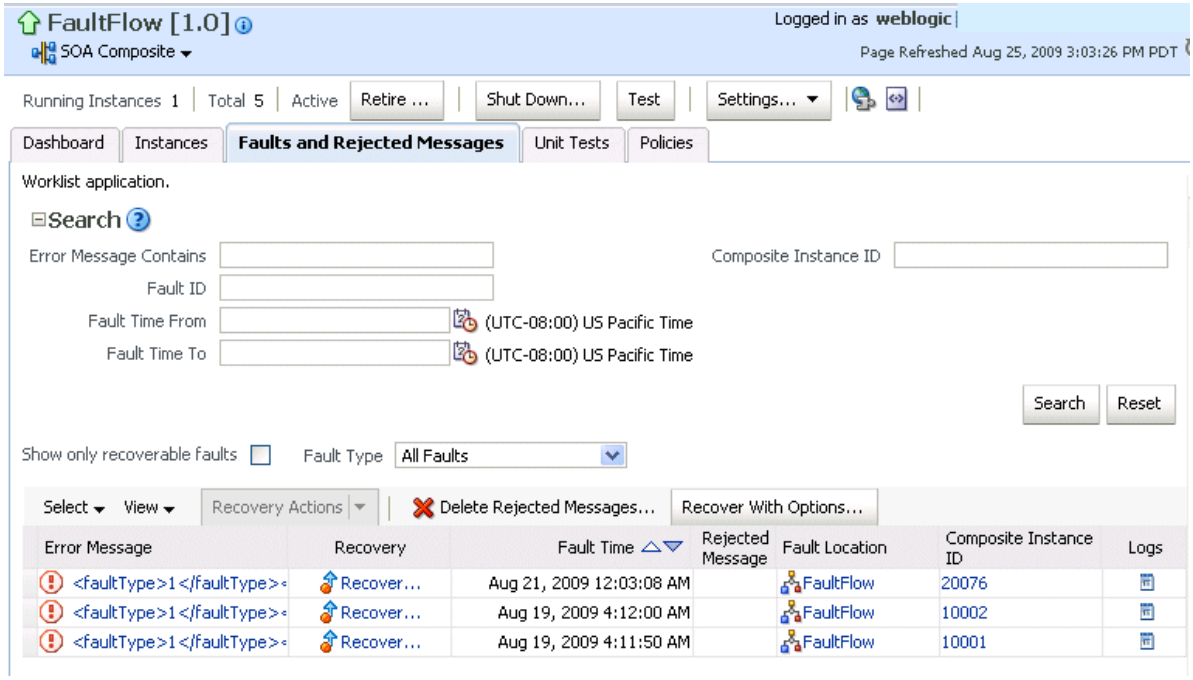
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**.	1. Under **soa-infra**, expand the partition.
2. Select **Deployed Composites**.	2. Select a specific SOA composite application.
3. In the **Composite** section, select a specific SOA composite application.	

2. Click the **Faults and Rejected Messages** tab.

   The Faults and Rejected Messages page displays the following details for the selected SOA composite application:

   - A utility for searching for a specific fault by specifying criteria and clicking **Search**. Click the **Help** icon for details.
   - Options for selecting instance recovery actions (for example, retry, abort, replay, and others), deleting rejected messages, and performing bulk message recovery.
   - Faults and rejected messages in SOA composite application instances, including the error message, whether you can recover from the fault, the time of the fault, if the fault message is classified as a rejected message (if so, a checkmark is displayed), the fault location, the composite instance ID, and links to log files that describe the fault.



Note: You cannot search for human workflow error messages by entering details in the **Error Message Contains** field because these faults do not persist in the dehydration store.

Faults identified as recoverable can be recovered.

3. Select faults for recovery. As with fault recovery at the SOA Infrastructure level and BPEL process and Oracle Mediator service component levels, you can perform single fault recovery, bulk fault recovery, and recovery of all faults. See Step 3 of Section 8.4, "Recovering from SOA Composite Application Faults at the SOA Infrastructure Level" for instructions on selecting faults to perform these types of recovery.

4. Select an action from the **Recovery Action** list.

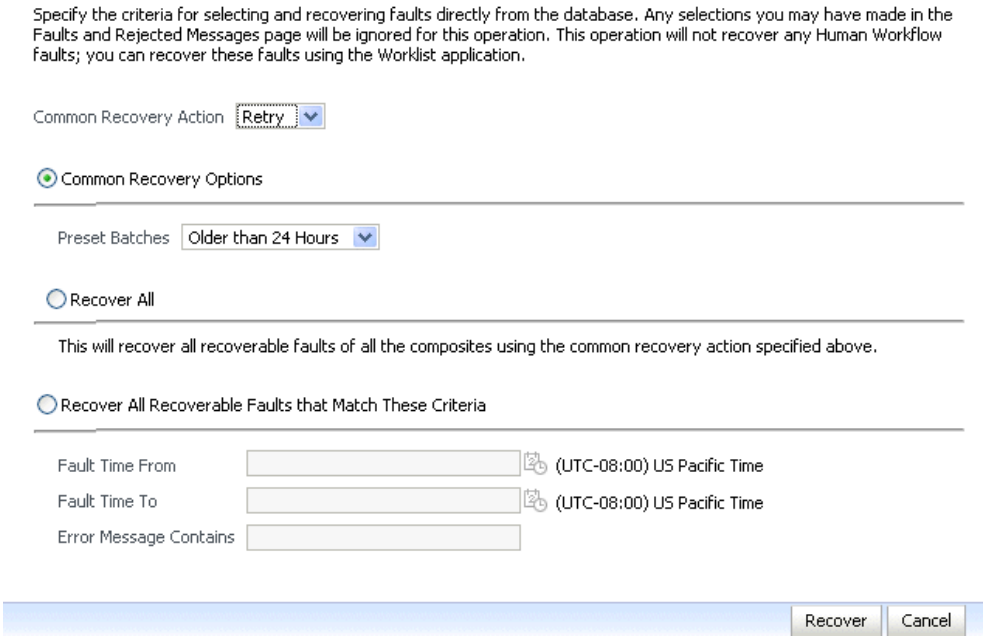
Action	Description	Action is Available for...
Retry	Retries the instance directly. An example of a scenario in which to use this recovery action is when the fault occurred because the service provider was not reachable due to a network error. The network error is now resolved.	BPEL process and Oracle Mediator
Abort	Terminates the entire instance.	BPEL process and Oracle Mediator
Replay	Replays the entire scope again in which the fault occurred.	BPEL process

Action	Description	Action is Available for...
Rethrow	Rethrows the current fault. BPEL fault handlers (catch branches) are used to handle the fault. By default, all exceptions are caught by the fault management framework unless an explicit rethrow fault policy is provided.	BPEL process
Continue	Ignores the fault and continues processing (marks the faulted activity as a success).	BPEL process

Note: In most cases, fault policy actions are automatically executed. The only exception is if you defined a fault policy that uses the action `ora-human-intervention`. This action creates a recoverable fault that can be recovered from Oracle Enterprise Manager Fusion Middleware Control.

5. If you want to delete rejected messages for the current SOA composite application, see Section 8.7, "Deleting Rejected Messages from the Application Home Page."
6. If you want to perform a bulk recovery of messages, click **Recover with Options**.

   This displays the Recover with Options dialog for specifying criteria for recovering BPEL and Oracle Mediator messages of the current composite directly from the database. Human workflow messages can be recovered manually from Oracle BPM Worklist. Business event and business rule messages cannot be recovered.



7. Specify criteria. **Retry** and **Abort** are the only recovery actions permitted.

> **Note:** For bulk fault recovery at the SOA composite application level, a check of the state of the composite is performed. If the state of the composite is set to off, a message is displayed warning you that a recovery cannot be performed.
>
> You are not notified when a fault has been skipped during recovery for any reason (for example, an unsupported service engine, an unrecoverable fault, and so on).

8. Click **Recover**. Depending upon the number of messages, recovery can take some time.
9. Perform the following additional monitoring tasks from within the faults table:
   a. From the **View** list, select **Columns** > **Fault ID** to display the fault IDs for each error message. The fault ID is automatically generated and uniquely identifies a fault. The fault ID is also displayed when you click an error message.
   b. In the **Fault Location** column, click a specific location to access the faults page for the location of the fault. The location can be a service, component, or reference.
   c. In the **Component Instance ID** column, click a specific service component ID to access task details about the instance (for example, the current state of a task). Rejected messages do not have a component instance ID.
   d. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

For more information, see the following sections:

- Section 1.2.4, "Introduction to Service Components and Service Component Instances"
- Section 1.4.3.1, "Introduction to Fault Recovery"
- Section 8.4.1, "Examples of Fault Recovery for BPEL Processes"
- Section 8.4.3, "Examples of Fault Recovery for Oracle Mediator"

8.6 Deleting Rejected Messages at the SOA Infrastructure Level

You can delete rejected messages for all composites in the SOA Infrastructure directly from the database by specifying a criteria in the Delete: Rejected Messages dialog.

To delete rejected messages for all composites at the SOA Infrastructure level:

1. Access this page through one of the following options:

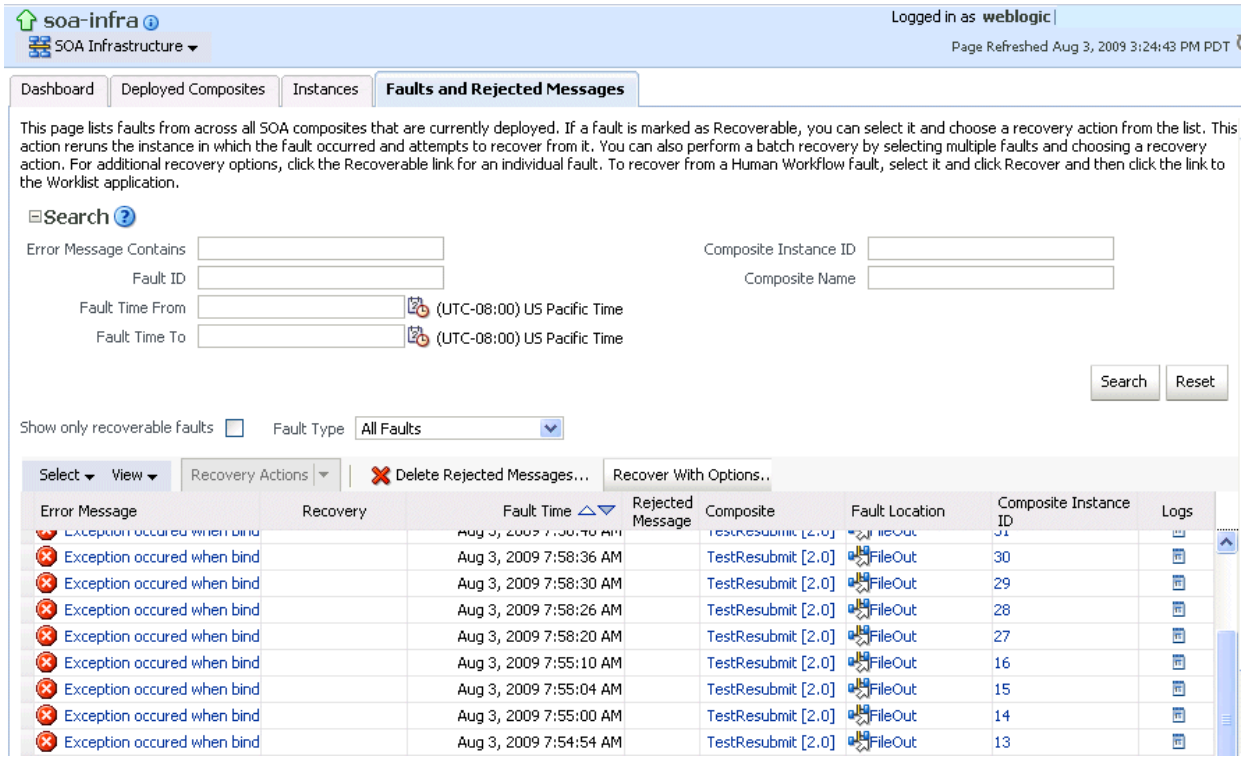
From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...	From the SOA Composite Menu...
1. Select **Home**.	1. Click **soa-infra**.	1. Select **SOA Infrastructure**.

2. Click the **Faults and Rejected Messages** tab.

   The Faults and Rejected Messages page displays the following details for all SOA composite application faults:

   - A utility for searching for a specific fault by specifying criteria and clicking **Search**. Click the **Help** icon for details.

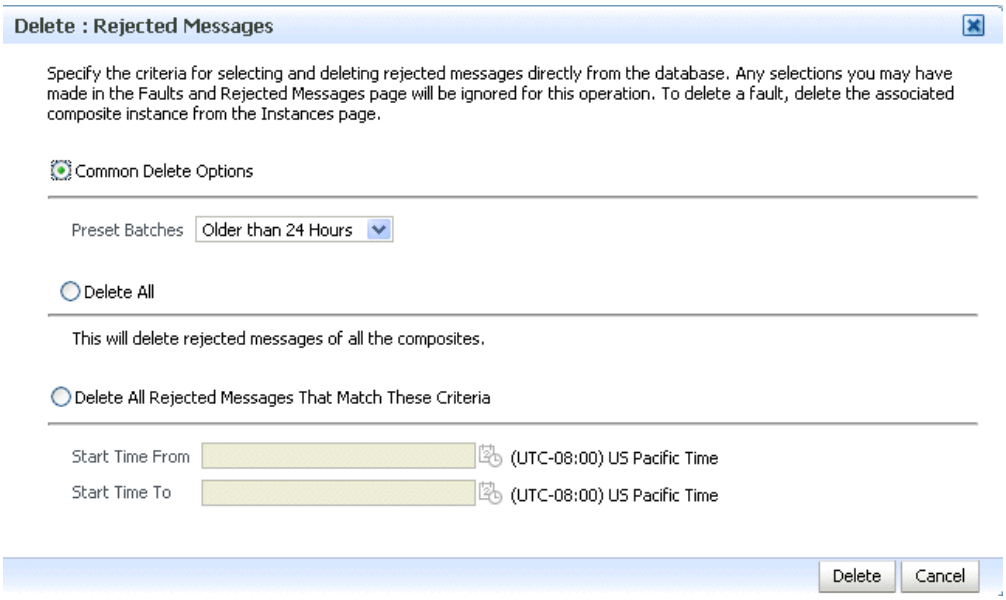
- Options for selecting instance recovery actions (for example, retry, abort, replay, and others), deleting rejected messages, and performing bulk message recovery.
- Faults and rejected messages, including the error message, whether you can recover from the fault, the time of the fault, if the fault message is classified as a rejected message (if so, a checkmark is displayed), the SOA composite application in which the fault occurred, the fault location, the instance ID, and a link to log files describing the fault.



3. Click **Delete Rejected Messages**.

Note: Oracle recommends that you run the purge scripts to delete composite instances in production environments. The purge scripts have better performance and scalability. Only use the **Delete Rejected Messages** option to manage exceptions not covered by the purge scripts. For more information about the purge scripts, see Section 9.3, "Deleting Large Numbers of Instances with the Purge Scripts."

This displays the Delete: Rejected Messages dialog for specifying criteria for deleting rejected messages of all the composites directly from the database.



4. Specify criteria and click **Delete**.

For information about recovering from faults at the SOA Infrastructure level, see Section 8.4, "Recovering from SOA Composite Application Faults at the SOA Infrastructure Level."

8.7 Deleting Rejected Messages from the Application Home Page

You can delete rejected messages for the current SOA composite application directly from the database by specifying a criteria in the Delete: Rejected Messages dialog.

To delete rejected messages for the current SOA composite application:

1. Access this page through one of the following options:

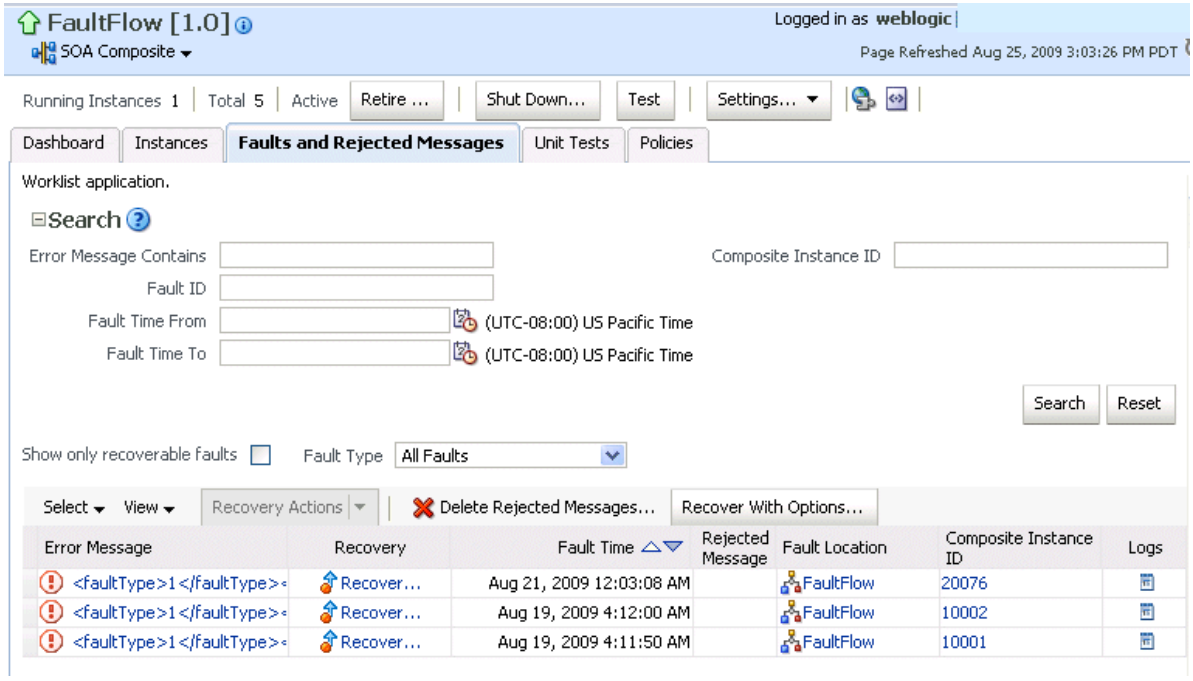
From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select **Deployed Composites**. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, expand the partition. 2. Select a specific SOA composite application.

2. Click the **Faults and Rejected Messages** tab.

   The Faults and Rejected Messages page displays the following details for all SOA composite application faults:

   - A utility for searching for a specific fault by specifying criteria and clicking **Search**. Click the **Help** icon for details.
   - Options for selecting instance recovery actions (for example, retry, abort, replay, and others), deleting rejected messages, and performing bulk message recovery.
   - Faults and rejected messages, including the error message, whether you can recover from the fault, the time of the fault, if the fault message is classified as a rejected message (if so, a checkmark is displayed), the SOA composite

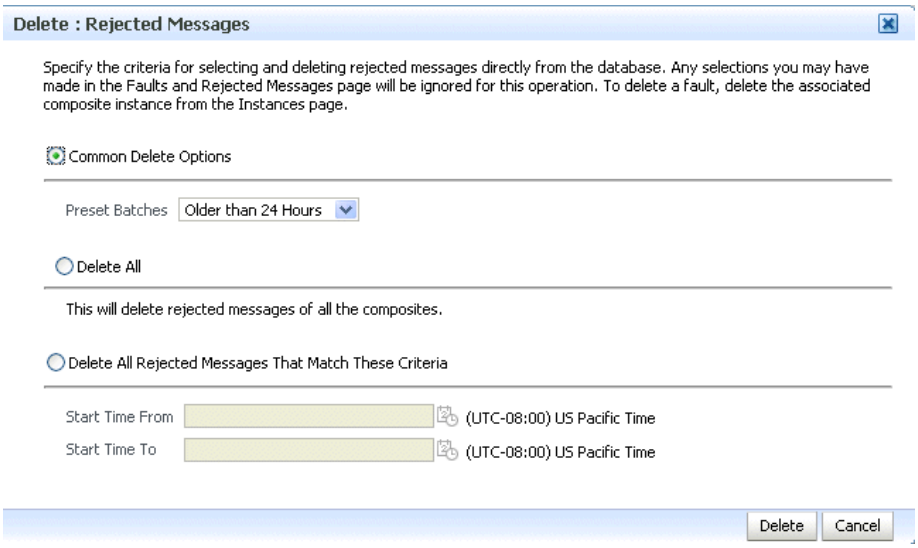
application in which the fault occurred, the fault location, the instance ID, and a link to log files describing the fault.



3. Click **Delete Rejected Messages**.

Note: Oracle recommends that you run the purge scripts to delete composite instances in production environments. The purge scripts have better performance and scalability. Only use the **Delete Rejected Messages** option to manage exceptions not covered by the purge scripts. For more information about the purge scripts, see Section 9.3, "Deleting Large Numbers of Instances with the Purge Scripts."

This displays the Delete: Rejected Messages dialog for specifying criteria for deleting rejected messages of the current composite directly from the database.



4. Specify criteria and click **Delete**.

For information about recovering from faults in the SOA composite application home page, see Section 8.5, "Recovering from SOA Composite Application Faults in the Application Home Page."

9

Managing Database Growth

This chapter describes how to manage the growth of data in your database with both the purge scripts for deleting large numbers of SOA composite application instances and database table partitioning for enabling schema tables to be range-partitioned on time intervals.

This chapter includes the following topics:

- Section 9.1, "Introduction to Managing Database Growth"
- Section 9.2, "Developing a Purging and Partitioning Methodology"
- Section 9.3, "Deleting Large Numbers of Instances with the Purge Scripts"
- Section 9.4, "Partitioning Component Tables"

> **Note:** Table partitioning is an advanced database task and must only be performed by an experienced database administrator (DBA).

For information about troubleshooting database growth issues, see Section B.6.4, "Extending Tablespaces to Avoid Problems at Runtime" and Section B.6.5, "Resolving Database Growth Issues Caused by a High Volume of Transactions."

For more information about database growth management strategies, see the white paper available at the following URL:

`http://www.oracle.com/technetwork/middleware/soasuite/overview/index.html`

9.1 Introduction to Managing Database Growth

When the amount of data in the Oracle SOA Suite database grows very large, maintaining the database can become difficult. To address this challenge, two methods for managing database growth are provided:

- Deleting large numbers of instances with the purge script
- Partitioning component database tables

9.1.1 Deleting Large Numbers of Instances with the Purge Script

You can use the purge script for instance and rejected message deletion. The purge script is located in the *MW_HOME*/*SOA_ORACLE_HOME*`/rcu/integration/soainfra/sql/soa_purge` directory.

The purge script is adequate for most Oracle SOA Suite installations. You can consider using table partitioning when the schemas grow so large that the purge script does not meet your performance needs.

For more information about the purge script, see Section 9.3, "Deleting Large Numbers of Instances with the Purge Scripts."

9.1.2 Partitioning the Component Database Tables

Oracle SOA Suite has been instrumented with partition keys that enable DBAs to take advantage of Oracle RDBMS partitioning features and capabilities. This action enables the schema tables to be range-partitioned on time intervals. This is useful when you must reduce the database maintenance window of large tables. (Though not discussed in this chapter, this also provides for the possibility of archiving partitioned data.)

The task of partitioning the Oracle SOA Suite tables must be performed by an experienced DBA. Since partitioning tables is considered a core DBA skill, this chapter does not provide detailed, step-by-step instructions on how to partition tables. Rather, it provides the DBA with the knowledge and understanding of Oracle SOA Suite schemas and their associated scripts. With this knowledge, the DBA can customize any partitioning strategy for their environment, and incorporate any tuning parameters in response to the performance of their database. Tuning is never a one-size-fits-all proposition or a one-off configuration change. Rather, it is an iterative process of monitoring and tuning.

The partitioning schemes discussed in this chapter can only be used with Oracle SOA Suite 11*g* Release 1 (11.1.1.4 or greater).

The following components are associated with their own database schemas:

- Oracle BPEL Process Manager
- Oracle Mediator
- Human workflow
- Oracle B2B
- SOA Infrastructure
- Oracle BPM Suite

For more information about table partitioning, see the Oracle database administration documentation library located at the following URL:

http://www.oracle.com/technetwork/indexes/documentation/index.html

Notes:

- A hash subpartition is an option the DBA may want to explore, especially for tables with large object (LOB) segments. This can assist with high water (HW) enqueue contention.
- A global hash index on primary keys that are monotonically increasing (like `CIKEY`) may relieve block contention.

9.1.2.1 Referential Integrity and Equipartioning

For performance reasons, the Oracle BPEL Process Manager, Oracle Mediator, human workflow, Oracle B2B, SOA Infrastructure, and Oracle BPM Suite schemas have no foreign key constraints to enforce integrity. This fact discounts the use of the 11*g* RDBMS feature known as referential partitioning. This feature provides significant

benefits because it equipartitions master and detail tables across foreign key constraints. Equipartioning means that the associated dependent table rows are in a database partition with the same partition key interval as their master table rows.

One benefit of this feature is that the state (for example, completed, faulted, and so on) of each detail row in the equipartition can be inferred from its associated master table row.

Although the 11*g* RDBMS referential partitioning feature cannot be used, similar behavior can be mimicked to achieve some of the same benefits. The Oracle BPEL Process Manager, Oracle Mediator, human workflow, Oracle B2B, SOA Infrastructure, and Oracle BPM Suite components ensure that the partition key of every detail table row is the same as the partition key of its master table row (that is, the date (timestamp) that is the partition key is pushed down). To then complete the setup, the DBA must ensure that the master and detail tables are range-partitioned on the same intervals. Some examples are provided in subsequent sections of this chapter.

Note: You may decide that referential integrity of aged partitions is not a concern for your site. For example, the site may have ample disk space, allowing data to significantly age, or there may be no apparent, adverse impact of allowing unreferenced data to be stored in the dependent tables.

9.1.2.2 Introduction to Partition Key Selection

The following factors were considered when selecting the schema partition keys:

- Convey or imply state (for example, completed) for referential integrity
- Allow range partitioning on time intervals for maintenance operations
- Be static to avoid row movement that may lead to unreferenced data
- Be static to avoid row movement when table maintenance operations are performed
- Provide performance benefits for console queries through partition pruning

9.2 Developing a Purging and Partitioning Methodology

This section summarizes the main points into an action plan that you can follow to purge and partition the dehydration store. Purging is an essential part of any plan and should be performed when data is consuming too much space or you have some other reason for removing the data.

There are three main strategies for reducing the size of the schemas:

- Purge script
- Purge script + partitioning (or, more correctly, dropping table partitions)
- Partitioning all tables

In the first two cases, the same purge script is used, although if you are partitioning, you must edit the purge script to comment out your partitioned tables.

The purge script uses standard SQL `DELETE` statements to remove rows from the BPEL tables. For most sites, this is sufficient. However, some sites accumulate so much data that the purge script takes too long to run. In this case, partitioning becomes the better solution. The trade off is that partitioning involves significantly more database maintenance. Moreover, partitioning is an advanced technique and requires a

knowledgeable and skilled DBA. By contrast, running the purge script is straightforward and does not require significant DBA knowledge.

Try to profile the input messages, database growth rate, and how much data is purged in the purge process. If the input rate and purge rate match, then regular purging is sufficient. Otherwise, consider partitioning.

If you use partitioning, Oracle recommends that you add disk space and eventually drop the partition. However, this creates additional requirements for managing disk capacity, deciding on the correct partition size, and so on. Do not use partitioning and then rely on the purge script to reclaim disk space.

9.3 Deleting Large Numbers of Instances with the Purge Scripts

Deleting thousands of instances with the **Delete With Options** button on the Instances page of a SOA composite application in Oracle Enterprise Manager Fusion Middleware Control takes time and may result in a transaction timeout. Instead, use the purge scripts for deleting instances. Note the following details about the purge scripts:

- The purge scripts delete instances that have completed or are in error (have faulted). For more information, see Section 9.3.3, "Purge States."

  Note: For Oracle Mediator resequencer tables, faulted messages are not purged. For more information, see Section 9.3.4, "Resequenced Message Purge States for Oracle Mediator."

- The purge scripts do not delete instances that are in-flight or can be recovered (are in a recovery required state).
- The purge scripts delete all Oracle SOA Suite-related tables *except* for Oracle B2B. If you have an installation in which Oracle SOA Suite and Oracle B2B are co-located, ensure that you also invoke the Oracle B2B purge scripts. If you have separate Oracle SOA Suite and Oracle B2B installations, you must run only the appropriate purge scripts on the respective product schemas. For information about purging Oracle B2B, see Chapter "Purging Data" and Chapter "B2B Command-Line Tools" of the *Oracle Fusion Middleware User's Guide for Oracle B2B*.

The following sections provide examples of how to use the script:

- Section 9.3.1, "Looped Purge Script"
- Section 9.3.2, "Looped Purge in Parallel Script with dbms_scheduler"
- Section 9.3.3, "Purge States"
- Section 9.3.4, "Resequenced Message Purge States for Oracle Mediator"
- Section 9.3.5, "Purging the Instances of a Specific SOA Composite Application"
- Section 9.3.6, "Executing the Purge Scripts"

Notes:

- If you use the `purge_soainfra_oracle.sql` PL/SQL script provided in releases *before* 11*g* Release 1 (11.1.1.4), note that this script is only supported on Oracle databases. There is no purge script support on the Microsoft SQL Server or IBM DB2 database, either with the `purge_soainfra_oracle.sql` purge script or with the newer purge script provided with release 11*g* Release 1 (11.1.1.4 or later). Only Oracle databases are supported.
- The `purge_soainfra_oracle.sql` PL/SQL purge script provided in pre-11.1.1.4 releases has been deprecated. If you are an existing user of this script, you can continue to use it against your database in 11*g* Release 1 (11.1.1.4 or later). However, starting with 11*g* Release 1 (11.1.1.4), this script is no longer shipped. Oracle recommends that you use the purge script provided with 11*g* Release 1 (11.1.1.4 or later).
- When upgrading from 11*g* Release 1 (11.1.1.3) to 11*g* Release 1 (11.1.1.4) or later, ensure that you run the purge setup scripts from the 11.1.1.4 or later location, respectively, as this contains the latest purge details. For more information about upgrade, see *Oracle Fusion Middleware Upgrade Guide for Oracle SOA Suite, WebCenter Portal, and ADF*.

9.3.1 Looped Purge Script

The master purge script includes a looping construct that allows for a batched purge. You can also provide this script with a `max_runtime` parameter that stops looping after the value for this parameter is exceeded.

The master script drives the purge of SOA database tables. You can use the `delete_instances` procedure to purge SOA database tables.

Note: Set `max_runtime` to a higher value if there are many instances to purge. In this case, you should expect to wait for a longer time before the script exits. Alternatively, use a smaller batch size if you want the purge script to exit sooner.

9.3.1.1 delete_instances Procedure

Use the `delete_instances` procedure to delete instances. Example 9–1 shows the syntax.

Example 9–1 delete_instances Procedure Syntax

```
procedure delete_instances (
                  min_creation_date in timestamp,
                  max_creation_date in timestamp,
                  batch_size in integer,
                  max_runtime in integer,
                  retention_period in timestamp,
                  purge_partitioned_component in boolean
                  composite_name in varchar2
                  composite_revision in varchar2
                  soa_partition_name in varchar2
                  );
```

Table 9–1 describes the script parameters.

Table 9–1 delete_instances Procedure Parameter Descriptions

Parameter	Description
`min_creation_date`	Beginning creation date for the composite instances.
`max_creation_date`	Ending creation date for the composite instances.
`batch_size`	Batch size used to loop the purge. The default value is `20000`.
`max_runtime`	Expiration at which the purge script exits the loop. The default value is `60`. This value is specified in minutes.

`retention_period`	Retention period is only used by the BPEL process service engine (in addition to using the creation time parameter). This functionality is not extended to other components.
	This parameter checks for and deletes records in the `cube_instance` table. The value for this parameter must be greater then or equal to `max_creation_date`. The default value is `null`.
	Specify a retention period if you want to retain the composite instances based on the `modify_date` of the BPEL instances (`cube_instance`).
	In this example, the `modify_date` of the BPEL instances table, which can be different than the composite `created_date`, is used as a second level of filtering:
	`min_creation_date = 1st June 2011`
	`max_creation_date = 30 June 2011`
	`retention_period = 1st July 2011`
	This deletes all composite instances in which the `creation_time` of the composite is between `1st June 2011` and `30 June 2011` and the `modify_date` of the `cube_instance` is less than `1st July 2011`

`purge_partitioned_component`	Users can invoke the same purge to delete partitioned data. The default value is `false`.
`composite_name`	The name of the SOA composite application. This parameter, along with the `composite_revision` and `soa_partition_name` parameters, enables you to purge the instances of a specific SOA composite application. For more information, see Section 9.3.5, "Purging the Instances of a Specific SOA Composite Application."
`composite_revision`	The revision number of the SOA composite application.
`soa_partition_name`	The partition in which the SOA composite application is included.

Note: If you do not provide a value for `retention_period`, the value for this property defaults to the value of `max_creation_date` (this is, if `retention_period` equals `null`, then `retention_period` = `max_creation_date`). This consequence also applies to the script parameters described in Section 9.3.2, "Looped Purge in Parallel Script with dbms_scheduler."

9.3.2 Looped Purge in Parallel Script with dbms_scheduler

This script is functionally the same as the looped purge script described in Section 9.3.1, "Looped Purge Script." However, this script uses the `dbms_scheduler` package to spawn multiple purge jobs, with each job working on subset data.

You can use the following procedure to purge SOA database tables.

> **Note:** If you have a multiple CPU host, use of the parallel script can be beneficial. However, Oracle recommends that you enable the parallel script only during off hours. In addition, when purging data during off hours, Oracle recommends that you drop indexes before purging large amounts of data and then add the indexes back in. This speeds up the purge process, and also keeps indexes from becoming unbalanced.

9.3.2.1 delete_instances Procedure in Parallel

Use the `delete_instances` procedure in parallel to delete instances. Example 9–2 shows the syntax.

Example 9–2 delete_instances Procedure in Parallel Syntax

```
PROCEDURE delete_instances_in_parallel (
                  min_creation_date in timestamp,
                  max_creation_date in timestamp,
                  batch_size in integer,
                  max_runtime in integer,
                  retention_period in integer,
                  DOP in integer
                  max_count integer,
                  purge_partitioned_component in boolean)
                  composite_name in varchar2
                  composite_revision in varchar2
                  soa_partition_name in varchar2
```

Table 9–2 describes the script parameters.

Table 9–2 delete_instances in Parallel Procedure Parameter Descriptions

Parameter	Description
`min_creation_date`	Beginning creation date for the composite instances.
`max_creation_date`	Ending creation date for the composite instances.
`batch_size`	Batch size used to loop the purge. The default value is `20000`.
`max_runtime`	Expiration time at which the purge script exits the loop. The default value is `60`. This value is specified in minutes.
`retention_period`	Retention period is only used by the BPEL process service engine (in addition to using the creation time parameter). The default value is `null`. For more information about this parameter, see Table 9–1.
`DOP`	Defines the number of parallel jobs to schedule. The default value is `4`.
`max_count`	Defines the number of rows processed (*not* the number of rows deleted). A big `temp` table is created and then jobs are scheduled to purge based on the data. This is the maximum purge row count to use; it defaults to one million. The default value is `1000000`.

Table 9–2 (Cont.) delete_instances in Parallel Procedure Parameter Descriptions

Parameter	Description
`purge_partitioned_component`	Users can invoke the same purge to delete partitioned data. The default value is `false`.
`composite_name`	The name of the SOA composite application. This parameter, along with the `composite_revision` and `soa_partition_name` parameters, enables you to purge the instances of a specific SOA composite application. For more information, see Section 9.3.5, "Purging the Instances of a Specific SOA Composite Application."
`composite_revision`	The revision number of the SOA composite application.
`soa_partition_name`	The partition in which the SOA composite application is included.

9.3.3 Purge States

Instances in the following states are purged with the purge scripts:

- Completed successfully
- Faulted
- Terminated by user
- Stale
- Unknown (instance tracking is disabled)

Purging of the following instance states is not supported:

- Instances pending recovery at the BPEL process service engine level or SOA composite application level
- Running instances

To purge these instances, you must first move them to one of the instance states supported by the purge scripts.

9.3.4 Resequenced Message Purge States for Oracle Mediator

The purge scripts include purge commands to purge the information persisted in the Oracle Mediator resequencer tables (`mediator_group_status` and `mediator_resequencer_message`). The following information is purged from the resequencer tables when you run the purge scripts:

- Completed and aborted messages for all resequencer types
- Timed out messages for standard resequencers
- Groups in a ready state for best effort and FIFO (first in/first out) resequencers (these are the only groups that can be purged)

To allow fault recovery and message processing to be completed, the purge scripts do not purge all resequenced message information. In addition, standard resequencer groups store information that should not be purged. The following are not purged when you run the purge scripts:

- Faulted messages for all resequencer types
- Running messages for all resequencer types
- Group information for standard resequencers

- Groups in a state other than ready for best effort and FIFO resequencers

> **Note:** The purge scripts for the Oracle Mediator resequencer purge messages first and then move on to groups. If there are messages for a group in the `mediator_resequencer_message` table, the group cannot be deleted.

The above describes processing for both looped and parallel processing of the purge scripts, and regardless of whether instance tracking is enabled or disabled. Before any sequence groups are purged, a check is performed to verify that all messages associated with the group are processed.

Below is a list of group state codes used in the resequencer tables:

- **0**: Ready
- **1**: Locked
- **2**: Error
- **4**: Timed out
- **6**: Group error

Below is a list of message state codes used in the resequencer tables:

- **0**: Ready
- **1**: Locked
- **2**: Completed
- **3**: Error
- **4**: Timed out (this is ignored)
- **5**: Aborted

9.3.5 Purging the Instances of a Specific SOA Composite Application

You can purge the instances of a specific SOA composite application and leave the instances of other composites unpurged. This action enables you to purge certain flows more frequently than others due to high volume or retention period characteristics.

The purge scripts include an option for purging based on `COMPOSITE_DN`. Purging based on `COMPOSITE_DN` is supported with the parameters `COMPOSITE_NAME` and `COMPOSITE_REVISION`.

The purge logic is based on flows (ECIDs) and not `COMPOSITE_IDs`. Therefore, apart from the intended `COMPOSITE_DNs`, other composites sharing the same ECID may get deleted. The following scenarios may occur:

- A composite instance is closed, but the flow is still open:

  In a scenario in which composite A calls composite B, the purge intends to delete instances of composite A. However, there may be a case in which an instance of composite A is closed, but the corresponding composite B instance is still open. Therefore, because the overall flow is still in an open state, the composite A instance (even though closed) is not purged.

- The composite instance is closed and the flow is also closed:

Composite A again calls composite B. The purge intends to delete instances of composite A. Therefore, in a case in which composite A is closed and composite B is also closed, because the overall flow is closed, both composite instances A and B are purged.

These scenarios maintain the consistency of the flow.

For information about the `COMPOSITE_NAME` and `COMPOSITE_REVISION` parameters, see Section 9.3.1, "Looped Purge Script" and Section 9.3.2, "Looped Purge in Parallel Script with dbms_scheduler."

9.3.6 Executing the Purge Scripts

This section describes how to execute the purge scripts.

To execute the purge scripts:

1. In SQL*Plus, connect to the database `AS SYSDBA`:

   ```
   CONNECT SYS AS SYSDBA
   ```

2. Execute the following SQL commands:

   ```
   GRANT EXECUTE ON DBMS_LOCK to USER;
   GRANT CREATE ANY JOB TO USER;
   ```

   where *`USER`* is the `soainfra` account to execute the scripts. These privileges are required to run the scripts.

3. Load the purge scripts by executing the main purge script in the *`MW_HOME`*`/`*`SOA_ORACLE_HOME`*`/rcu/integration/soainfra/sql/soa_purge` directory.

   For a parallel purge, the debug logs from the jobs spawned by a parallel purge are logged into files created in the directory named `SOA_PURGE_DIR`. This directory must be accessible to the Oracle database.

4. Create `SOA_PURGE_DIR` and grant write permissions to the `soainfra` user.

   ```
   mkdir -p /tmp/purgelog
   CREATE OR REPLACE DIRECTORY SOA_PURGE_DIR AS 'SERVER_DIRECTORY'
   ```

   where *`SERVER_DIRECTORY`* is the name of the directory to create (for example, `'/tmp/purgelog/'`). Note the required single quotes around the directory path.

5. If you want to execute the scripts in debug mode, run `common/debug_on.sql` and set `serverout` to `on` in SQL*Plus. This step is optional.

   ```
   SET SERVEROUTPUT ON
   ```

   The logs from the spawned jobs are logged into the directory created in Step 4 (separate files per job). The rest of the logs are displayed on `stdout` (or the spool file, if configured).

   There are two options for purging:

   - Looped purge
   - Parallel purge

6. Execute the purge scripts as shown below. Examples are provided for both options.

   a. For looped purge:

      ```
      DECLARE
      ```

```
 MAX_CREATION_DATE timestamp;
 MIN_CREATION_DATE timestamp;
 batch_size integer;
 max_runtime integer;
 retention_period timestamp;

BEGIN

 MIN_CREATION_DATE := to_timestamp('2010-01-01','YYYY-MM-DD');
 MAX_CREATION_DATE := to_timestamp('2010-01-31','YYYY-MM-DD');
  max_runtime := 60;
  retention_period := to_timestamp('2010-01-31','YYYY-MM-DD');
 batch_size := 10000;
   soa.delete_instances(
   min_creation_date => MIN_CREATION_DATE,
   max_creation_date => MAX_CREATION_DATE,
   batch_size => batch_size,
   max_runtime => max_runtime,
   retention_period => retention_period,
   purge_partitioned_component => false);
END;
/
```

b. For parallel purge:

```
DECLARE

 max_creation_date timestamp;
 min_creation_date timestamp;
 retention_period timestamp;
BEGIN

 min_creation_date := to_timestamp('2010-01-01','YYYY-MM-DD');
 max_creation_date := to_timestamp('2010-01-31','YYYY-MM-DD');
 retention_period := to_timestamp('2010-01-31','YYYY-MM-DD');

  soa.delete_instances_in_parallel(
   min_creation_date => min_creation_date,
   max_creation_date => max_creation_date,
   batch_size => 10000,
   max_runtime => 60,
   retention_period => retention_period,
   DOP => 3,
   max_count => 1000000,
   purge_partitioned_component => false);

END;
```

9.4 Partitioning Component Tables

The runtime and schema code for the following components has been modified to store the flow creation date column with their transactional tables.

- Oracle BPEL Process Manager
- Oracle Mediator
- Human workflow
- Oracle B2B

- SOA Infrastructure (includes component instances)
- Oracle BPM Suite

The `CPST_CREATED_DATE` column contains the flow creation date time populated by the instance tracking code. This is available as the normalized message property `oracle.integration.platform.instance.CommonConstants.COMPOSITE_INSTANCE_CREATED_TIME`.

All SOA components are partitioned on the same partition key. These partitioned components use the same time range and partition ID.

> **Note:** Before performing complete or partial partitioning, run the purge scripts.

9.4.1 Configuring Partitions

Partitioning is *not* configured by default; it is a postinstallation step that must be performed manually. Once you decide to implement partitioning of the database, you must perform some initial configuration tasks only once:

- Using the information in this chapter, decide which groups you want to partition.
- For each of those groups, decide which tables you want to partition, remembering that there are some mandatory tables in each group that must be partitioned.
- For each group, decide on the partition interval.
- Create the partition scripts to partition the Oracle SOA Suite schemas. No scripts are supplied; each DBA is responsible for creating the partition scripts appropriate for their environment.
- Edit the purge script and remove references to any tables that you partitioned.

9.4.2 Introduction to the Verification Script

A verification script is provided for a DBA to identify when to drop a partition and its equipartitioned dependent table. The verification script also identifies if there are active, long running instances. You can then move these instances to a different partition, and then drop the original partition.

> **Note:** The verification script does not drop any partitions; it just ensures that partitions are eligible to be dropped.

9.4.3 Component Tables

This section describes partitioning constraints and lists the component tables, the groups to which they belong, and their partition key.

9.4.3.1 Partitioning Constraints

Note the following table partitioning constraints:

- You have the choice of the following approach to partitioning:
  - Complete partitioning: All tables of a service component/service engine are partitioned.
  - No partitioning: No tables of a service component/service engine are partitioned.

– Partial partitioning: Restrict partitioning to specific tables with a high growth rate. Table 9–3 lists the master and dependent tables that can be partitioned.

Table 9–3 Partial Partitioning

Master Table	Dependent Table of Master Table
COMPOSITE_INSTANCE	REFERENCE_INSTANCE
CUBE_INSTANCE	CUBE_SCOPE
XML_DOCUMENT	None
MEDIATOR_INSTANCE	MEDIATOR_CASE_INSTANCE
MEDIATOR_PAYLOAD	None

You can partition any of the tables in Table 9–3 by following these constraints:

* If you want to partition a dependent table, you must also partition its master table.
* All tables should be equipartitioned along the same date ranges and the same name.
* Always partition the COMPOSITE_INSTANCE table. This constraint is essential when the **Audit Level** property is set to **Development** or **Production** for any of the composites. The verification script checks for active flows based on the active composite instances within that partition. Therefore, if the COMPOSITE_INSTANCE table is not partitioned, the entire verification script logic based on the equipartitioning of all the tables fails.

- Regardless of the group and component, all tables that are partitioned use the same time range and the partition ID.

9.4.3.2 Component Tables, Range Partition Keys, and Groups

Table 9–4 through Table 9–9 are divided into three groups.

- Group 1: This includes tables that are directly related to the end-to-end flow trace of a composite. A majority of the tables fall into this group.
- Group 1A: This includes a small set of tables that are not directly related to the flow trace (for example, REJECTED_MESSAGE).
- Group 2: This includes a small set of tables that have a dependency on multiple tables from Group 1 and 1A tables. You must first execute the group 1 verification script and drop the group 1 partitions before executing the group 2 verification script.

Note: Groups 1 and 1A are combined in the verification script. Executing the verification script does not require you to have knowledge of this classification.

Table 9–4 Component: SOA Infrastructure

Table	Range Partition Key	Group
COMPOSITE_INSTANCE	PARTITION_DATE	1
REFERENCE_INSTANCE	CPST_PARTITION_DATE	1

Table 9–4 (Cont.) Component: SOA Infrastructure

Table	Range Partition Key	Group
COMPOSITE_ INSTANCE_FAULT	CPST_PARTITION_DATE	1
COMPOSITE_ SENSOR_VALUE	CPST_PARTITION_DATE	1
COMPONENT_ INSTANCE	CPST_PARTITION_DATE	1
REJECTED_MESSAGE	CREATED_TIME	1A
REJECTED_MSG_ NATIVE_PAYLOAD	RM_PARTITION_DATE	1A
INSTANCE_PAYLOAD	CREATED_TIME	2
COMPOSITE_ INSTANCE_ASSOC	CREATED_TIME	2

Table 9–5 Component: Oracle BPEL Process Manager

Table	Range Partition Key	Group
CUBE_INSTANCE	CPST_INST_CREATED_TIME	1
CI_INDEXES	CI_PARTITION_DATE	1
CUBE_SCOPE	CI_PARTITION_DATE	1
DOCUMENT_CI_REF	CI_PARTITION_DATE	1
AUDIT_TRAIL	CI_PARTITION_DATE	1
AUDIT_DETAILS	CI_PARTITION_DATE	1
DLV_SUBSCRIPTION	CI_PARTITION_DATE	1
WORK_ITEM	CI_PARTITION_DATE	1
AUDIT_COUNTER	CI_PARTITION_DATE	1
WI_FAULT	CI_PARTITION_DATE	1
DLV_MESSAGE	RECEIVE_DATE	1A
HEADERS_ PROPERTIES	DLV_PARTITION_DATE	1A
DOCUMENT_DLV_ MSG_REF	DLV_PARTITION_DATE	1A
XML_DOCUMENT	DOC_PARTITION_DATE	2

Table 9–6 Component: Oracle Mediator

Table Name	Range Partition Key	Group
MEDIATOR_ INSTANCE	COMPOSITE_CREATION_DATE	1
MEDIATOR_CASE_ INSTANCE	MI_PARTITION_DATE	1
MEDIATOR_CASE_ DETAIL	MI_PARTITION_DATE	1
MEDIATOR_AUDIT_ DOCUMENT	MI_PARTITION_DATE	1

Table 9–6 (Cont.) Component: Oracle Mediator

Table Name	Range Partition Key	Group
MEDIATOR_ DEFERRED_MESSAGE	CREATION_DATE	1A
MEDIATOR_PAYLOAD	CREATION_TIME	2

Table 9–7 Component: Human Workflow

Table	Range Partition Key	Group
WFTASK	COMPOSITECREATEDTIME	1
WFTask_TL	COMPOSITECREATEDTIME	1
WFTaskHistory	COMPOSITECREATEDTIME	1
WFTaskHistory_TL	COMPOSITECREATEDTIME	1
WFComments	COMPOSITECREATEDTIME	1
WFMessageAttribute	COMPOSITECREATEDTIME	1
WFAttachment	COMPOSITECREATEDTIME	1
WFAssignee	COMPOSITECREATEDTIME	1
WFReviewer	COMPOSITECREATEDTIME	1
WFCollectionTarget	COMPOSITECREATEDTIME	1
WFRoutingSlip	COMPOSITECREATEDTIME	1
WFNotification	COMPOSITECREATEDTIME	1
WFTaskTimer	COMPOSITECREATEDTIME	1
WFTaskError	COMPOSITECREATEDTIME	1
WFHeaderProps	COMPOSITECREATEDTIME	1
WFEvidence	COMPOSITECREATEDTIME	1
WFTaskAssignmentStatistic	COMPOSITECREATEDTIME	1
WFTaskAggregation	COMPOSITECREATEDTIME	1

Table 9–8 Component: Oracle B2B

Table	Range Partition Key	Group
B2B_BUSINESS_ MESSAGE	CPST_INST_CREATED_TIME	1
B2B_APP_MESSAGE	CPST_INST_CREATED_TIME	1
B2B_WIRE_MESSAGE	CPST_INST_CREATED_TIME	1
B2B_DATA_STORAGE	CPST_INST_CREATED_TIME	1
B2B_EXT_BUSINESS_ MESSAGE	CPST_INST_CREATED_TIME	1

Table 9–9 Component: Oracle BPM Suite

Table	Range Partition Key	Group
BPM_AUDIT_QUERY	CI_PARTITION_DATE	1
BPM_MEASUREMENT_ ACTIONS	CI_PARTITION_DATE	1
BPM_MEASUREMENT_ ACTION_EXCEPS	CI_PARTITION_DATE	1
BPM_CUBE_ AUDITINSTANCE	CI_PARTITION_DATE	1
BPM_CUBE_ TASKPERFORMANCE	CI_PARTITION_DATE	1
BPM_CUBE_ PROCESSPERFORMANCE	CI_PARTITION_DATE	1

9.4.4 Executing the Verification Script

A verification script is provided for the DBA to identify when to drop a partition and its equipartitioned dependent table. The verification script is located in *MW_HOME*/*SOA_ORACLE_HOME*/rcu/integration/soainfra/sql/verify.

To execute the verification script:

1. Create a directory with the SQL command PART_DIR. For example:

```
CREATE DIRECTORY PART_DIR AS '/tmp/verify'
```

2. Provide the soainfra user with write privileges on this directory. The log and SQL files are generated in this directory.
3. For executing the stored procedure, the client script soa_exec_verify.sql can be used. Edit soa_exec_verify.sql and enter the partition names that require verification in the array mySoa_drv_list.
   a. To execute function verify_soa.verify_1, pass 1 as the parameter.
   b. To execute function verify_soa_verify_2, pass 2 as the parameter.
4. Review the logs and SQL files generated in the PART_DIR directory.

Note: A verification script is not provided for business rules.

9.4.5 Verifying and Dropping Partitions

To verify and drop partitions:

1. Execute function verify_soa.verify_1.
2. Check the log file in the PART_DIR folder with the name SOA_*PARTITION_NAME*_LOG_1 for any failures. If you have active, long running instances, see Section 9.4.6, "Moving Active, Long Running Instances to a Different Partition."
3. Drop the partitions that can be dropped by using the script generated in the PART_DIR folder with the name SOA_*PARTITION_NAME*_RESULT_1.sql.
4. Execute verify_soa.verify_2.

5. Check the log file in the `PART_DIR` folder with the name `SOA_`*`PARTITION_NAME`*`_LOG_2` for any failures.
6. Drop the droppable partitions using the script generated in the `PART_DIR` folder with the name `SOA_`*`PARTITION_NAME`*`_RESULT_2.sql`.

> **Note:** There is an issue caused by the existence of foreign key constraints in the Oracle B2B table. When dropping the partition, the B2B partition purge is invoked, the foreign key constraints must be disabled before dropping the partition and enabled afterward. To perform this action, execute the PL/SQL procedures `b2b_disable_constraints` and `b2b_enable_constraints` at the appropriate steps in the above procedure. Because foreign keys are enabled and disabled in these procedures, it is not recommended to run them on a live system.

9.4.6 Moving Active, Long Running Instances to a Different Partition

The verification script checks if there are active instances in the partition. When no instances are active, the partition can be dropped. However, you may have active, long running instances present in the partition. These active instances prevent a partition from being dropped. To avoid this problem, you can move long running instances to a different partition.

The Oracle database provides an option for enabling and disabling row movements across partitions. When you create or alter a partitioned table, a row movement clause, either `ENABLE ROW MOVEMENT` or `DISABLE ROW MOVEMENT`, can be specified. This clause either enables or disables the movement of a row to a new partition if its key is updated. This option can handle long-running processes that prevent partition drops.

The verify scripts provide a count of total instances, open instances, and the percentage of open instances in a partition. Based on this information, you now have a choice of executing the row movement procedure. This updates the partition keys of the targeted tables, which in turn initiates a row movement of these instances to a different partition. Once all active instances from all partitioned tables have been moved to a different partition, the targeted partition can be dropped.

Notes:

- Create a separate partition for active, long running transactions. You can then move these instances from the current partition to the new partition by providing the *new_partition_date* that falls into that partition range. It is recommended that you run a periodical purge on these long-running instances.
- Row movement involves expensive updates to multiple rows. Only exercise row movement when the number of active processes is small. While the verification script provides a count of active instances, many of the dependent tables have a many-to-one relationship with the master tables. This can become a larger set of rows to move between partitions (if the dependent tables are also partitioned). Use row movement discretely based on the tables partitioned, the active instances in the partition, the data shape, and the available infrastructure setup.
- Enabling row movement between partitions can significantly degrade runtime performance if the partition key is updated frequently. However, a partition key column is never changed for the Oracle SOA Suite database tables once they are created. Therefore, no row movement at runtime occurs.

To move long running instances to a different partition:

1. Run the verification script for group 1. This script is described in Section 9.4.4, "Executing the Verification Script."
2. Check the log scripts to see if there are any active instances preventing the partition drop.
3. Run the row movement procedure. Based on the logs from step 2, check the count of open instances. Based on this count, decide if the row movement script should be executed or if dropping of partitioning should be postponed.
   a. Log in to SQL*Plus as the `SOAINFRA` user:

      ```
      CONNECT SOAINFRA/password
      ```

   b. Execute the following PL/SQL procedure to move rows for group 1 tables:

      ```
      SQL> PROCEDURE exec_row_movement_1( partition_name in varchar2,
      new_partition_date in timestamp );
      ```

      where:

      partition_name is the name of the partition on which to execute row movement.

      new_partition_date is the new date with which to update the partition key column.
4. Execute the purge script to delete nonpartitioned tables as described in Section 9.3.6, "Executing the Purge Scripts." The `purge_partitioned_component` parameter of the purge procedures must be set to `false`.
5. Drop the group 1 partitions (based on step 3).
6. Run the verification script for group 2.
7. Check if there are any active instances holding back the partition drop.

8. Run the row movement procedure. Based on the logs from step 7, check the count of open instances. Based on this count, decide if the row movement script should be executed or if dropping of partitioning should be postponed.

   a. Return to SQL*Plus as the `SOAINFRA` user.

   ```
   CONNECT SOAINFRA/password
   ```

   b. Execute the following PL/SQL procedure to move rows for group 2 tables:

   ```
   SQL> PROCEDURE exec_row_movement_2( partition_name in varchar2,
   new_partition_date in timestamp );
   ```

   where:

   `partition_name` is the name of the partition on which to execute row movement.

   `new_partition_date` is the new date with which to update the partition key column.

9. Drop the group 2 partitions (based on step 8).

9.4.7 Partial Partitioning of Components

If you have an environment in which some components are partitioned, while other components are not partitioned, the nonpartitioned data set must be purged using the purge scripts described in Section 9.3, "Deleting Large Numbers of Instances with the Purge Scripts."

For example, assume human workflow is not partitioned, while other components are partitioned. The verification script reports that all SOA partitions can be dropped using the command for dropping partitions. However, the human workflow tables continue to hold workflow data until the data is purged using the loop/parallel purge scripts.

10

Programmatically Managing SOA Composite Applications

This chapter describes the Facade API, which exposes operations and attributes such as composites, components, services, and references for programmatically managing SOA composite applications during runtime. The Facade API is part of Oracle SOA Suite's Infrastructure Management Java API.

This chapter includes the following topics:

- Section 10.1, "Introduction to Programmatically Managing SOA Composite Applications"
- Section 10.2, "Facade API Interfaces"
- Section 10.3, "Facade API Examples"

10.1 Introduction to Programmatically Managing SOA Composite Applications

You can programmatically manage your SOA composite applications during runtime with the Facade API. The Facade API is part of Oracle SOA Suite's Infrastructure Management Java API. The Facade API exposes operations and attributes of composites, components, services, references and so on. The Facade API provides an alternative to managing composites with Oracle Enterprise Manager Fusion Middleware Control.

The `oracle.soa.management.facade.Locator` interface exposes a method interface as the top level entry point for most Facade APIs. The `oracle.soa.management.facade.Locator` interface exposes a method, `createConnection`, which returns a `DirectConnection` to a composite. You can use the `LocatorFactory` implementation to obtain the `DirectConnection`. For more information about the `LocatorFactory` and `DirectConnection`, see section "Introduction to the Direct Binding Invocation API" of *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*. The `DirectConnection` and `DirectConnectionFactory` (for creating direct binding connections) classes are in the `oracle.soa.api.invocation` package.

Partition management APIs are not exposed by the `Locator`. Instead, the top level entry point for partition management APIs is the `ServerManager`. You use the `ServerManagerFactory` implementation shown in Example 10–1 to create `ServerManager` objects for managing the SOA server.

Example 10–1 ServerManagerFactory

```
ServerManagerFactory smf = ServerManagerFactory.getInstance();
```

```
 Hashtable jndiProps = new Hashtable();

 jndiProps.put(Context.PROVIDER_URL, "server_JNDI_provider_url");
 jndiProps.put(Context.INITIAL_CONTEXT_FACTORY, "server_initial_context_factory_
classname");
 jndiProps.put(Context.SECURITY_PRINCIPAL, "jndi_user");
 jndiProps.put(Context.SECURITY_CREDENTIALS, "jndi_password");
 ServerManager sm = smf.createInstance(jndiProps);
```

For more information about the Facade API, see *Oracle Fusion Middleware Infrastructure Management Java API Reference for Oracle SOA Suite*.

> **Note:** The Infrastructure Management Java API also includes the Direct Binding Invocation API for inbound invocation of SOA composite applications and outbound invocations of Oracle Service Bus (OSB) or another SOA composite application. For information on using the direct binding invocation API, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

10.2 Facade API Interfaces

Table 10–1 provides an overview of the Facade API interfaces.

Table 10–1 Facade API Interfaces

Interface	Description
`ActivityInstance`	Retrieves the ID (key) of the instance to which this activity belongs, the component name, the creation date of the activity, the expiration date of the current activity, and so on.
`AsynchronousJob`	Retrieves details about asynchronous job types. **Note**: This interface has been deprecated.
`AsynchronousResult`	Retrieves the results of asynchronous processes.
`Binding`	Retrieves binding details such as URI, type and subtype.
`BindingComponent`	Retrieves the binding type and the list of faults specified by the given fault filter.
`BusinessEventInfo`	Retrieves business event details such as the consistency level, filter type, local name, namespace URI, and prefix.
`Component`	Retrieves component details such as distinguished name (DN); active, faulted, and total instances; deployment time; and so on.
`ComponentInstance`	Retrieves component instance details such as the audit trail, component instance state (for example, successfully completed, faulted, or running), conversation ID, creation date, and so on).
`Composite`	Retrieves composite details such as the distinguished name (DN); active, faulted, and total instances; deployment time; and so on.
`CompositeInstance`	Retrieves the composite instance state (for example, successfully completed, faulted, running, and so on).
`EventBridge`	Retrieves the event bridge type and the list of faults specified by the given fault filter.

Table 10–1 (Cont.) Facade API Interfaces

Interface	Description
`Fault`	Retrieves fault details such as the binding type if the fault occurred in a binding component, the component instance ID, the name of the component in which the fault occurred, the time at which the fault was generated, the service engine type if the fault occurred in a service engine, and so on.
`FaultRecoveryActionTypeConstants`	Provides action types such as terminating instances, marking a faulted activity as complete, storing the rejected message in a file, marking a faulted activity to be recovered, retrying an activity, and so on.
`FaultRecoveryResult`	Retrieves faults, errors, recoverable faults, nonrecoverable faults, and so on.
`FaultRecoveryServiceEngine`	Provides the fault recovery service engine.
`ImportInfo`	Retrieves the location of the imported resource and the import type.
`InterfaceType`	Retrieves the callback interface and interface properties.
`Locator`	Exposes a method interface as an entry point for Facade API clients.
`Message`	Retrieves message details such as the component name, composite DN, content of the message, conversation ID, message storage location, message recoverable state, and so on.
`Partition`	Performs partitioning lifecycle management tasks such as starting, stopping, activating, and retiring all the composites in a partition.
`Property`	Retrieves property details such as the default value, name, override attribute value of the property, property value source, and so on.
`Reference`	Retrieves reference details such as the WSDL URL for the target service, properties for the reference, reference name, reference bindings, and so on.
`ReferenceInstance`	Retrieves the type of the binding. DN of the composite, composite instance ID, time of instance creation, and so on.
`ServerManager`	Gets and creates partitions.
`Service`	Retrieves service details such as bindings, composite DN, interface type, name, multiplicity, and so on.
`ServiceEngine`	Retrieves the list of deployed components in the service engine, engine type, and list of faults specified by the given fault filter.
`ServiceInstance`	Retrieves service instance details such as the type of the binding. DN of the composite, composite instance ID, time of creation, and so on.
`WebServiceBinding`	Retrieves the default address URI, endpoint address URI, port, transport type, and so on.
`WebServiceBindingPort`	Retrieves the port name, port URI, service name, and so on.
`WireInfo`	Retrieves the wire reference, wire service, source URI, and target URI.
`WSDLInterfaceBinding`	Retrieves the WSDL URL.

10.3 Facade API Examples

This section provides several examples of using the Facade API to perform composite management.

10.3.1 Retrieving the State of a Composite

You can retrieve the state of a composite with the Facade API.

- Mode: active | retired

  This setting decides whether new instances can be created (active) or old ones are allowed to finish without new ones being allowed to be created (retired).

- State: on | off

  This setting is the composite state and overrides the active or retired composite modes in either, allowing call access (invoke/callback) to the composite revision (on) or not allowing call access (off).

Use `oracle.soa.management.facade.Locator#getComposite(compositeDN)` to get a reference to a composite of interest.

From the composite reference, you can query the mode and state using the following methods:

- `String getMode()`
- `String getState()`

10.3.2 Finding Composite and Component Instances

You can find composite and component instances with the Facade API.

Use the `Locator` interface to find a `Composite` (`locator.lookupComposite(compositeDN)`). If the composite has not yet been invoked, there are no instances.

You can then find its instances and get related information (for example, the instance ID, which components were executed, and so on).

Example 10–2 provides details.

Example 10–2 Finding Composite and Component Instances

```
Composite composite = locator.lookupComposite("default/OrderBookingComposite!1.0");

// The context is already the composite, so there is no need to set the DN as a filter criterion
CompositeInstanceFilter filter = new CompositeInstanceFilter();
filter.setMinCreationDate(new java.util.Date((System.currentTimeMillis() - 20000)));

// Get composite instances by filter
List compositeInstances = composite.getInstances(filter);

// for each of the returned composite instances..
for (CompositeInstance instance : compositeInstances) {
    long   instanceId = instance.getId();
    Date   created    = instance.getCreationDate();
    String state      = instance.getState();

    // Configure a component instance filter
    ComponentInstanceFilter cInstanceFilter =
```

```
                        new ComponentInstanceFilter ();

    // Get the child component instances
    List componentInstances =
        instance.getChildComponentInstances(cInstanceFilter);

    for (ComponentInstance compInstance : componentInstances) {
        String compName       = compInstance.getComponentName();
        long   compInstanceId = compInstance.getId();
        String type           = compInstance.getServiceEngine().getEngineType();

        // State values correspond to constants defined by the ComponentInstance interface
        int state = compInstance.getState();
    }

    // Retrieve composite sensors
    List sensorData = instance.getSensorData();
}
```

11

Diagnosing Problems with SOA Composite Applications

This chapter describes how to identify Oracle SOA Suite problems early and take the proper corrective actions with the assistance of the WebLogic Diagnostic Framework (WLDF) and the Oracle Fusion Middleware Diagnostic Framework.

This chapter includes the following topics:

- Section 11.1, "Introduction to the Diagnostic Frameworks"
- Section 11.2, "Creating Watches and Notifications"
- Section 11.3, "Manually Triggering and Executing Dumps"
- Section 11.4, "Viewing Incident Packages with ADR Tools"
- Section 11.5, "Querying Problems and Incidents"

For information about troubleshooting, see Appendix B, "Troubleshooting Oracle SOA Suite and Oracle BPM Suite."

11.1 Introduction to the Diagnostic Frameworks

When you monitor and diagnose problems in Oracle SOA Suite, you face the following challenges:

- Capturing diagnostic data at the moment of occurrence (also known as just-in-time diagnostics), especially for intermittent issues such as obtaining multiple thread dumps when the system hangs or obtaining heap dumps before a system runs out of memory.
- Obtaining advanced information such as data shape (counts by state and growth patterns for the SOA schema and MDS schema).
- Communicating back and forth with Oracle Support Services to provide basic information such as versions, logs, configuration files, patches applied, and so on.
- Detecting problems and taking corrective actions early before they escalate.

To address these challenges, Oracle SOA Suite is integrated with the following diagnostic frameworks that assist you in identifying problems early and taking the proper corrective actions:

- WLDF: For monitoring diagnostic scenarios using watches and notifications.
- Diagnostic Framework: For gathering SOA-specific diagnostic scenarios into data dumps that are formatted for viewing and analyzing. These data dumps can be

made accessible through the Support Work Bench and uploadable as part of a service request (SR).

11.1.1 Introduction to WLDF

WLDF is a monitoring and diagnostics framework included with Oracle WebLogic Server that defines and implements a set of services that run within server processes and participate in the standard server life cycle.

Using WLDF, you can capture diagnostic data from Oracle SOA Suite. You configure WLDF watches and notifications from Oracle WebLogic Server Administration Console to collect diagnostic data to identify problems. This data enables you to isolate and diagnose faults when they occur.

For more information about WLDF, see *Oracle Fusion Middleware Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*.

11.1.1.1 Introduction to Watches and Notifications

Watches monitor server and application states and send notifications based on criteria that you set. Watches and notifications are configured as part of a diagnostic module targeted to one or more server instances in a domain. When you create a watch, you build rule expressions for monitoring using the attributes of Oracle SOA Suite and Oracle WebLogic Server MBeans in Oracle WebLogic Server Administration Console.

For example, assume you want to be notified when the percentage of free heap memory falls below 100%. You create a watch that uses the Oracle WebLogic Server MBean `weblogic.management.runtime.JRockitRuntimeMBean` and its attribute `HeapFreePercent`. You then define logic indicating that when `HeapFreePercent` is less than 100%, you want to receive a notification. You can also use the MBean `JVMRuntimeMBean` when running with a non-JRockit virtual machine (VM).

For information about creating watches and notifications in Oracle WebLogic Server Administration Console, see Section 11.2, "Creating Watches and Notifications" and Section "Configuring the Diagnostic Framework" of *Oracle Fusion Middleware Administrator's Guide*.

11.1.1.2 Introduction to Diagnostic Scenarios and MBeans

The watch rule expressions that you create use the attributes of Oracle SOA Suite and Oracle WebLogic Server MBeans to collect data and perform monitoring. You diagnose scenarios with available MBeans to provide statistics about that scenario or to log messages. Managed beans (MBeans) are Java objects that represent JMX manageable resources in a distributed environment. The attributes of the following MBeans are available for defining in watches to monitor scenarios:

- Oracle WebLogic Server MBeans
- Diagnostic Oracle SOA Suite MBeans
- DMS metrics exposed as MBeans

Oracle SOA Suite provides several diagnostic scenarios that you can monitor with watches and notifications. Table 11–1 provides details about the supported diagnostic scenarios and the MBeans to use for monitoring.

Table 11–1 Supported Diagnostic Scenarios and MBeans

Scenario	Description	Diagnostic Data Source
Memory issues (startup, deployment, and runtime)	Monitor the free heap available. If the free heap percentage is below a threshold, a notification is triggered to generate a thread stack dump and heap dump.	Oracle WebLogic Server MBean
Deployment hanging	Monitor the elapsed time of a deployment. If it exceeds a threshold, a notification is triggered.	Oracle SOA Suite deployment MBean
Data source issues	Monitor the suspension and connection pool/transaction time outs	JDBC MBeans
Server overload	Monitor the server's self-health.	Oracle WebLogic Server MBean
Stuck threads	Monitor stuck threads. If any are found, a notification is triggered.	A stuck thread watch/Diagnostic Framework notification is automatically included. Diagnostic Framework incident packages can be created with a tool such as the ADR Command Interpreter (ADRCI). Incidents are created automatically.

Table 11–2 lists some of the available MBeans and DMS Metrics to select when creating watches for monitoring and collecting diagnostic data.

Table 11–2 MBeans and DMS Metrics

Diagnostic Data Source (MBean) and Usage	Description	Oracle WebLogic Server MBean	SOA MBean or DMS Metric
`weblogic.management.runtime.JVMRuntimeMBean/ weblogic.management.runtime.JRockitRuntimeMBean`	For memory statistics	Yes	- -
`oracle.fabric.management.wldf.mbean.DeploymentWatchMXBeanImpl`	For deployment elapsed time	- -	SOA MBean
`weblogic.management.runtime.ServerRuntimeMBean`	For health state information	Yes	- -
`weblogic.management.runtime.JDBCDataSourceRuntimeMBean` `weblogic.management.runtime.JTARuntimeMBean`	For JDBC data sources and for accessing transaction runtime characteristics	Yes	- -

For more information about Oracle WebLogic Server MBeans, see *Oracle Fusion Middleware Oracle WebLogic Server MBean Reference*.

11.1.2 Introduction to the Diagnostic Framework

The Diagnostic Framework is an Oracle Fusion Middleware feature that aids in detecting, diagnosing, and resolving problems. The problems that are targeted are critical errors such as those caused by code bugs, metadata corruption, customer data corruption, deadlocked threads, and inconsistent state. The Diagnostic Framework detects critical failures and captures dumps of relevant diagnostics information (logs, metrics, server images, and so on). WLDF watches and notifications trigger events for which the Diagnostic Framework listens and generates appropriate data dumps. The dumps are formatted into incident packages for viewing and analysis.

The problems captured as incidents include critical errors such as those described in Table 11–1. Each incident package is identified by a unique ID. When a critical error occurs, it is assigned this unique ID known as an incident number. Diagnostic data for the error (such as log files) is immediately captured and tagged with this number.

The data is then stored in the Automatic Diagnostic Repository (ADR). ADR is a file-system repository for cataloging occurrences of failures and storage of associated diagnostic data. The data is retrieved by incident package number, formatted, viewed with Oracle database tools such as ADRCI, and analyzed.

ADRCI enables you to view the names of the dump files and the alert log with XML tags stripped. This viewing enables you to investigate problems, and package and upload first-failure diagnostic data to Oracle Support Services.

You can also use the Diagnostic Framework WLST commands to perform the following tasks:

- Query problems
- View incident dump files
- Create manual incidents
- Manually execute dumps

The Diagnostic Framework is supported on all JRF-supported platforms.

The Diagnostic Framework automatically includes several dumps. For information about these dumps, see Section "Investigating, Reporting, and Solving a Problem" of *Oracle Fusion Middleware Administrator's Guide*.

In addition to these dumps, several Oracle SOA Suite dumps are also supported. For information about Oracle SOA Suite dumps, see Section 11.1.4, "Supported Oracle SOA Suite Diagnostic Dumps."

For more information about solving problems, incidents, and WLDF and Diagnostic Framework integration, see Chapter "Diagnosing Problems" of *Oracle Fusion Middleware Administrator's Guide*.

For more information about ADR, see Section 11.4, "Viewing Incident Packages with ADR Tools."

11.1.2.1 Controlling the Number of Incident Packages

If you have a recurring problem in Oracle SOA Suite, this can cause for the creation of multiple incident packages. To prevent the server from being overloaded when many failures are occurring, the Diagnostic Framework automatically flood controls some incidents. To avoid this problem, you can configure the Diagnostic Framework to control the number of incident packages. For more information, see Section "Configuring the Diagnostic Framework" of *Oracle Fusion Middleware Administrator's Guide*.

11.1.3 Predefined Incident Processing Rules

When you create a watch in the Oracle WebLogic Server Administrator's Console, you also define a notification. A notification named **FMWDFW notification** is automatically available for selection. While you can create your own notifications, Oracle recommends that you select **FMWDFW notification** because it creates the Oracle SOA Suite dumps described in Section 11.1.4, "Supported Oracle SOA Suite Diagnostic Dumps."

When an error is detected, the FMWDFW notification handler creates an incident and the Diagnostic Framework takes over incident processing semantics. These semantics are controlled by incident processing rules. The incident processing rules are defined in an XML file and loaded and registered with the Diagnostic Framework during SOA Infrastructure startup.

If you encounter scenarios different from those listed in Table 11–1, you must work with Oracle Support Services to obtain a copy of customized incident processing rules and follow the directions related to registration and server restart.

Oracle Support Services can place a customized rules file (named `custom-rules.xml`) under the SOA Server instance home directory (for example, *`FMW_HOME`*`/user_projects/domains/`*`domain_name`*`/servers/`*`server_name`*). Example 11–1 shows a sample custom rules file in which when an `ERROR` level message is detected in the `*-diagnostic.log` from the `oracle.soa.bpel.engine.ws` module, the `soa.composite.trail` dump is executed.

Example 11–1 Sample Custom Rules File

```
<?xml version="1.0" encoding="UTF-8"?>
<diagnosticRules xmlns="http://www.oracle.com/DFW/DiagnosticsFrameworkRules"
 xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">
<logDetectionConditions>
<condition module="oracle.soa.bpel.engine.ws"/>
</logDetectionConditions>

<defaultActions>
<dumpAction name="soa.composite.trail">
<argument name="ecid" value="ECID" valueType="Fact" mandatory="true"/>
</dumpAction>
</defaultActions>

</diagnosticRules>
```

11.1.4 Supported Oracle SOA Suite Diagnostic Dumps

In addition to the diagnostic dumps available with Oracle WebLogic Server, Oracle SOA Suite supports the creation of the diagnostic dumps shown in Table 11–3.

Table 11–3 Supported Oracle SOA Suite Diagnostic Dumps

Dump	Description
`soa.env`	Runtime environment dumps
`soa.config`	Runtime platform configuration dumps
`soa.db`	Database dumps
`soa.composite`	Deployed composite metadata dumps
`soa.composite.trail`	Instance audit trail dumps
`soa.edn`	Event dumps
`soa.wsdl`	Deployed composite WSDL/schema cache dumps

The Diagnostic Framework outputs and records the diagnostic dumps. You can list details about all the supported diagnostic dumps with the WLST `listDumps` and `describeDump` commands.

> **Note:** You must start `WLST` from *`MW_HOME`*`/oracle_common/common/bin`. Otherwise, the ODF functions are missing.

To list the supported dumps:

1. Connect to the server on which the SOA Infrastructure is installed.

```
wls:/offline> connect('user_name', 'password',
't3://myhost:8001')
Connecting to t3://myhost:8001 with userid user_name ...
Successfully connected to managed Server 'soa_server1' that belongs to
domain 'soainfra'.
```

2. List the supported Diagnostic Framework dumps.

```
wls:/soainfra/serverConfig> listDumps()
odl.activeLogConfig
jvm.classhistogram
dms.ecidctx
wls.image
odl.logs
dms.metrics
odl.quicktrace
http.requests
jvm.threads

Use the command describeDump(name=<dumpName>) for help on a specific dump.
```

3. List the supported Oracle SOA Suite dumps.

```
wls:/soainfra/serverConfig> listDumps(appName='soa-infra')
adf.ADFConfigDiagnosticDump
soa.composite
soa.composite.trail
soa.config
soa.db
soa.edn
soa.env
soa.wsdl

Use the command describeDump(name=<dumpName>) for help on a specific dump.
```

The supported Oracle SOA Suite dumps are described in the following subsections.

For more information about `listDumps` and `describeDump`, see Chapter "Diagnostic Framework Custom WLST Commands" of *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

For more information about Diagnostic Framework dumps, see Chapter "Diagnosing Problems" of *Oracle Fusion Middleware Administrator's Guide*.

11.1.4.1 Supported Runtime Environment Diagnostic Dumps (soa.env)

Table 11–4 provides details about supported runtime environment diagnostic dumps.

Table 11–4 Runtime Environment Diagnostic Dumps

Dump Name	Dump Parameters/Dump Mode	Information Captured
`soa.env`	■ Dump parameters: None ■ Dump Mode: `ASYNC_SYNC`	■ SOA runtime version, label (can be obtained from the Discovery MBean), and topology (information about the cluster of which the runtime version is a member). ■ Topology: Cluster and Oracle Coherence information such as cluster name, member name, whether the cluster is the leader, local members, machine ID, rack ID, and so on. The leader is generally the oldest node in the cluster. This may change over time as members leave and join the cluster. This senior member is responsible for maintaining cluster membership and making other decisions for the cluster. It also acts as the final arbiter in various protocols, such as the panic protocol. ■ Oracle Coherence messaging mode: Either unicast or multicast.

11.1.4.1.1 WLST Command Dump Description and Execution

1. Enter the following WSLT command line syntax to display a dump description and execute a dump of `soa.env`.

```
wls:/soainfra/serverConfig> describeDump(name='soa.env',appName='soa-infra')
Name: soa.env
Description: SOA Dump for runtime environment info: SOA runtime version, label,
topology. ...
Mandatory Arguments:
Optional Arguments:

wls:/soainfra/serverConfig> executeDump(name='soa.env',appName='soa-infra')
Begin dumping EMInstanceProperties
key=soaClusterName
value=NULL
key=label
value=PCBPEL_11.1.1.6.0_GENERIC_111018.1315.1477
key=CompositeLabel
value=NULL
key=version
value=11.1.1.6.0
key-HttpUrl
value=http://myhost.us.oracle.com:8001
End dumping EMInstanceProperties
Cluster Info:
-----------------------------------------------
cluster name:null
cluster is leader:true
cluster is standalone:true
```

11.1.4.2 Supported Runtime Platform Configuration Diagnostic Dumps (soa.config)

Table 11–5 provides details about supported runtime platform configuration diagnostic dumps.

Table 11–5 Runtime Platform Configuration Diagnostic Dumps

Dump Name	Dump Parameters/Dump Mode	Information Captured
`soa.config`	■ Dump parameters: `zip`: (Optional) Supports the following values: - `true`: (Default value) Zips the output file and artifacts into one ZIP file. - `false`: Writes the dump text file and artifacts to the dump path location without compressing them into one ZIP file. `output`: (Optional) Specifies the alternate directory location to which to write dump files. If not specified, the diagnostic dump uses the Diagnostic Framework dump path. ■ Dump Mode: `ASYNC_SYNC`	`deployed-composites.xml` - A catalog of deployed composites, including their revisions. Service engine configurations: The following configurations are persisted in MDS (`soa/configuration/default/*.xml`): ■ `adapter-config.xml` ■ `b2b-config.xml` ■ `bpel-config.xml` ■ `bpmn-config.xml` ■ `businessrules-config.xml` ■ `cep-config.xml` ■ `edn-config.xml` ■ `mediator-config.xml` ■ `soa-infra-config.xml` ■ `workflow-config.xml` ■ `workflow-identity-config.xml` ■ `workflow-notification-config.xml`

11.1.4.2.1 WLST Command Dump Description and Execution

1. Enter the following WSLT command line syntax to display a dump description and execute a dump of `soa.config`:

```
wls:/soainfra/serverConfig> describeDump(name='soa.config',appName='soa-infra')
Name: soa.config
Description: Description: SOA dump that captures SOA runtime configuration
                          info, such as deployed-composites.xml, service engine
                          configurations, etc.
Mandatory Arguments:
Optional Arguments:
   Name    Type     Description
   zip     BOOLEAN  Boolean flag (default to 'true') indicating if the dump
                    files need to be compressed into one single zip file,
                    true - dump into a single zip, false, otherwise.
   output  STRING   Path to output destination, a directory where the dump
                    content will be written to, default to DFW dump path if
                    not specified ...

wls:/soainfra/serverConfig> executeDump(name='soa.config',appName='soa-infra')

Start dumping artifacts from MDS
URI:deployed-composites/deployed-composites.xml to /tmp/soa_
dumps139780820495.tmp/soa_config734535543.d

Finished dumping specified MDS metadata to /tmp/soa_dumps139780820495.tmp/soa_
config734535543.d

Start dumping artifacts from MDS URI:soa/configuration/default/ to /tmp/soa_
dumps139780820495.tmp/soa_config734535543.d/se-configurations

Finished dumping specified MDS metadata to: /tmp/soa_dumps139780820495.tmp/soa_
config734535543.d/se-configurations
```

```
MDS artifacts dumped, next, compressing all dump files into:
/tmp/oracle-dfw--1770056289547048694.tmp/soa_config364634563344231671.zip
```

The `executeDump` command dumps `deployed-composites.xml` from the MDS repository and service engine configurations for all installed service engines into a single, compressed ZIP file (`soa_config364634563344231671.zip`).

2. Enter the following WSLT command line syntax to execute a dump of `soa.config` with the `zip` parameter set to `false`. This setting writes the dump text file and artifacts to the dump path location without compressing them into one ZIP file.

```
wls:/soainfra/serverConfig> executeDump(name='soa.config',appName='soa-infra',
args={'zip':'false'})
Start dumping artifacts from MDS
 URI:deployed-composites/deployed-composites.xml to:
 /tmp/oracle-dfw-7178460573556479044.tmp/soa_config199325881615155981.d
Finished dumping specified MDS metadata to:
/tmp/oracle-dfw-7178460573556479044.tmp/soa_config199325881615155981.d
Start dumping artifacts from MDS URI:soa/configuration/default/ to:
/tmp/oracle-dfw-7178460573556479044.tmp/soa_
config199325881615155981.d/se-configurations
Finished dumping specified MDS metadata to:
/tmp/oracle-dfw-7178460573556479044.tmp/soa_
config199325881615155981.d/se-configurations
```

3. Examine the contents under the default dump path:

```
[jdoe@myhost /tmp]$ ls -alR oracle-dfw-7178460573556479044.tmp
oracle-dfw-7178460573556479044.tmp:
total 52
drwxr-----   3 jdoe dba   4096 Oct 24 15:43 .
drwxrwxrwt 104 root root 36864 Oct 24 15:37 ..
drwxr-----   4 jdoe dba   4096 Oct 24 15:43 soa_config199325881615155981.d
-rw-r-----   1 jdoe dba    561 Oct 24 15:43 soa_config199325881615155981.txt

oracle-dfw-7178460573556479044.tmp/soa_config199325881615155981.d:
total 16
drwxr----- 4 jdoe dba 4096 Oct 24 15:43 .
drwxr----- 3 jdoe dba 4096 Oct 24 15:43 ..
drwxr----- 2 jdoe dba 4096 Oct 24 15:43 deployed-composites
drwxr----- 2 jdoe dba 4096 Oct 24 15:43 se-configurations

oracle-dfw-7178460573556479044.tmp/soa_
config199325881615155981.d/deployed-composites:
total 12
drwxr----- 2 jdoe dba 4096 Oct 24 15:43 .
drwxr----- 4 jdoe dba 4096 Oct 24 15:43 ..
-rw-r----- 1 jdoe dba 1437 Oct 24 15:43 deployed-composites.xml

oracle-dfw-7178460573556479044.tmp/soa_
config199325881615155981.d/se-configurations:
total 56
drwxr----- 2 jdoe dba 4096 Oct 24 15:43 .
drwxr----- 4 jdoe dba 4096 Oct 24 15:43 ..
-rw-r----- 1 jdoe dba  267 Oct 24 15:43 adapter-config.xml
-rw-r----- 1 jdoe dba  425 Oct 24 15:43 b2b-config.xml
-rw-r----- 1 jdoe dba 2040 Oct 24 15:43 bpel-config.xml
-rw-r----- 1 jdoe dba 1525 Oct 24 15:43 bpmn-config.xml
-rw-r----- 1 jdoe dba  895 Oct 24 15:43 businessrules-config.xml
-rw-r----- 1 jdoe dba  119 Oct 24 15:43 cep-config.xml
-rw-r----- 1 jdoe dba  215 Oct 24 15:43 edn-config.xml
```

```
-rw-r----- 1 jdoe dba  836 Oct 24 15:43 mediator-config.xml
-rw-r----- 1 jdoe dba 1148 Oct 24 15:43 soa-infra-config.xml
-rw-r----- 1 jdoe dba 2693 Oct 24 15:43 workflow-config.xml
-rw-r----- 1 jdoe dba 2146 Oct 24 15:43 workflow-identity-config.xml
-rw-r----- 1 jdoe dba  605 Oct 24 15:43 workflow-notification
```

11.1.4.3 Supported Database Diagnostic Dumps (soa.db)

Table 11–6 provides details about supported database diagnostic dumps. The types of database information captured include data shape information such as counts by state and growth patterns for Oracle SOA Suite schemas and the MDS schema.

Table 11–6 Database Diagnostic Dumps

Dump Name	Dump Parameters/Dump Mode	Information Captured
`soa.db`	▪ Dump parameters: None ▪ Dump Mode: `ASYNC_SYNC`	BPEL database table growth data using JDBC-based access to execute the query and dump the result: **SELECT** dt.table_name table_name,ds.bytes/1024/1024 segment_size_mb, ds.extents extents_used, dt.num_rows total_rows, to_char(dt.last_analyzed,'YYYY-MM-DD HH24:MI:SS') last_analyzed_date **FROM** dba_segments ds, dba_tables dt **WHERE** dt.owner = ds.owner and dt.owner = '*schema_user_name*' and dt.tablespace_name = ds.tablespace_name and dt.table_name = ds.SEGMENT_NAME and ds.segment_type = 'TABLE' and dt.table_name in ('CUBE_INSTANCE', 'MEDIATOR_CASE_INSTANCE','COMPOSITE_INSTANCE', 'AUDIT_TRAIL', 'WORK_ITEM', 'DLV_MESSAGE', 'XML_DOCUMENT','DOCUMENT_CI_REF')

11.1.4.3.1 WLST Command Dump Description and Execution

1. Enter the following WSLT command line syntax to display a dump description and execute a dump of `soa.db`:

```
wls:/soainfra/serverConfig> describeDump(name='soa.db',appName='soa-infra')
Name: soa.db
Description: SOA Dump for SOA runtime DB usage info.
Mandatory Arguments:
Optional Arguments:

wls:/soainfra/serverConfig> executeDump(name='soa.db',appName='soa-infra')
EXECUTE QUERY
SELECT dt.table_name table_name,ds.bytes/1024/1024 segment_size_mb, ds.extents
extents_used, dt.num_rows total_rows, to_char(dt.last_analyzed,
'YYYY-MM-DD HH24:MI:SS') last_analyzed_date FROM dba_segments ds,
dba_tables dt WHERE dt.owner = ds.owner and dt.owner ='schema_user_name'
and dt.tablespace_name = ds.tablespace_name and dt.table_name = ds.SEGMENT_NAME
and ds.segment_type = 'TABLE' and dt.table_name in ('CUBE_INSTANCE',
'MEDIATOR_CASE_INSTANCE','COMPOSITE_INSTANCE', 'AUDIT_TRAIL', 'WORK_ITEM',
'DLV_MESSAGE','XML_DOCUMENT','DOCUMENT_CI_REF')
COMMENT: BPEL DB GROWTH DATA.
TABLE_NAME   SEGMENT_SIZE_MB  EXTENTS_USED TOTAL_ROWS LAST_ANALYZED_DATE
--------------------------------------------------------------------------
ResultSet Metadata:weblogic.jdbc.wrapper.ResultSetMetadata_oracle_jdbc_driver_
OracleResultSetMetadata@76a4b0
RECORD #1:
TABLE_NAME: COMPOSITE_INSTANCE
SEGMENT_SIZE_MB:0
EXTENTS_USED:1
```

```
TOTAL_ROWS:9
LAST_ANALYZED_DATE:2011-09-30 22:02:25
RECORD #2:
TABLE_NAME:DOCUMENT_CI_REF
SEGMENT_SIZE_MB:0
EXTENTS_USED:1
TOTAL_ROWS:0
LAST_ANALYZED_DATE:2011-09-30 22:02:34
RECORD #3:
TABLE_NAME:WORK_ITEM
SEGMENT_SIZE_MB:0
EXTENTS_USED:1
TOTAL_ROWS:0
LAST_ANALYZED_DATE:2011-09-30 22:02:36
RECORD #4:
TABLE_NAME:AUDIT_TRAIL
SEGMENT_SIZE_MB:0
EXTENTS_USED:1
TOTAL_ROWS:8
LAST_ANALYZED_DATE:2011-09-30 22:02:39
```

This dump shows the query string and records from the result set.

11.1.4.4 Supported Deployed Composite Metadata Diagnostic Dumps (soa.composite)

Table 11–7 provides details about supported deployed composite metadata diagnostic dumps. The types of information captured include the current composite processed when an incident occurs, MDS artifact references (for example, namespace exports), and abnormal transactions.

Table 11–7 Deployed Composite Metadata Diagnostic Dumps

Dump Name	Dump Parameters/Dump Mode	Information Captured
soa.composite	■ Dump parameters: compositeName: (Optional) Composite name that includes MDS recorded artifacts to dump. If a value is not specified (null or blank), compositeName assumes a wild card (*). partition: (Optional) Partition name in which the composite is deployed. If not specified, the partition of the default composite specified in the deployed-composites.xml file is assumed (for example, default, partition_01, and my_partition). revision: (Optional) Composite revision that includes MDS recorded artifacts to dump. If not specified, the revision of the default composite as specified in the deployed-composites.xml file is assumed. A wild card (*) to match any revision is supported. zip: (Optional) Whether or not to compress the dump output into a ZIP file. The following values are supported: - true: (Default value) Compresses dump files into one ZIP file. - false: Writes dump to a text file and artifacts to the dump path location without compressing them into one ZIP file. - output: (Optional) Alternate directory location to which to write dump files. If not specified, the diagnostic dump uses the Diagnostic Framework dump path. ■ Dump Mode: ASYNC_SYNC	Per composite metadata from MDS: ■ All the MDS recorded artifacts for the specified composites. ■ The text dump file contains logging information about which composite's MDS artifacts are dumped and to where. ■ All dump files compressed into one ZIP file. ■ User-specified output file location for WLST use. ■ One or more generated scratch_entries.txt files that contain the listings of file and directory entries of the scratch area for the composite. The location of the scratch area is indicated by the composite-revision element in deployed-composites/deployed-composites.xml.

11.1.4.4.1 WLST Command Dump Description and Execution

1. Enter the following WSLT command line syntax to display a dump description and execute a dump of soa.composite:

```
wls:/soainfra/serverConfig> describeDump(name='soa.composite'
,appName='soa-infra')
Name: soa.composite
Description: SOA Dump that captures artifacts from MDS, and listing of entries
in each composite's scratch area, dump MDS artifacts for composites that
matches specified composite name, specified partition, specified revision, also
support wild card '*' as composite name/partition/revision, meaning dump MDS
artifacts for composites with any composite name/partition/revision.
Mandatory Arguments:
Optional Arguments:
  Name        Type      Description
  zip         BOOLEAN   Boolean flag (default to 'true') indicating if the dump
                        files need to be compressed into one single zip file,
                        true - dump into a single zip file, false, otherwise.
  revision    STRING    composite revision, e.g. '1.0', '2.0', or wild card
                         '*'- will match any revision.
  partition   STRING    partition of the composite, default is 'default',
                        support wild card '*' as partition name, meaning any
```

```
                        partition.
  compositeName STRING composite name, e.g., 'OrderProcessing', or wild card
                         '*' - dump MDS recorded artifacts for all the deployed
                        composites in the specified partition.
   output      STRING   Path to output destination, a directory where the
                        dump content will be written to, default to DFW dump
                        path if not specified.

wls:/soainfra/serverConfig> executeDump(name='soa.composite',
 appName='soa-infra'),arg=('compositeName':'WSInMedPubBpelSubFileOut',
'revision':'1.0,'partition':'partition_1','output'':'/home/myhome//COMPOSITE_
DUMP_
DIR_3'})
Start dumping composite [name=WSInMedPubBpelSubFileOut, partition=partition_1,
revision= 1.0 ] MDS metadata to :/tmp/soa_dumps1019331262763443905.tmp/
soa_composite4533356433976235311.d/partition_1/WSInMedPubBpelSubFileOut_
rev1.0
Finished dumping composite MDS metadata to :/tmp/soa_dumps
1019331262763443905.tmp/soa_composite4533356433976235311.d/partition_
1/WSInMedPubBpelSubFileOut_rev1.0
```

In this example, no `zip` parameter is specified. Because the default value is `true`, a single dump output file is created at the dump location `/home/myhome//COMPOSITE_DUMP_DIR_3`.

11.1.4.5 Supported Instance Audit Trail Diagnostic Dumps (soa.composite.trail)

Table 11–8 provides details about supported instance audit trail diagnostic dumps. The type of information captured includes the deployed composite instance audit trail for a given ECID.

Table 11–8 Instance Audit Trail Diagnostic Dumps

Dump Name	Dump Parameters/Dump Mode	Information Captured
`soa.composite.trail`	▪ Dump parameters: `ecid` (mandatory). The execution content ID. If not specified, a `DumpExecutionException` error is thrown. ▪ Dump Mode: `ASYNC_SYNC`	▪ The audit trail associated with the ECID (per composite) (therefore, all of its components). This is for the execution of message routing triggered by an inbound message. The audit trail is an XML CLOB obtained by calling the following Facade API in the current runtime context: `oracle.soa.management.facade.Locator.getAuditTrail(ECID ecid)` ▪ Composite instances information. ▪ Component instances information for each composite instance. ▪ EDN information from the Facade API. ▪ Sensor instances information for each composite instance.

11.1.4.5.1 WLST Command Dump Description and Execution

1. Enter the following WSLT command line syntax to display a dump description and execute a dump of `soa.composite.trail`:

```
wls:/soainfra/serverConfig> describeDump(name='soa.composite.trail'
,appName='soa-infra')
Name: soa.composite.trail
Description: Composite instance audit trail dump for a given ECID
```

```
Mandatory Arguments:
   Name        Type       Description
   ecid        STRING     ECID from an incident
Optional Arguments:

wls:/soainfra/serverConfig> executeDump(name='soa.composite.trail',
appName='soa-infra',args=('ecid':'cf1cec292805f383.236c6d88:132bc9b7486:-8000=0
000000000121b'})
BEGIN----------------AUDIT TRAIL::: ECID:cf1cec292805f383.236c6d88:132bc9b7486
:-8000=0000000000121b++++++++++++++++++++++++++++++++++++++++++++++++++++++++
<audit_trail ecid="cf1cec292805f383.236c6d88:132bc9b7486:-8000=0000000000121b">
<entry instanceId="service:10001" date="20-11-09-39T16:21:46"
timestamp="1317424906775" state="2" actionType="operation" actionName="execute"
subType="binding.ws" type="service"><entityName>MyInputService</entityName>
<compositeInstanceId>10001</compositeInstanceId><compositeName>WSInMedPubBpelSu
bFileOut</compositeName><compositeDN>default/WSInMedPubBpelSubFileOut!1.0
</compositeDN><applicationName>default</applicationName></entry><entry
instanceId="mediator:F7BF03D8EBBA11E0BFEEFB426F4C514D"parentInstanceId=
"service:10001" date="2011-0930T16:21:48"timestamp="131742908689"
state="2" <type="component"subType="mediator">entityName>MediatorPub
</entityName><compositeInstanceId>10001</compositeInstanceId><compositeName>
WSInMedPubBpelSubFileOut</compositeName><compositeDN>default/
WSInMedPubBpelSubFileOut!1.0</compositeDN><applicationName>default
</applicationName></entry></audit trail>
END--------------AUDIT TRAIL:::
ECID:cf1cec292805f383.236c6d88:132bc9b7486:-8000=0000000000121b

BEGIN------------------FACADE(EDN) +++++++++++++
oracle.soa.management.internal.facade.edn.EDNImpl@14b5caec
END------------------FACADE(EDN) +++++++++++++
BEGIN------------------FACADE(Composite Instances) +++++++++++++
DN: default/WSInMedPubBpelSubFileOut!1.0 > Instance: 10001 creation-date: Fri
Sep 30 16:21:46 PDT 2011 state {-1}: STATE_UNKNOWN
-> componentinstance: MediatorPub type: mediator state: STATE_COMPLETED_
SUCCESSFULLY
DN: partition_3/WSInMedPubBpelSubFileOut!1.0 > Instance: 8 creation-date: Thu
Sep 29 15:04:13 PDT 2011 state (-1): STATE_UNKNOWN
-> componentinstance: BpelSub type: bpel state: STATE_SUSPENDED
```

For information about obtaining the ECID, see Section 13.1, "Monitoring the Audit Trail and Process Flow of a BPEL Process Service Component."

11.1.4.6 Supported Event Diagnostic Dumps (soa.edn)

Table 11–9 provides details about supported event diagnostic dumps. The types of information captured include Oracle Streams Advanced Queuing (SAQ) logging records and event metadata among currently deployed applications.

Table 11–9 Event Diagnostic Dumps

Dump Name	Dump Parameters/Dump Mode	Information Captured
soa.edn	■ Dump parameters: dumpEDNDBLOG: Supports the following values: - true: Dumps all records logged during AQ stored procedure calls (for debugging purposes). - false: Does not dump EDN database log messages. ■ Dump Mode: ASYNC_SYNC	■ EDN event bus state variables. ■ EDN subscriber/publisher information for all deployed composites, and so on. All information is written to the dump text file.

11.1.4.6.1 WLST Command Dump Description and Execution

1. Enter the following WSLT command line syntax to display a dump description and execute a dump of soa.edn:

```
wls:/soainfra/serverConfig> describeDump(name='soa.edn',appName='soa-infra')
Name: soa.edn
Description: SOA Dump for EDN BusinessEventBus status info, and EDN DB Log
records.
Mandatory Arguments:
Optional Arguments:
   Name         Type       Description
   dumpEDNDBLOG BOOLEAN    SOA Dump for EDN BusinessEventBus status info, and
                           EDN DB Log records.

wls:/soainfra/serverConfig> executeDump(name='soa.edn',appName='soa-infra')

Type:oracle.integration.platform.blocks.event.saq.SAQBusinessEventBus
Configuration:null
Status: running=true started=true
ThreadCount:3
RetryCount:3
In Global: Tx:false
Cluster Info:oracle.integration.platform.blocks.cluster.CoherenceCluster
Interface1mpt@163bd717
SharedEDN:false
OOAO Queue Name:edn_ooao_queue
Java Subscriber Name:edn java subscriber
Subscription Info:
No namespace subscription...
QName subscriptions:
==============================================================
qname={http://schemas.oracle.com/events/edl/ActionOccur}ADEvent
subscriptions=
id=default/WSInMedPubBpelSubFileOut!1.0*soa_
7a055d6a-8402-49c2-ac56-5f85cbf3d7f/
BpelSub, consistencyLevel=ONE_AND_ONLY_ONE, filter=XPath Filter: starts-with(/
be:business-event/be:content/ns0:ActionOccurrence/ns0:ParentEntityType/@value,
 'A'), runAsRoles=[$publisher]
id=partition_1/WSInMedPubBpelSubFileOut!1.0*soa_
80a169ab1-395a-4b87-9986-9fa2742a8bd3/
BpelSub, consistencyLevel=ONE_AND_ONLY_ONE, filter=XPath Filter: starts-with(/
be:business-event/be:content/ns0:ActionOccurrence/ns0:ParentEntityType/@value,
```

```
 'A'), runAsRoles=[$publisher]
EventThreadContextInfo:
EventTargets:
Event:partition_
1/WSInMedPubBpelSubFileOut!1.0*soa0a169ab1-395a-4b87-9986-9fa2742a9bd3/
BpelSub:::oracle.fabric.BPELServiceEngine@163bd6b5
Event:default/WSInMedPubBpelSubFileOut!1.0*soa7a055d6a-8402-49c2-ac56-5f85cbf3
d7f/BpelSub:::oracle.fabric.BPELServiceEngine@163bd6b5
EDN DB Log enabled:false
```

The `executeDump` command dumps the event bus status variables (`1mpt@163bd717`) and event publish and subscription information. No log messaging information is dumped because the `edn-db-log` is set to `false`. You can enable the `edn-db-log` at the following URL:

`http://`*`host_name`*`:`*`port`*`/soa-infra/events/edn-db-log`

Notes:

- You must have the administrator privilege to enable/disable the `edn-db-log`.
- Always disable the `edn-db-log` after a debugging session to disable logging. This prevents excessive database growth in the EDN database log table. If the `edn-db-log` remains enabled, then debugging messages related to events that are published/enqueued into the database and subscribed to/dequeued from the database continue to be persisted into certain EDN database log tables. This causes the table to grow indefinitely.

11.1.4.7 Supported Deployed Composite WSDL/Schema Cache Diagnostic Dumps (soa.wsdl)

Table 11–10 provides details about supported deployed composite WSDL/schema cache diagnostic dumps.

Table 11–10 Deployed Composite WSDL/Schema Cache Diagnostic Dumps

Dump Name	Dump Parameters/Dump Mode	Information Captured
`soa.wsdl`	■ Dump parameters: `compositeName`: (Optional) Composite name that includes key service definition information (WSDLs) to dump, including WSDLs referenced and XSDs referenced. If a value is not specified (null or blank), `compositeName` assumes a wild card (`*`). `partition`: (Optional) Partition name in which the composite is deployed. If not specified, the partition of the default composite specified in the `deployed-composites.xml` file is assumed. A wild card (`*`) to match any composite is supported. `revision`: (Optional) Composite revision that includes the service definition information (from WSDLs) to dump. If not specified, the revision of the default composite as specified in the `deployed-composites.xml` file is assumed. A wild card (`*`) to match any revision is supported. ■ Dump Mode: `ASYNC_SYNC`	■ Composite distinguished name (DN). For example: `compositeDN:partition_1/WSInMedPubBpelSubFileOut!1.0*soa_8a169ab1-395a-4b87-9986-9fa2742a8bd3`. ■ Is it the default in the series ■ Composite name ■ Composite state (on or off) ■ Composite mode (active or retired) ■ The qualified name and the target namespace for all service definitions (including those from shared WSDLs): `javax.wsdl.Definition` objects: - Service name: QName `javax.wsdl.Definition.getQName()` - Target namespace: `javax.wsdl.Definition.getTargetNamespace()` ■ `SchemaManager` state variables: - `SchemaManager.isPostDeploy()` - `SchemaManager.isShared()` - `SchemaManager.schemaAddedSinceLastBuild()` ■ XML schema definitions referenced by service definitions: - The message type QName - The message type `SchemaTargetNamespace` - The message type `TargetNS`

11.1.4.7.1 WLST Command Dump Description and Execution

1. Enter the following WSLT command line syntax to display a dump description and execute a dump of `soa.wsdl`:

```
wls:/soainfra/serverConfig> describeDump(name='soa.wsdl',appName='soa-infra')

Name: soa.wsdl
Description: SOA Dump that captures service definition info caches for
             composite(s) that matches specified composite name, specified
             partition, specified revision, also support wild card '*' as
             composite name/partition/revision, meaning dump service
             definitions WSDL/XSD caches for composites with any composite
             name/partition/revision.
Mandatory Arguments:
Optional Arguments:
   Name        Type       Description
   revision    STRING     composite name, e.g. '1.0', '2.0', or wild card '*'
                          - will match any revision.
   partition   STRING     partition of the composite, default is 'default',
                          support wild card '*' as partition name, meaning
                          any partition.
   compositeName STRING  composite name, e.g., 'OrderProcessing', or wild card
                          '*' - dump cache for all the deployed composites in
                          the specified partition.
```

```
wls:/soainfra/serverConfig> executeDump(name='soa.wsdl',appName='soa-infra'
,arg=('compositeName':'WSInMedPubBpelSubFileOut',
'revision':'1.0','partition':'partition_1'})compositeDN:partition_
1/WSInMedPubBpelSubFileOut!1.0*soa_8a169ab1-2060-4b87-9986-9fa2742a8bd3
----------------------------------------------------------------------
is default in series:true
composite application name:partition_1
composite name:WSInMedPubBpelSubFileOut
composite state:on
composite mode:active
WSDLManager Instance:oracle.fabric.common.wsdl.WSDLManager@164ec201
Cached wsdl Definitions for the specified composite: including any shared
wsdls if there are any:
key=http://xmlns.oracle.com/bpel/workflow/taskService
value=QName:{http://xmlns.oracle.com/bpel/workflow/taskService}
TaskServicetarget name space:http://xmlns.oracle.com/bpel/workflow/taskService
key=urn:MyTestWsdl
value=QName:{urn:MyTestWsdl}target name space:urn:MyTestwsdl
key=http://xmlns.oracle.com/adapter/file/LixinSandboxes/WSInMedPubBpelSubFile
Out/FileWrite
value=QName:{http://xmlns.oracle.com/pcbpel/adapter/file/LixinSandboxes/
WSInMedPubBpelSubFileOut/FileWrite}FileWritetarget name space:http:
 //xmlns.oracle.com/pcbpel/adapter/file/LixinSandboxes/
WSInMedPubBpelSubFileOut/FileWrite
count of all cached entries:3
SchemaManager instance:oracle.fabric.common.wsdl.SchemaManager@164ec1f8
SchemaManager.isPostDeploy():true
SchemaManager.isShared():false
SchemaManager.schemaAddressSinceLastBuild():false
Cached schema count:0
WSDLManager instance:oracle.fabric.common.wsdl.WSDLManager@14b5b8b0
cached wsdl definitions for the specified composite: including shared wsdls
if there is any:
--------------------------------------------------------------------------
key=http://xmlns.oracle.com/bpel/workflow/taskService
value=QName:{http://xmlns.oracle.com/bpel/workflow/taskService}
TaskServicetarget name space:http://xmlns.oracle.com/bpel/workflow/taskService
count of all cached entries:1
SchemaManager instance:oracle.fabric.common.wsdl.SchemaManager@145b8cc
SchemaManager.isPostDeploy():false
SchemaManager.isShared():true
SchemaManager.schemaAddressSinceLastBuild():true
Cached schema count:0

WSDLManager instance:oracle.fabric.common.wsdl.WSDLManager@14b58b0
WSDLManager instance:oracle.fabric.common.wsdl.WSDLManager@14b58b0
cached wsdl definitions for the specified composite: including shared wsdls
if there is any:
-------------------------------------------------

key=http://xmlns.oracle.com/bpel/workflow/taskService
value=QName:{http://xmlns.oracle.com/bpel/workflow/taskService}
TaskServicetarget name space:http://xmlns.oracle.com/bpel/workflow/taskService
count of all cached entries:1
SchemaManager instance:oracle.fabric.common.wsdl.SchemaManager@145b8cc
SchemaManager.isPostDeploy():false
SchemaManager.isShared():true
SchemaManager.schemaAddressSinceLastBuild():true
Cached schema count:0
```

11.1.5 Supported DMS Metrics

DMS metrics with noun types are exposed as Oracle SOA Suite MBeans to use for diagnosing problems. This section describes the supported DMS metrics.

Table 11–11 shows the supported service engine sensors.

Table 11–11 Service Engine Sensors

Noun Path	Noun Type	Sensor	Type
`/soainfra/engines/[bpel|workflow|mediator|decision]/message_processing`	`soainfra_message_processing`	`faultedRequestProcessingTime`	Phase Event
		`faultedPostProcessingTime`	Phase Event
		`requestProcessingTime`	Phase Event
		`postProcessingTime`	Phase Event
		`activeRequest`	State
`/soainfra/engines/bpel/requests/[REQUEST_TYPE]`		`active`	State
		`scheduled`	State
`/soainfra/engines/workflow/Task/service`	`soainfra_wfRequest`	`time`	Phase
		`count`	Event
`/soainfra/engines/workflow/Task/[METHOD_NAME]`		`time`	Phase
		`count`	Event
`/soainfra/engines/workflow/TaskQuery/[METHOD_NAME]`		`time`	Phase
		`count`	Event
`/soainfra/engines/workflow/TaskMetadata/getTaskDefinition`		`time`	Phase
		`count`	Event
`/soainfra/engines/workflow/Verification/[METHOD_NAME]` (only methods: `authenticateUser`, `getPermissableTaskActions`, and `canUserPerfomTaskAction`)		`time`	Phase
		`count`	Event
`/soainfra/engines/workflow/TaskNotification/notifyForTask`		`time`	Phase
		`count`	Event
`/soainfra/engines/workflow/AssignmentRules/executeRules`		`time`	Phase
		`count`	Event

Table 11–11 (Cont.) Service Engine Sensors

Noun Path	Noun Type	Sensor	Type
/soainfra/engines/bpel/dispatcher/	soainfra_bpel_dipatcher	maxThreads	State
		avgLifeTime	State
		avgRequestCountPerSecond, and so on	State

Table 11–12 shows the supported binding sensors.

Table 11–12 Binding Sensors

Noun Path	Noun Type	Sensor	Type
/soainfra/bindings/[inbound|outbound]/[ws|sdo|jca|b2b]	soainfra_binding	processRequests	Phase
		requests	Event
		errors	Event

Table 11–13 shows the supported composite sensors.

Table 11–13 Composite Sensors

Noun Path	Noun Type	Sensor	Type
/soainfra/apps/[APP_NAME]/[COMPOSITE_DN]	soainfra_composite	status	State
		instantiated	Event
		successfulInstanceProcessingTime	Phase
		faultedInstanceProcessingTime	Phase
/soainfra/apps/[APP_NAME]/[COMPOSITE_NAME]/[REVERSION]/[COMPONENT_NAME] /decision/[INTERACTION_PATTERN]/[INTERACTION_PATTERN_NAME]	soainfra_decision_interaction	executed	Event
		executionTime	Phase
/soainfra/apps/[COMPOSITE_DN]/[COMPONENT_NAME]/[ACTIVITY_NAME] (for bpel)	soainfra_bpel_activity	started	Event
		executionTime	Phase
		faultedExecutionTime	Phase
		Standard deviation (the user interface calculates this as it comes through)	

Table 11–13 (Cont.) Composite Sensors

Noun Path	Noun Type	Sensor	Type
/soainfra/apps/[APP_NAME]/[COMPOSITE_NAME]/[VERSION]/[COMPONENT_NAME] /state/	soainfra_wfStateEvent	[ASSIGNED |COMPLETED |ERRORED |EXPIRED |SUSPENDED]	Event
/soainfra/apps/[APP_NAME]/[COMPOSITE_NAME]/[VERSION]/[COMPONENT_NAME] /outcome/	soainfra_wfOutcomeEvent	[OUTCOME NAME]	Event
/soainfra/apps/[APP_NAME]/[COMPOSITE_NAME]/[VERSION]/[COMPONENT_NAME] /taskCompletion/	soainfra_wfTaskCompletionTime	time	Phase

Table 11–14 shows the supported reference and service sensors.

Table 11–14 Reference and Service Sensors

Noun Path	Noun Type	Sensor	Type
/soainfra/apps/[APP_NAME]/[COMPOSITE_NAME]/[REVERSION]/[REFERENCE_NAME]	soainfra_reference	processOutboundMessagesEvents	Event
		Errors	Event
		processOutboundMessages	Phase
/soainfra/apps/[APP_NAME]/[COMPOSITE_NAME]/[REVERSION]/[SERVICE_NAME]	soainfra_service	processInboundMessagesEvents	Event
		Errors	Event
		processInboundMessages	Phase

Table 11–15 shows the supported Oracle B2B binding sensors.

Table 11–15 Oracle B2B Binding Sensors

Noun Path	Noun Type	Sensor	Type
/soainfra/bindings/b2b/document_type/[inbound|outbound]/[DOCUMENT_NAME]	soainfra_b2b_document	processMessagesEvents	Event
/soainfra/bindings/b2b/[inbound|outbound]	soainfra_b2b_document_dir		
		processMessagesErrors	Event
		processMessages	Phase Event
		processMessageSize	State
/soainfra/bindings/b2b/trading_partner/[from|to]/[TRADING_PARTNER_NAME]	soainfra_b2b_tradingPartner	processMessagesEvents	Event
/soainfra/bindings/b2b/[from|to]/	soainfra_b2b_tradingPartner_dir		
		processMessagesErrors	Event
		processMessages	Phase Event

Table 11–15 (Cont.) Oracle B2B Binding Sensors

Noun Path	Noun Type	Sensor	Type
		processMessageSize	State
/soainfra/bindings/b2b/endpoint/[inbound|outbound]/[END_POINT]	soainfra_b2b_endpoint	endPointProtocol	State
/soainfra/bindings/b2b/[inbound|outbound]/	soainfra_b2b_endpoint_dir		
		endPointStatus	State
		processMessagesEvents	Event
/soainfra/bindings/b2b/agreement/[AGREEMENT_NAME]	soainfra_b2b_agreement	processMessagesEvents	Event
/soainfra/bindings/b2b/activeEntities	soainfra_b2b_active_entities	activeTradingPartners	State
		activeAgreements	State
		activeDocuments	State
/soainfra/apps/[APP_NAME]/[COMPOSITE_NAME]/[REVERSION]/[SERVICE_NAME]/[TRADING_PARTNER_NAME]	soainfra_service_b2b_tradingPartner	processMessagesEvents	Event
		processMessagesErrors	Event
		processMessages	Phase Event
		processMessageSize	State
/soainfra/apps/[APP_NAME]/[COMPOSITE_NAME]/[REVERSION]/[REFERENCE_NAME]/[TRADING_PARTNER_NAME]	soainfra_reference_b2b_tradingPartner	processMessagesEvents	Event
		processMessagesErrors	Event
		processMessages	Phase Event
		processMessageSize	State

Table 11–16 shows the supported Oracle User Messaging Service sensors.

Table 11–16 Oracle User Messaging Service Event Bridge Metrics

Noun Path	Noun Type	Sensor	Type
/soainfra/eventBridge/rfidBridge	soinfra_rfidBridge	eventsIn	Event
		eventsOut	Event
		eventsProcess	Phase Event
		errors	Event
/soainfra/eventBridge/rfidBridge/device/[SERVER_NAME]/[DEVICE_NAME]	soainfra_rfidBridge_device	eventsIn	Event
		eventsOut	Event

Table 11–16 (Cont.) Oracle User Messaging Service Event Bridge Metrics

Noun Path	Noun Type	Sensor	Type
		`eventsProcess`	Phase Event
		`status`	State
`/soainfra/eventBridge/rfidBridge/server/[SERVER_NAME]`	`soainfra_rfidBridge_server`	`status`	State

11.2 Creating Watches and Notifications

You can create Oracle SOA Suite watches and send notifications around diagnosable conditions based on metrics collected from Oracle SOA Suite MBeans. When a watch expression evaluates to true (for example, heap space exceeds a specified amount), a notification is sent.

The message IDs shown in Table 11–17 have been assigned for diagnostic purposes. When you create a watch in Oracle WebLogic Server Administrator's Console, you must follow these naming conventions. The prefix for Oracle SOA Suite-related watches is `SOA-`*`message_ID`*.

Table 11–17 Message Prefixes

Scenario	Message-ID	Dumps Executed
Memory	`SOA-900000`	▪ `soa.env` ▪ `soa.config` ▪ `java.sysprops`
Deployment hang	`SOA-900001`	▪ `soa.env` ▪ `soa.config`
Data source	`SOA-900002`	▪ `soa.env` ▪ `soa.config` ▪ `soa.db`

You can also link a WLDF notification to the watch. If you link the out-of-the-box Oracle Fusion Middleware Diagnostic Framework notification (named **FMWDFW notification**), then a set of SOA specific dumps are executed. These dumps provide runtime information about the situation and environment. The list of dumps to execute is determined by predefined XML incident rules files.

Other notifications (like email) can also be linked to the watch.

The following watch automatically triggers dumps of `soa.env` and `soa.config`.

To create watches and notifications:

1. Log in to Oracle WebLogic Server Administration Console.

   `http://`*`host`*`:`*`port`*`/console`

2. In the **Domain Structure**, expand **Diagnostics**, and select **Diagnostic Modules**.

   The Summary of Diagnostic Modules page appears.

   You configure a diagnostic system module to monitor an aspect of a server or server resource. You can configure multiple system modules to monitor different aspects of a server, but only one such module can be active on a server.

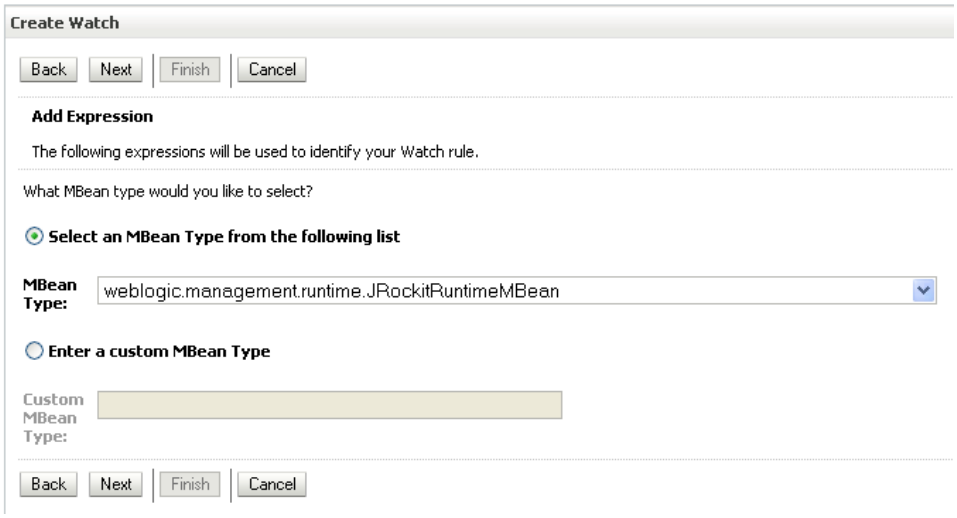
3. In the **Diagnostic System Modules** section, click **Module-[*Module_Name*]** (for example, **Module-FMWDFW**).
4. On the Settings for *Module_Name* page, select **Watches and Notifications** > **Watches**.
5. In the **Watches** section, click **New**.

   The Create a Diagnostic Watch page is displayed.
6. Enter the following details to create a watch, and click **Next**.

Field	Description
Watch Name	Enter a name for the watch (for this example, `SOA-900000#soa-infra#MemoryWatch` is entered). The name of the watch must conform to the following pattern: *`message-id`*`#soa-infra#`*`some_other_text`* For example, `SOA-900000#soa-infra#MemoryWatch`. This is necessary because the watch name coordinates the Diagnostic Framework incident processing actions when watch conditions evaluate to true. Not following this pattern results in Oracle SOA Suite dumps not getting triggered when Oracle SOA Suite incidents are created. For additional information, see Table 11–17. If the watch is set up with the **FMWDFW notification**, the notification handler creates an incident that corresponds to the *`message-id`* specified in the watch name.
Watch Type	Select an option: ▪ **Collected Metrics**: Sets a watch based on metrics collected from MBean attributes. It is recommended that you select this option because it works for the scenarios described in Table 11–1. ▪ **Server Log**: Sets a watch based on data written to server logs. This type is only useful for extending an existing log watch such as `StuckThread` to include Oracle SOA Suite dumps. ▪ **Event Data**: This option is not applicable because Oracle SOA Suite is not using any WLDF-based instrumentation.
Enable Watch	Select to enable a watch.

   The Configure Watch Rule Expressions page for adding an expression to the watch is displayed.
7. Click **Add Expression**.

   The Add Expression wizard is displayed.
8. In the **MBean Server location** list, select the Oracle WebLogic Server MBean server for the expression you want to configure (for example, **ServerRuntime**).
9. Click **Next**.
10. Select the **Select an MBean Type from the following list** button.
11. In the **MBean Type** list, select the MBean to use for collecting diagnostic information (for this example, Oracle WebLogic Server MBean **weblogic.management.runtime.JRockitRuntimeMBean** is selected).



12. Click **Next**.

    The Select Instances page is displayed.

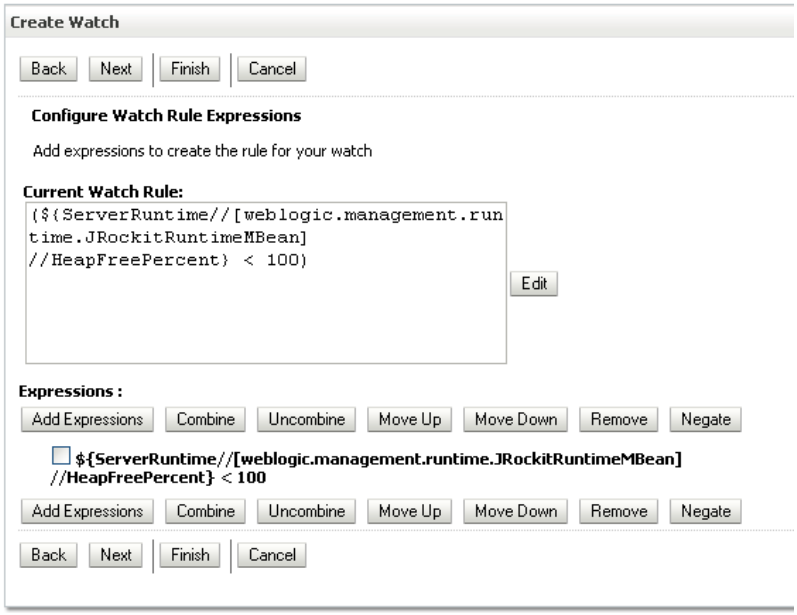
13. From the **Instance** list, select the instance name or specify an instance name pattern to use to identify the metric for the expression.

14. Click **Next**.

15. Enter the following details to create a watch rule expression, and click **Finish**.

Field	Description
Message Attribute	Select a message attribute (for this example, **HeapFreePercent** is selected). The attributes that are displayed for selection are part of the MBean that you selected in Step 11. For example, if you selected: **`oracle.dms.name=/soainfra/engines/bpel/request/system.type=soa_infra_bpel_requests`** You see assigned attributes such as **active_count**, **active_maxValue**, **active_minValue**, **scheduled_count**, and others.
Operator	Select an operator (for this example, **<** is selected).
Value	Enter a value (for this example, **100** is specified).

The Configure Watch Rule Expressions page is displayed with the watch rule expression you created.



16. Click **Next**.

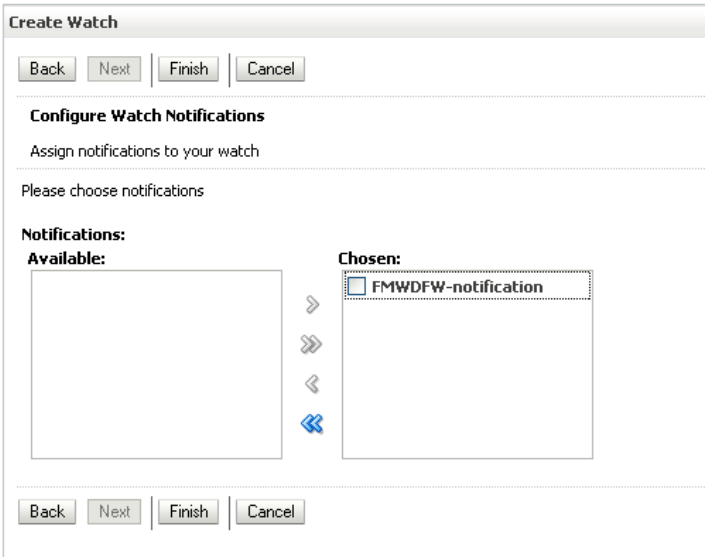
    The Config Watch Alarm page is displayed.

17. Optionally specify an alarm and the alarm's reset value for the watch.

18. Click **Next**.

    The Configure Watch Notifications page is displayed.

19. In the **Available** table, select a notification to assign to the watch and click **>**.

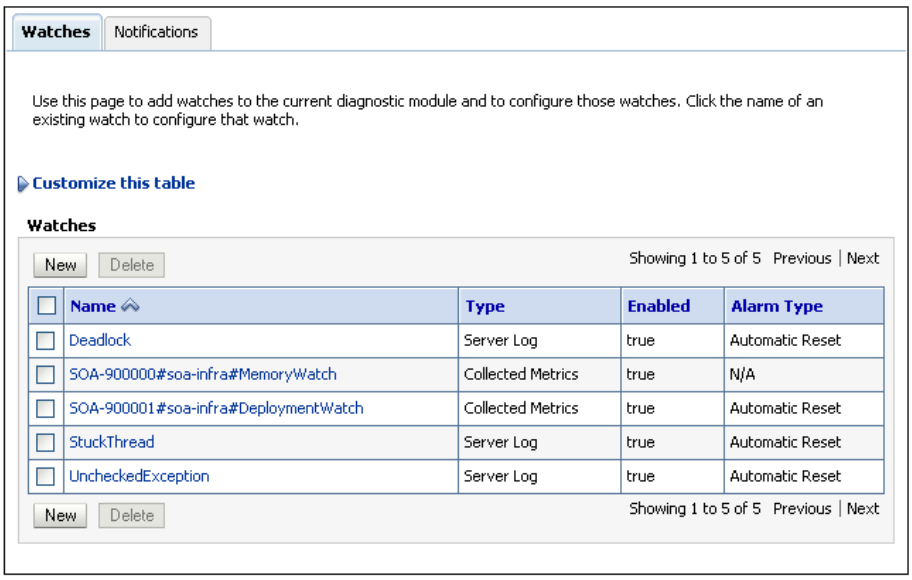


When a watch rule expression evaluates to true, a notification is triggered. This notification is handled by the Diagnostic Framework if the **FMWDFW notification** is selected, which links it to the watch. The **FMWDFW notification** is automatically shipped with Oracle SOA Suite. Oracle recommends that you select this notification because it creates the Oracle SOA Suite dumps described in Section 11.1.4, "Supported Oracle SOA Suite Diagnostic Dumps."

The notification handler creates a problem incident package that contains appropriate SOA dumps in the ADR. The incident package name corresponds to the message ID specified in the watch name. The incident package dumps can be viewed later using standard ADR tools. This feature enables you to take corrective actions for the problem scenario.

20. Click **Finish**.

    The watch you created is displayed at the bottom of the Settings for *Module_Name* page. In addition, three WLDF watches that are automatically shipped with Oracle WebLogic Server (**Deadlock**, **StuckThread**, and **UncheckedException**) are also displayed.



21. In the **Name** column, click the specific watch name to display configuration details about the watch.

11.3 Manually Triggering and Executing Dumps

You can manually execute existing dumps with the WLST command `executeDump` and create incidents when one has not been automatically created.

To manually execute dumps:

1. Specify the `executeDump` command to place dump contents in a file (for this example, `soa.config` is the dump executed).

```
wls:/soainfra/serverConfig> executeDump(name='soa.config', appName='soa-infra')

Start Dumping deployedCompositesCatalog from MDS
URI:deployed-composites/deployed-composites.xml to: /myhome/fmwhome/user_
projects/domains/mydomain/servers/myserver/adr/diag/ofm/mydomain/
myserver/incident/incdir_9/deployedCompositesCatalog

Finished dumping specified MDS metadata to : /myhome/fmwhome/user_
projects/domains/mydomain/servers/myserver/adr/diag/ofm/mydomain/myserver/incid
ent/incdir_9/deployedCompositesCatalog

Start Dumping soaServiceEnginesConfigurations from MDS
URI:soa/configuration/default to: /myhome/fmwhome/user_projects/
domains/mydomain/servers/myserver/AdminServer/adr/diag/ofm/mydomain/
```

```
myserver/incident/incdir_9/soaServiceEnginesConfigurations

Finished dumping specified MDS metadata to : /myhome/fmwhome/user_projects/
domains/mydomain/servers/myserver/AdminServer/adr/diag/ofm/mydomain/
myserver/incident/incdir_9/soaServiceEnginesConfigurations
```

This creates the following output:

2. Specify the `executeDump` command to display dump contents to the screen.

```
executeDump(name='soa.edn')

Type:oracle.integration.platform.blocks.event.saq.SAQBusinessEventBus
Configuration:null
Status: running=true started=true
ThreadCount:3
RetryCount:3
In Global: Tx:false
Cluster Info:oracle.integration.platform.blocks.cluster.CoherenceCluster
Interface1mpt@163bd717
SharedEDN:false
OOAO Queue Name:edn_ooao_queue
Java Subscriber Name:edn java subscriber
Subscription Info:
No namespace subscription...
QName subscriptions:
==============================================================
qname={http://schemas.oracle.com/events/edl/ActionOccur}ADEvent
subscriptions=
id=default/WSInMedPubBpelSubFileOut!1.0*soa_
7a055d6a-8402-49c2-ac56-5f85cbf3d7f/
BpelSub, consistencyLevel=ONE_AND_ONLY_ONE, filter=XPath Filter: starts-with(/
be:business-event/be:content/ns0:ActionOccurrence/ns0:ParentEntityType/@value,
 'A'), runAsRoles=[$publisher]
id=partition_1/WSInMedPubBpelSubFileOut!1.0*soa_
80a169ab1-395a-4b87-9986-9fa2742a8bd3/
BpelSub, consistencyLevel=ONE_AND_ONLY_ONE, filter=XPath Filter: starts-with(/
be:business-event/be:content/ns0:ActionOccurrence/ns0:ParentEntityType/@value,
 'A'), runAsRoles=[$publisher]
EventThreadContextInfo:
EventTargets:
Event:partition_
1/WSInMedPubBpelSubFileOut!1.0*soa0a169ab1-395a-4b87-9986-9fa2742a9bd3/
BpelSub:::oracle.fabric.BPELServiceEngine@163bd6b5
Event:default/WSInMedPubBpelSubFileOut!1.0*soa7a055d6a-8402-49c2-ac56-5f85cbf3
d7f/BpelSub:::oracle.fabric.BPELServiceEngine@163bd6b5
EDN DB Log enabled:false
```

You can also manually create incidents when one has not been automatically created. For example, this is useful when you notice performance issues and want to create an incident to send to Oracle Support Services. The incident can include SOA dumps, according to the SOA message ID mapping.

This can be performed with the following WLST command:

```
createIncident(messageId="SOA-90000", appName="soa-infra")
```

This has the same effect as WLDF watch notification execution, in which the watch has the message ID of `SOA-90000` and application name of `soa-infra`.

For more information about `executeDump`, see Chapter "Diagnostic Framework Custom WLST Commands" of *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

11.4 Viewing Incident Packages with ADR Tools

ADRCI is a command-line utility that enables you to investigate problems and package and upload first-failure diagnostic data to Oracle Support Services. ADRCI also enables you to view the names of dump files in the ADR, and to view the alert log with XML tags stripped, with and without content filtering.

ADRCI is installed in the following directory:

```
MW_HOME/wlserver_10.3/server/adr
```

For more information about ADRCI, see Chapter "ADRCI: ADR Command Interpreter" of *Oracle Database Utilities*.

For information about other tools, see Chapter "Managing Diagnostic Data" of *Oracle Database Administrator's Guide*

11.5 Querying Problems and Incidents

The Diagnostic Framework provides WLST commands that you can use to view information about problems and incidents, including the following:

- Querying problems across Oracle WebLogic Servers
- Querying incidents across Oracle WebLogic Servers
- Viewing dump files associated with an incident on an Oracle WebLogic Server

For more information about these WLST commands, see Section "Understanding the Diagnostic Framework" of *Oracle Fusion Middleware Administrator's Guide* and Chapter "Diagnostic Framework Custom WLST Commands" of *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

Part V

Administering BPEL Process Service Components and Engines

This part describes how to administer BPEL process service components and engines.

This part includes the following chapters:

- Chapter 12, "Configuring BPEL Process Service Components and Engines"
- Chapter 13, "Monitoring BPEL Process Service Components and Engines"
- Chapter 14, "Managing BPEL Process Service Components and Engines"

12

Configuring BPEL Process Service Components and Engines

This chapter describes how to configure BPEL process service components and service engines, including configuring properties such as audit level, audit trail threshold, and dispatcher thread values; automatic recovery for BPEL processes; master node recovery scheduling; and automatic recovery attempts for invoke and callback messages.

This chapter includes the following topics:

- Section 12.1, "Configuring BPEL Process Service Engine Properties"
- Section 12.2, "Configuring Automatic Recovery for Oracle BPEL Process Manager"
- Section 12.3, "Configuring Master Node Recovery Scheduling"
- Section 12.4, "Configuring Automatic Recovery Attempts for Invoke and Callback Messages"
- Section 12.5, "Setting the Audit Level at the BPEL Process Service Component Level"

12.1 Configuring BPEL Process Service Engine Properties

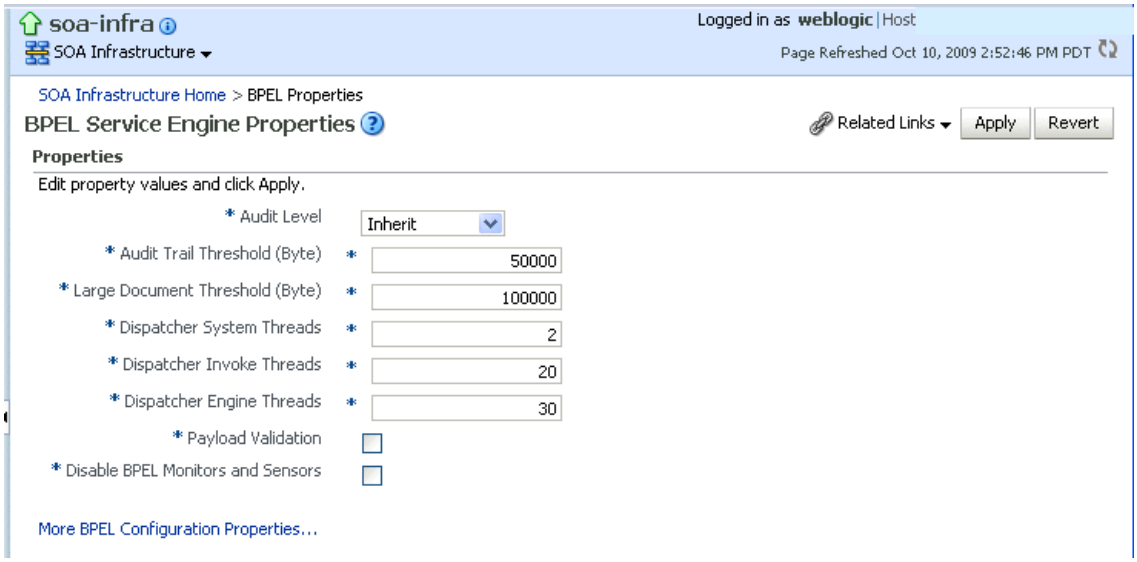
You can configure BPEL process service engine properties, which are used by the BPEL process service engine during processing of BPEL process service components.

To configure BPEL process service engine properties:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **SOA Administration** > **BPEL Properties**.	1. Right-click **soa-infra**. 2. Select **SOA Administration** > **BPEL Properties**.

The BPEL Service Engine Properties page displays properties for setting audit trail and large document thresholds, setting dispatcher thread properties, validating payload schema, and setting the audit trail level.



2. Make changes to the service engine properties that are appropriate to your environment.

Property	Description
Audit Level	Select one of the following options: ▪ **Off**: Composite instance tracking and payload tracking information is not collected. ▪ **Inherit**: Logging equals the SOA Infrastructure audit level. This setting enables the BPEL audit level to automatically change when the global setting is changed. Setting a different audit level tracking in this page overrides the tracking set at the SOA Infrastructure level. ▪ **Minimal**: The BPEL service engine does not capture any audit details. Therefore, they are not available in the flow audit trails. All other events are logged. ▪ **Production**: The BPEL service engine does not capture the payload. The payload details are not available in the flow audit trails. Payload details for other BPEL activities are collected, except for assign activities. This level is optimal for most standard operations and testing. ▪ **Development**: Allows both composite instance tracking and payload tracking. All events are logged. However, it may have an impact on performance. This level is useful mostly for debugging purposes.
Audit Trail Threshold	Enter the maximum size in bytes of an instance audit trail before it is chunked and saved in a dehydration store table separate from the audit trail. If the threshold is exceeded, the **View XML** link is shown in the audit trail instead of the payload.
Large Document Threshold	Enter the maximum size of a generated document within a BPEL process component instance before it is stored in a separate table in the dehydration store.

Property	Description
Dispatcher System Threads	Specify the total number of threads allocated to process system dispatcher messages. System dispatcher messages are general cleanup tasks that are typically processed quickly by the server (for example, releasing stateful message beans back to the pool). Typically, only a small number of threads are required to handle the number of system dispatch messages generated during runtime. The default value is `2` threads. Any value less than `1` thread is changed to the default. For information about monitoring these threads, see Section 13.6, "Monitoring BPEL Process Service Engine Request and Thread Statistics."
Dispatcher Invoke Threads	Specify the total number of threads allocated to process invocation dispatcher messages. Invocation dispatcher messages are generated for each payload received and are meant to instantiate a new instance. If the majority of requests processed by the service engine are instance invocations (as opposed to instance callbacks), greater performance may be achieved by increasing the number of invocation threads. Higher thread counts may cause greater CPU utilization due to higher context switching costs. The default value is `20` threads. Any value less than `1` thread is changed to the default. For information about monitoring these threads, see Section 13.6, "Monitoring BPEL Process Service Engine Request and Thread Statistics."
Dispatcher Engine Threads	Specify the total number of threads allocated to process engine dispatcher messages. Engine dispatcher messages are generated whenever an activity must be processed asynchronously. If most of the processes deployed are durable with a large number of dehydration points (midprocess receive, onMessage, onAlarm, and wait activities), greater performance may be achieved by increasing the number of dispatcher engine threads. Higher thread counts can cause greater CPU utilization due to higher context-switching costs. The default value is `30` threads. Any value less than `1` thread is changed to the default. For information about monitoring these threads, see Section 13.6, "Monitoring BPEL Process Service Engine Request and Thread Statistics."
Payload Validation	Select to enable validation of inbound and outbound messages. Nonschema-compliant payload data is intercepted and displayed as a fault. **Note:** This setting is independent of the SOA composite application and SOA Infrastructure payload validation level settings. If payload validation is enabled at both the service engine and SOA Infrastructure levels, data is checked twice: once when it enters the SOA Infrastructure, and again when it enters the service engine.
Disable BPEL Monitors and Sensors	Select this checkbox to disable all BPEL monitors and sensors defined for all BPEL components across all deployed SOA composite applications.

3. Click **Apply**.
4. If you want to configure advanced BPEL properties in the System MBean Browser, click **More BPEL Configuration Properties**. Properties that display include, but are not limited to, the following. Descriptions are provided for each property.
   - **AsynchAuditBatchSize**: Stores multiple audit trail messages (across instances) in a single transaction on Oracle Exalogic platforms. For more information, see Section 14.5, "Storing Instance and Callback Message Data in Oracle Coherence Distributed Cache on Oracle Exalogic Platforms."

- **BpelcClasspath**: The extra BPEL class path to include when compiling BPEL-generated Java sources.
- **DisableAsserts**: Disables the execution of assertions in BPEL, including the `bpelx:assert` activity.
- **DisableSensors**: Disables all calls to sensors.
- **ExpirationMaxRetry**: The maximum number of times a failed expiration call (wait/onAlarm) is retried before failing.
- **ExpirationRetryDelay**: The delay between expiration retries.
- **InstanceKeyBlockSize**: The size of the block of instance IDs to allocate from the dehydration store during each fetch.
- **MaximumNumberOfInvokeMessagesInCache**: The number of invoke messages stored in the in-memory cache.
- **MaxRecoverAttempt**: The number of automatic recovery attempts to submit in the same recoverable instance. For more information, see Section 12.4, "Configuring Automatic Recovery Attempts for Invoke and Callback Messages" and Section 14.5.4, "Configuring the Storage of Multiple Audit Trail Messages in One Transaction."
- **OneWayDeliveryPolicy**: Changes whether one-way invocation messages are delivered.
- **QualityOfService**: Enables or disables Oracle Coherence cache for the BPEL process service engine. For more information, see Section 14.5.3, "Configuring Oracle Coherence Caching."
- **StatsLastN**: The size of the most recently processed request list.
- **SyncMaxWaitTime**: The maximum time a request and response operation takes before timing out.

5. Make changes appropriate to your environment.

12.2 Configuring Automatic Recovery for Oracle BPEL Process Manager

Oracle SOA Suite provides an automatic recovery feature in Oracle Enterprise Manager Fusion Middleware Control that enables you to configure and recover:

- All activities (for example, wait activities and OnAlarm branches of pick activities) that have an associated expiration date and are scheduled with the SOA Infrastructure to be rescheduled
- All activities that are not complete over a provided threshold time
- All invoke and callback messages that are unresolved

To configure automatic recovery:

1. In the navigator, right-click **soa-infra** and select **SOA Administration** > **BPEL Properties**.
2. Click **More BPEL Configuration Properties**.
3. In the **Name** column, click **RecoveryConfig**.
4. Expand **RecurringScheduleConfig**.

   This section enables you to configure recurring recovery attempts.

5. Set the following properties to values appropriate to your environment, and click **Apply**.

Property	Description
maxMessageRaiseSize	The maximum number of messages to submit for each recurring recovery attempt. Use this property to limit the impact of recovery on the server. This value specifies the maximum number of messages to filter from activity, invoke, and callback queries; that is, 50 messages from each of the activity, invoke, and callback tables. The default value is `50`. A negative value causes all messages selected from the database to be submitted for recovery. A `0` value causes no messages to be selected from the database (effectively disabling recovery).
startWindowTime	The start time for the daily recovery window, specified in a 24-hour notation. Therefore, 2:00 pm is specified as `14:00`. The leading zero does not need to be specified for single digit hour values (`1:00-9:00`). The default value is midnight (`00:00`). Any invalid parsed time value is defaulted to midnight.
stopWindowTime	The stop time for the daily recovery window, specified in a 24-hour notation. Therefore, 2:00 pm is specified as `14:00`. The leading zero does not need to be specified for single digit hour values (`1:00-9:00`). If you do not want daily recovery, set the start and stop window times to be the same value. If the stop window time is earlier than the start window time, both the start and stop window times are changed to their respective default values. The default value is midnight (`04:00`), effectively setting recurring recovery to run until `04:00`. Any invalid parsed time values default to `00:00`.
subsequentTriggerDelay	The number of seconds between recovery attempts during daily recurring startup recovery periods. If the next recovery trigger falls outside of the current recovery period, that trigger is not scheduled until the next recurring recovery period (tomorrow). The default value is `300` (five minutes). A negative value causes the default to be selected.
threshHoldTimeInMinutes	This is the threshold time in minutes to ignore for automatic recovery processing. For automatic invoke and callback recovery, this value is used for picking messages with a received date less than the threshhold time. For automatic activities recovery, this value is used for picking activities with a modification date less than the threshhold time. This property prevents the message contention scenario in which a BPEL process service engine picks up a message for recovery while another thread on the service engine is in the middle of processing the message. This property ensures that the recovery part of the service engine only attempts recovery on messages older than the value for **threshHoldTimeInMinutes**. The default value is `10` minutes. A negative value causes the default to be selected.

6. Expand **StartupScheduleConfig**.

   This section enables you to configure server startup recovery attempts.

7. Set the following properties to values appropriate to your environment, and click **Apply**.

Property	Description
maxMessageRaiseSize	The maximum number of messages to submit for each startup recovery attempt. Use this property to limit the impact of recovery on the server. This value specifies the maximum number of messages to filter from activity, invoke, and callback queries; that is, `50` messages from each of the activity, invoke, and callback tables. The default value is `50`. A negative value causes all messages selected from the database to be submitted for recovery. A zero value causes no messages to be selected from the database (effectively disabling recovery).
startupRecoveryDuration	Specifies the number of seconds that the startup recovery period lasts. After the server starts, it goes into a startup recovery period. During this period, pending activities and undelivered callback and invocation messages are resubmitted for processing. The default value is `600` (ten minutes). A negative or zero value disables startup recovery.
subsequentTriggerDelay	The number of seconds between recovery attempts during the server startup recovery period. If the next recovery trigger falls outside the server startup period, that trigger is not scheduled and the server moves into the recurring recovery period. The default value is `300` (five minutes). A negative value causes the default to be selected.

Note: In a cluster, it is possible for different nodes to concurrently attempt an automatic recovery of the same items. The first node to lock the item attempts the recovery, while other nodes may raise an exception that can be safely ignored.

12.3 Configuring Master Node Recovery Scheduling

You can configure a clustered environment to use master node recovery scheduling. In this environment, the master node is dedicated to performing recovery for all nodes in the cluster.

Note: This feature does not work if you are using a pre-Oracle Fusion Middleware Release 1 (11.1.1.3) database schema.

Master node recovery scheduling enables you to perform the following tasks:

- Recover activities with expiration dates (for example, a wait activity or OnAlarm branch of a pick activity) that are past due. The master node picks expired work items and reschedules them.
- Recover stranded work items
- Recover callback messages
- Recover invoke messages
- Fail over expired activities: When the master node detects a failed node, it tries to reschedule work items that have an expiration date.

To configure master node recovery scheduling:

1. Log in to Oracle Enterprise Manager Fusion Middleware Control.
2. Right-click **soa-infra**.
3. Select **SOA Administration** > **BPEL Properties**.
4. Click **More BPEL Configuration Properties**.
5. In the **Name** column, click **RecoveryConfig**.
6. Expand **ClusterConfig**. The ClusterConfig properties work in association with the recurring recovery attempt properties and server startup recovery attempt properties that you set for **RecurringScheduleConfig** and **StartupScheduleConfig**, respectively.
7. Set the following properties to values appropriate to your environment, and click **Apply**.

> **Note:** Once an instance/message becomes recoverable, a recovery is attempted. However, the number of retries is not tracked. If a recovery fails, it continues to pick the same record, retry, and fail again.

Property	Description
clusterDbTimeRefresh	Specifies how often to refresh the local copy of the database time. This takes into account the clock drift on different computers. All nodes in the cluster rely on the database time, regardless of its accuracy. The default value is `12` hours (specified as `43200` seconds).
heartBeatInterval	Specifies how often a node polls the cluster message table to check for messages published by other nodes in the cluster. The default value is `5` seconds. The following tasks are performed each interval: ▪ Updates the node's last updated time in the `cluster_node` table. ▪ Attempts to claim ownership of the master role. ▪ If the master role is claimed, the recovery manager resumes work. ▪ Checks for all nodes that have update times not updated for the **nodeReapThreshold** value, deletes those nodes from the `cluster_node` table, and reschedules all expiring work items from this node.
masteAliveThreshold	Specifies the number of seconds a master node is considered to be active. Master nodes that have not checked in with the cluster for this number of seconds are considered to be terminated. Whichever node gets an exclusive lock on the `cluster_master` table after this point can claim the master role. The default value is `15` minutes (specified as `900` seconds).
nodeReapInterval	Specifies how often the heartbeat thread is borrowed to mark old cluster nodes. Only the master node performs this job. The default value is `2` hours (specified as `7200` seconds).

Property	Description
nodeReapThreshold	Specifies the number of seconds a node is considered to be active. Nodes that have not checked in with the cluster for this number of seconds are considered to be terminated. During its heartbeat cycle, the master node tries to clean up the `cluster_node` table. The default value is `15` minutes (specified as `900` seconds).

12.4 Configuring Automatic Recovery Attempts for Invoke and Callback Messages

You can configure the number of automatic recovery attempts to submit in the same recoverable instance. The value you provide specifies the maximum number of times invoke and callback messages are recovered. If the value is `0` (the default value), it recovers all messages. Once the number of recovery attempts on a message exceeds the specified value, a message is marked as nonrecoverable.

To configure automatic recovery attempts for invoke and callback messages:

1. In the navigator, right-click **soa-infra** and select **SOA Administration** > **BPEL Properties**.
2. Click **More BPEL Configuration Properties**.
3. Go to **MaxRecoverAttempt**.
4. In the **Value** field, enter a value.
5. Click **Apply**.

For information about recovering invoke and callback messages, see Section 14.4, "Performing BPEL Process Service Engine Message Recovery."

12.5 Setting the Audit Level at the BPEL Process Service Component Level

You can set the audit level for a BPEL process service component. This setting takes precedence over audit level settings at the SOA Infrastructure, service engine, and SOA composite application levels. The service component level setting is only available for BPEL processes and is *not* supported for the Oracle Mediator, human workflow, and business rule service components.

There are two ways to set the audit level for BPEL process service components. Supported values are **Off**, **Minimal**, **Inherit**, **Development**, and **Production**.

To set the audit level for BPEL process service components:

- In the System MBean Browser of Oracle Enterprise Manager Fusion Middleware Control:
  1. In the navigation tree, expand the **SOA** folder.
  2. Right-click **soa-infra**, and select **Administration** > **System MBean Browser**.
  3. Select **Application Defined MBeans** > **oracle.soa.config** > **Server:** *server_name* > **SCAComposite** > *Composite_Name* > **SCAComposite.SCAComponent** > *BPEL_Service_Component* > **Properties**.
  4. Click the **Add** icon.
  5. Expand the **Element**_*number* folder.

6. From the **many** list, select **false**.
7. In the **name** field, enter `bpel.config.auditlevel`.
8. In the **value** field, enter a value.
9. Click **Apply**.

- In Oracle JDeveloper:

  1. Set the `bpel.config.auditLevel` property to an appropriate value in the `composite.xml` file of your SOA project.

```
<component name="BPELProcess">
   <implementation.bpel src="BPELProcess.bpel" />
   <property name="bpel.config.auditLevel">Off</property>
</component>
```

For more information about audit levels, see Section 1.4.1.1, "Introduction to the Order of Precedence for Audit Level Settings."

13

Monitoring BPEL Process Service Components and Engines

This chapter describes how to monitor BPEL process service components and service engines, including monitoring the audit trail and process flow, service component and service engine instances and faults; sensor data; service engine requests and thread statistics; and deployed BPEL process service components.

This chapter includes the following topics:

- Section 13.1, "Monitoring the Audit Trail and Process Flow of a BPEL Process Service Component"
- Section 13.2, "Monitoring BPEL Process Service Component Recent Instances and Faults"
- Section 13.3, "Monitoring BPEL Process Service Component Instances"
- Section 13.4, "Monitoring Sensor Data and Values in BPEL Process Service Components"
- Section 13.5, "Monitoring BPEL Process Service Engine Instances and Faults"
- Section 13.6, "Monitoring BPEL Process Service Engine Request and Thread Statistics"
- Section 13.7, "Monitoring BPEL Process Service Engine Instances"
- Section 13.8, "Monitoring Deployed BPEL Process Service Components in the Service Engine"

For more information, see the following sections:

- Section 1.2.4, "Introduction to Service Components and Service Component Instances"
- Section 1.2.6, "Introduction to Service Engines"

13.1 Monitoring the Audit Trail and Process Flow of a BPEL Process Service Component

This section describes how to view the audit trail and process flow of a BPEL process service component in a SOA composite application instance.

> **Note:** This section assumes a SOA composite application instance has been initiated. If not, see Section 8.1, "Initiating a SOA Composite Application Test Instance" for instructions.

To view the audit trail and process flow of a BPEL process service component:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**.	1. Under **soa-infra**, expand the partition.
2. Select the **Deployed Composites** tab.	2. Select a specific SOA composite application.
3. In the **Composite** section, select a specific SOA composite application.	

   The Dashboard page for the selected composite application appears.

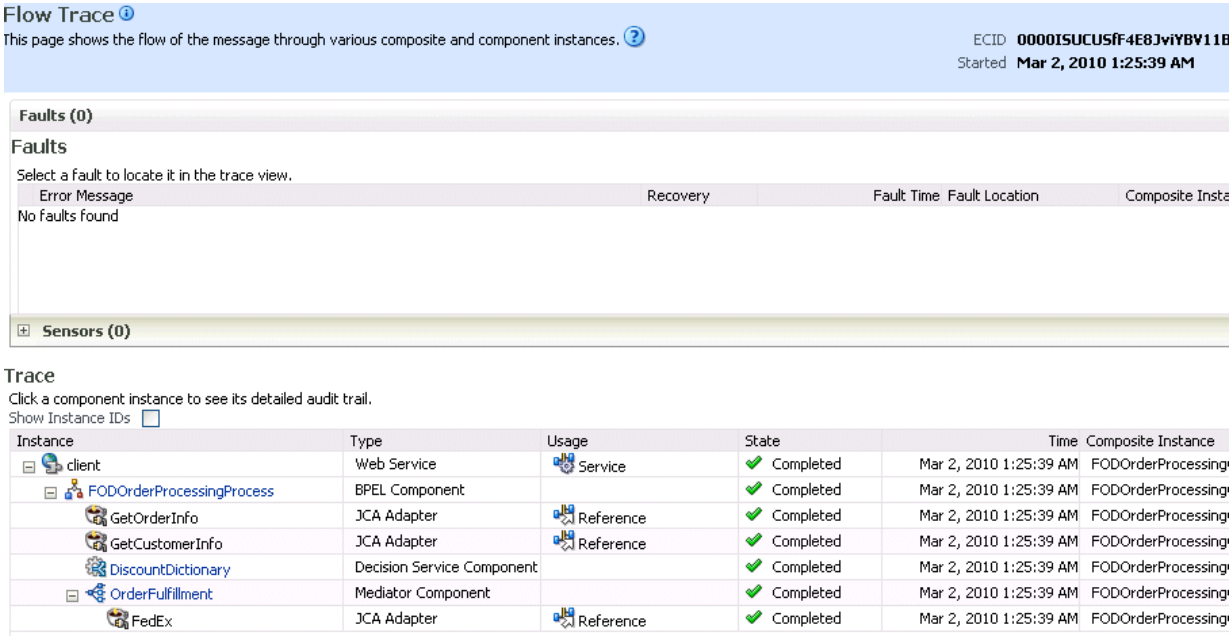
2. Use one of the following methods to select an instance of the application:
   - For recent instances of this application, click the instance number of an instance in the **Instance ID** column of the **Recent Instances** section.
   - For all instances of this application, click the **Instances** tab, then click a specific instance in the **Instance ID** list.

   The Flow Trace page displays the following details:
   - The **Faults** section shows the faults occurring in the services, service components, and references that comprise the SOA composite application. Sensors enable you to monitor BPEL process activities, variables, and faults during runtime. Selecting a fault highlights the row in the **Trace** section in which the fault occurred. Closing the fault clears the selection in the **Trace** section.
   - The **Sensors** section displays details about composite sensors included in the service and reference binding components of the SOA composite application. The total number of sensors is shown in the section header. Composite sensors can be added to service and reference binding components and service components that have subscribed to business events during design time in Oracle JDeveloper. Selecting a composite sensor in this section highlights the service or reference in the **Trace** section in which composite sensor data was collected. Closing the sensor clears the selection in the **Trace** section.

     Note: Expand the **Faults** or **Sensors** sections one at a time. The fault or sensor information is only displayed for viewing in this way.

   - The **Trace** section shows the sequence of the message flow through the services, service components, and references that comprise the SOA composite application.

   The flow trace is a runtime trail of a message flow identified by an execution context ID (ECID) that is displayed in the upper right-hand corner of the page. An ECID enables you to track a message flow that crosses instances of different composite applications. The flow trace lists all services, references, and components across composite applications participating in the flow.



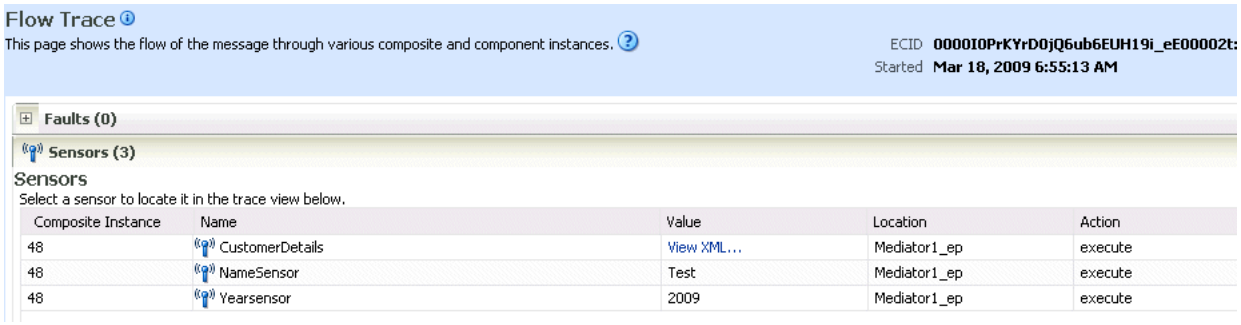
For the flow example in the **Trace** section, the service binding component, service components, and reference binding component involved in the flow have successfully received and processed messages.

3. Select a fault in the **Faults** section.

   This highlights the row in the **Trace** section in which the fault occurred.

4. Close the fault to clear the selection in the **Trace** section.

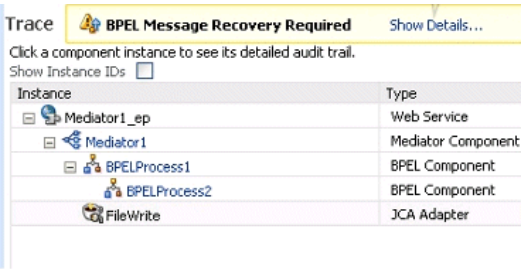
5. Expand the **Sensors** section to display composite sensors.



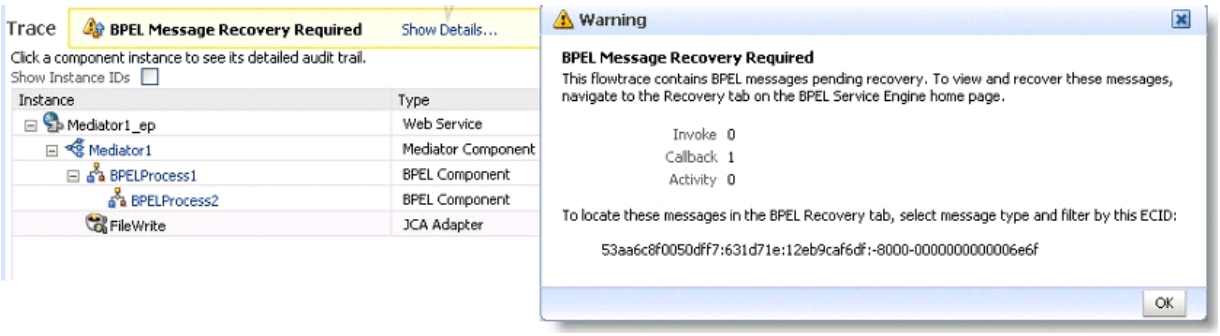
6. Select a sensor in the **Sensors** section.

   This highlights the row in the **Trace** section in which the composite sensor data was collected.

   If there are BPEL process messages that require recovery from the Recovery page of the BPEL process service engine, a `BPEL Message Recovery Required` inline warning message and recovery icon are displayed.



7. Click **Show Details** or the recovery icon to display a Warning dialog with the following recovery details:
   - The number of invoke, callback, and activity recoverable message types
   - The ECID value



Use this information for creating search criteria for filtering the recoverable messages on the Recovery page of the BPEL process service engine. You can copy the ECID number from the Warning dialog, paste it into the **ECID** field, and select the recoverable message type from the **Type** list. For more information about creating search criteria with the ECID number in the Recovery page, see Section 14.4, "Performing BPEL Process Service Engine Message Recovery."

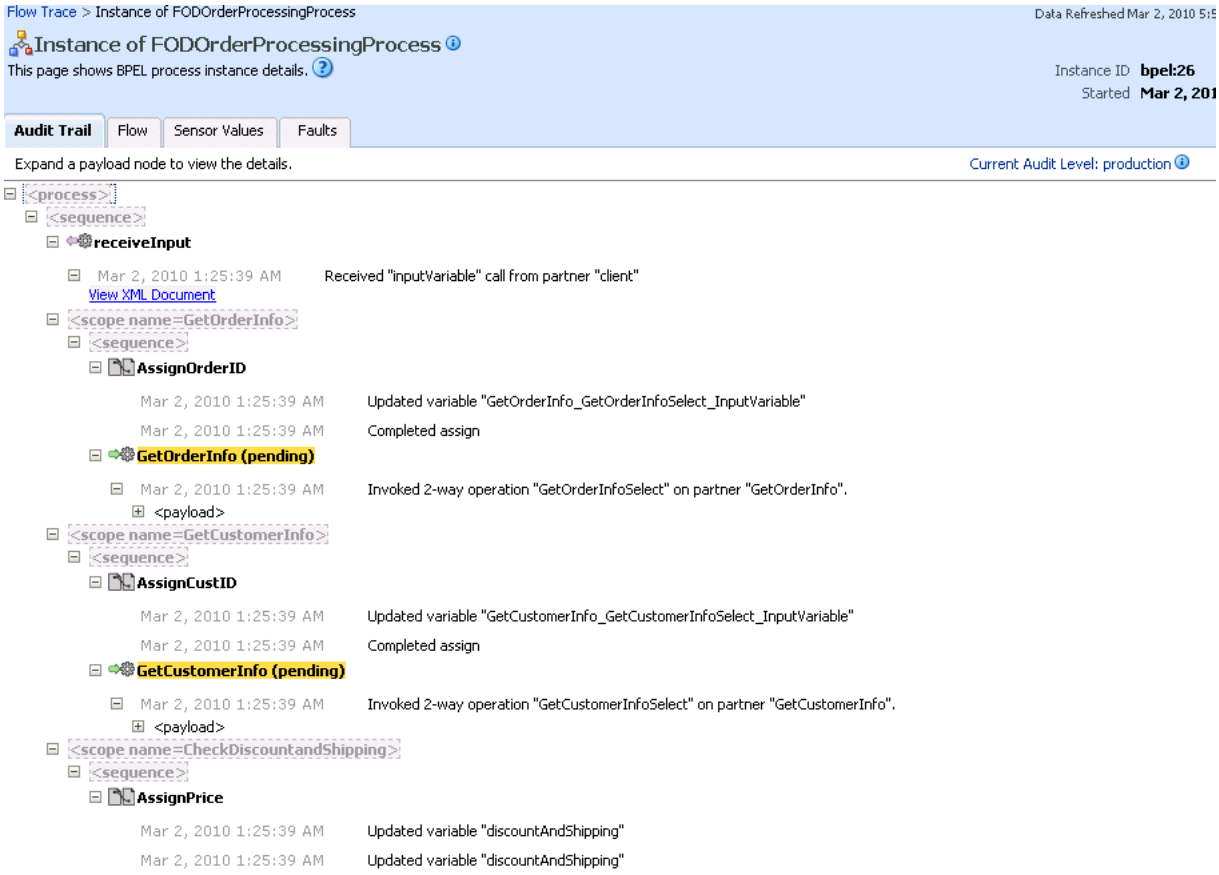
The display of this message recovery information on the Flow Trace page is controlled by the **AuditConfig** property in the System MBean Browser. By default, this property is set to **All**, which enables this information to be displayed. To prevent this information from displaying on the Flow Trace page, set the **bpelRecoveryStatus** key to **Off** for the **AuditConfig** property in the **More SOA Infra Advanced Configuration Properties** section of the SOA Infrastructure Common Properties page. For more information, see Section 3.1, "Configuring SOA Infrastructure Properties."

Note the following restrictions with ECIDs:

- A separate ECID is displayed for each instance of a composite application and not for the composite level ECID that can track the complete flow of any instances for the composite application.
- To get complete flow information, you must find the composite level ECID in the log files. Use that value to get all information for a particular composite and therefore all its executed instances.
- ECIDs are not propagated through business events. This can limit the amount of logging information that is collected. For example, if you publish an event that is subscribed to in the same composite application, limited logging information is available.

8. In the **Instance** column of the **Trace** section, click a specific BPEL process service component instance. Service component instances can be accessed from this section; services and references cannot be accessed.

   The Instance page appears.



   Use these four pages to view the audit trail, flow, sensor values, and faults of a BPEL process service component instance. The following links provide additional details about the instance:

   - **Flow Trace link**: Click the breadcrumbs in the upper left-hand corner of the page to access the flow trace for the ECID (composite instance) that contains this BPEL component instance.
   - **Information icon**: Click the information icon to the right of the name of the BPEL component (in the page title) to see biographical information about this BPEL instance. This information includes a summary of the instance, including instance ID, ECID, instance startup time or last modification time, instance state (for example, running), and number of faults.

     This icon is only displayed on the Audit Trail pages of BPEL processes and Oracle Mediators, and not on the pages of human tasks and business rules.
   - **Audit Level Settings**: Click to display information details, such as the audit level used by this instance.
   - **View Raw XML**: Click to display the raw XML of the audit trail.

   The Audit Trail page displays execution details about the activities in the BPEL process.

9. Scroll through the audit trail to check for errors and expand the payload links to view their contents at a given point in the flow.

Notes:

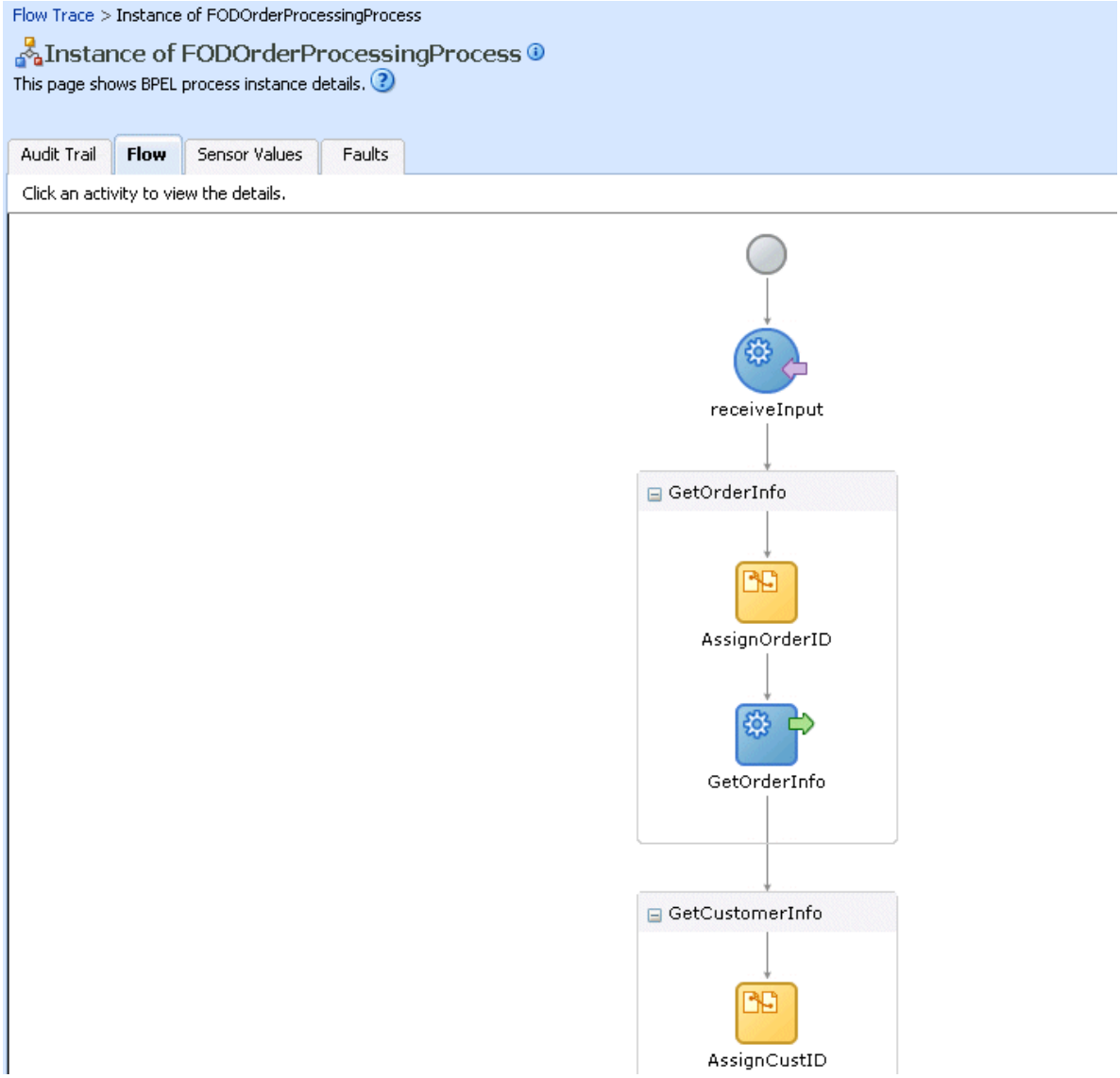
- Canceled onMessage branches of pick or scope activities that did not execute are displayed in the audit trail. However, the flow diagram does not show these same canceled onMessage branches. This is the expected behavior.
- The following error message appears when a transaction is displayed as rolled back in the Audit Trail page:

  ```
  The transaction was rolled back. The work performed for bpel
  instance "instance_ number" was rolled back to the previous
  dehydration point,  but the audit trail has been saved.
  You can recover the instance from the recovery console by
  resubmitting the callback message or activity for execution
  ```

  This message does not specifically state whether recovery should happen on either the activity or the callback. This is the intended behavior. Oracle recommends that you do not recover each instance through the audit messages. Instead, set up automatic recovery to recover these instances.

10. Click the **Flow** tab.

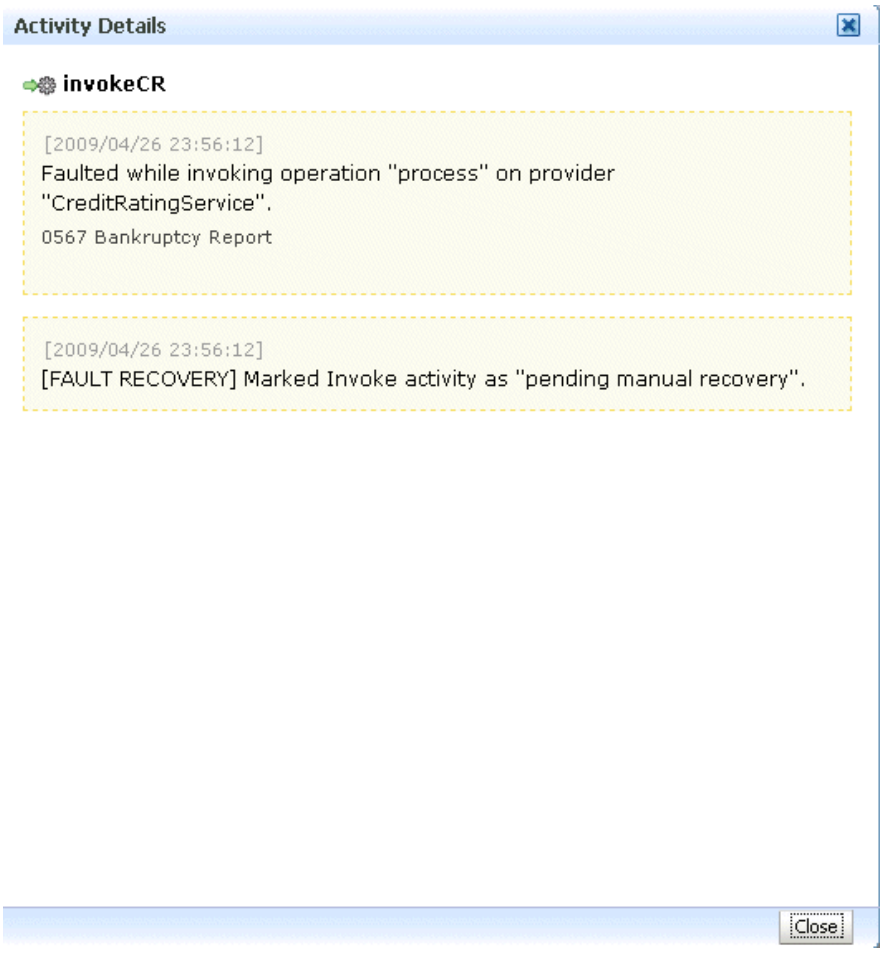
    A flow diagram of the BPEL process activities appears. This flow diagram shows a fault highlighted in a BPEL process activity.



11. Click an activity to view the flow of the payload through the process.

> **Note:** If using Microsoft Internet Explorer, you can click **Copy details to clipboard** to copy the activity details to the clipboard. If using Mozilla Firefox, this link does not appear. Instead, you must manually select the text, and copy and paste it to a file.

12. Scroll through the flow diagram to check for errors and click the highlighted activity to view error messages.



13. Close the message.
14. Click the **Faults** tab.

    This page shows the error message, whether you can recover from the fault, the time at which the fault occurred, and the activity in which the fault occurred. This page displays the faults in the BPEL component instance (but not the faults that occurred in a service or reference binding component).

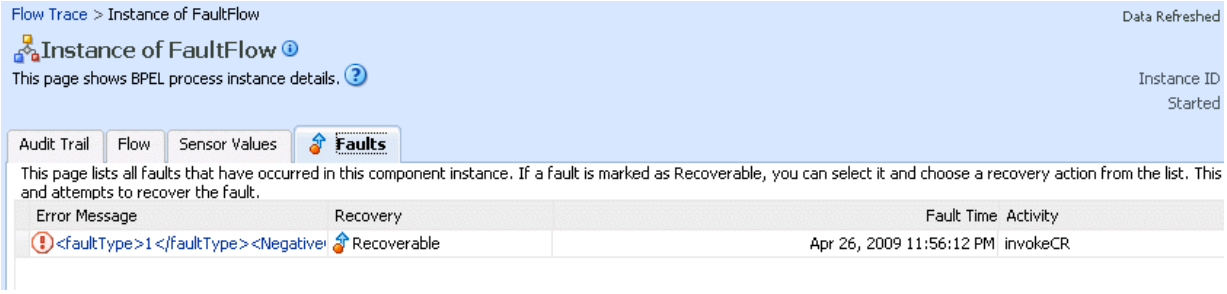
    If a fault occurs when processing activities, the activity location of the fault is not usually shown in the **Activity** column.

    - For Oracle BPEL Process Manager, this column only shows a receive activity that has timed out. In all other cases, this column is empty.
    - For Oracle BPM, this column is always empty.

    This is the expected behavior.

    You can recover from instance faults identified as recoverable. This page lists all instance faults, recoverable or not. The component instance faults that occurred in a service or reference are not listed here.

    This page enables you to target individual faults from which to recover, and provides a degree of fault recovery granularity not available on other pages.



However, you cannot perform bulk fault recoveries on this page. To perform bulk fault recovery, use one of the following pages:

- Faults and Rejected Messages page of a specific SOA composite application or of the SOA Infrastructure
- Faults page of the BPEL process service engine or of a specific BPEL process service component

15. Select a fault for recovery that has been identified as recoverable through one of the following methods. The page refreshes to display a fault recovery section at the bottom of the page.
    - If you click a fault in the **Error Message** column, a popup message displays details about the fault, including the fault ID, fault time, fault location, fault type, and complete error message text. If the fault is identified as recoverable, a **Recover Now** button that you can click is displayed.
    - You click a fault identified as recoverable in the **Recovery** column.
16. Select an action from the **Recovery Action** list.

Action	Description
Retry	Retries the instance with an option to provide a retry success action. An example of a scenario in which to use this recovery action is when the fault occurred because the service provider was not reachable due to a network error. The network error is now resolved.
Abort	Terminates the entire instance.
Replay	Replays the entire scope activity again in which the fault occurred.
Rethrow	Rethrows the current fault. BPEL fault handlers (catch branches) are used to handle the fault. By default, all exceptions are caught by the fault management framework unless an explicit rethrow fault policy is provided.
Continue	Ignores the fault and continues processing (marks the faulted activity as a success).

Your selection causes additional fields to appear. For example, the following fields are displayed if you select **Rethrow**:

17. Use the **After Successful Retry** list to select defined actions to invoke after a successful retry. If you select a variable in the **Variable** list, you can edit the value in the **Value** text box.
18. Click the **Back** button of your browser to exit the flow diagram.

13.1.1 Flow Traces for Composite-to-Composite Invocations on Multiple Servers

When you have multiple SOA composite applications on multiple servers invoking one another, the BPEL process flow trace cannot be rendered to show the correct parent-child relationship. For example, assume you have the following scenario:

- Composite 1 invokes composite 2, which invokes composite 3.
- Each composite includes a BPEL process service component.
- Composite 1 and composite 3 are on server 1, while composite 2 is on server 2.

In this environment, the parent ID of component 3 is set to component 2. However, because composite 2 is on a separate server, the flow trace does not show composite 3 as having a parent. Therefore, composite 3 is placed at the root of the flow trace.

13.1.2 Monitoring BPEL 2.0 Activities in the Audit Trail and Process Flow

You can view the audit trail and process flow for BPEL service component instances that support version 2.0 of the BPEL specification. The following BPEL 2.0 activities can be viewed:

- Process structural flow-related activities: These activities are similar to other workflow or programming languages for sequencing, iteration, and branching:
  - forEach: This activity is the same as the Oracle extension flowN activity in BPEL 1.1. This activity supports processing multiple sets of activities in both parallel and serial fashion.
  - if: This activity is the same as the switch activity in BPEL 1.1.
  - repeatUntil: This activity provides for repeated execution of a contained activity until the given Boolean condition evaluates to true.
- Exception handling and recovery activities:
  - compensateScope: This activity starts compensation on a specified inner scope that has completed successfully.
  - rethrow: This activity rethrows the fault that was originally caught by the immediately enclosing fault handler.

- Other activities
  - exit: This activity is the same as the terminate activity in BPEL 1.1.
  - dehydrate: This activity enables you to dehydrate an instance. This activity is also available in BPEL 1.1.

For example, Figure 13–1 shows an **if** activity in the audit trail of a BPEL process.

Figure 13–1 if Activity in the Audit Trail of a BPEL Process

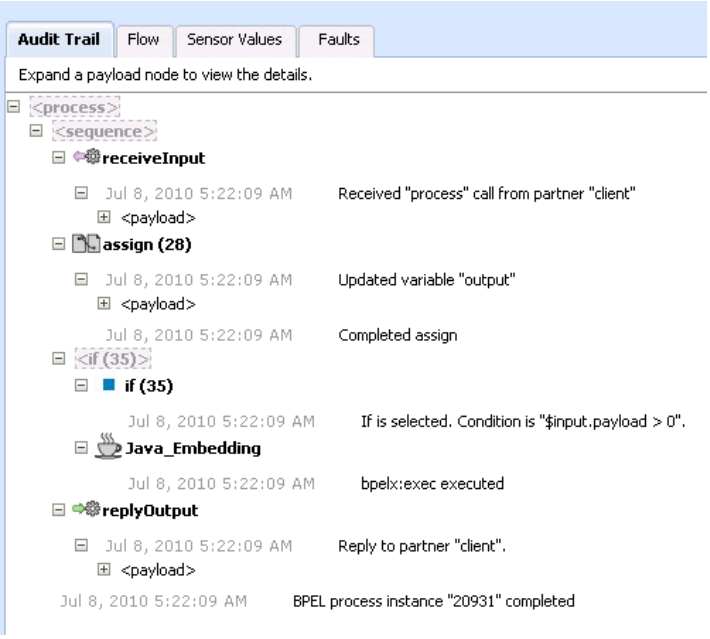
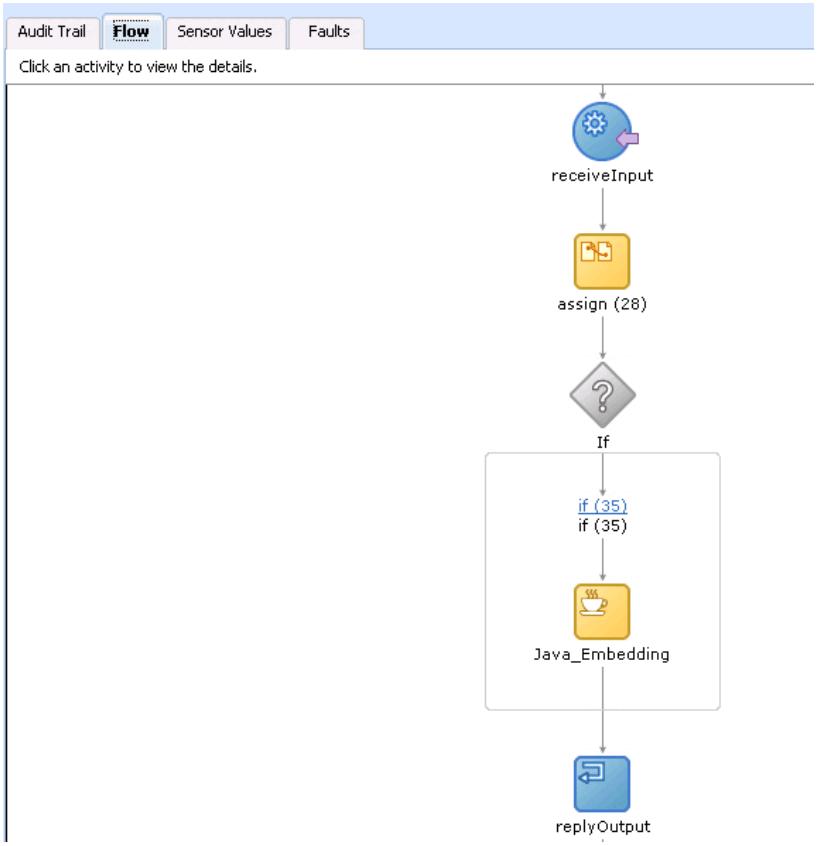


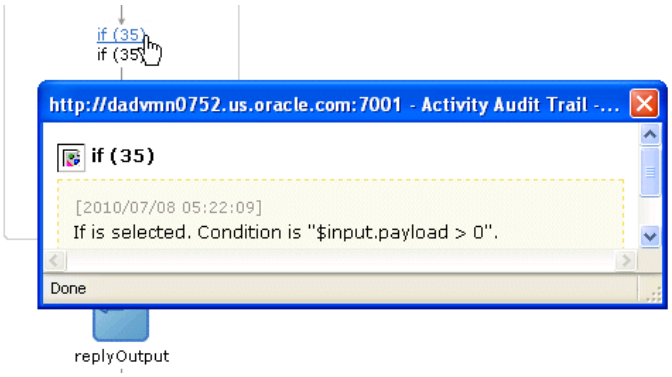
Figure 13–2 shows the same **if** activity in the flow of the BPEL process.

Figure 13–2 if Activity in the Flow of a BPEL Process



Clicking the **if** activity displays the condition defined for this activity.

Figure 13–3 if Activity Condition



For more information about BPEL 2.0, see the *Web Services Business Process Execution Language Version 2.0* specification located at the following URL:

`http://www.oasis-open.org`

For more information about designing BPEL process service components that support version 2.0 of the BPEL specification, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

13.2 Monitoring BPEL Process Service Component Recent Instances and Faults

You can monitor recent instances and faults for BPEL process service components. Each service component in a SOA composite application has its own instance ID. These IDs are different from the overall instance ID of the SOA composite application of which each service component is a part.

To monitor BPEL process service component instances and faults:

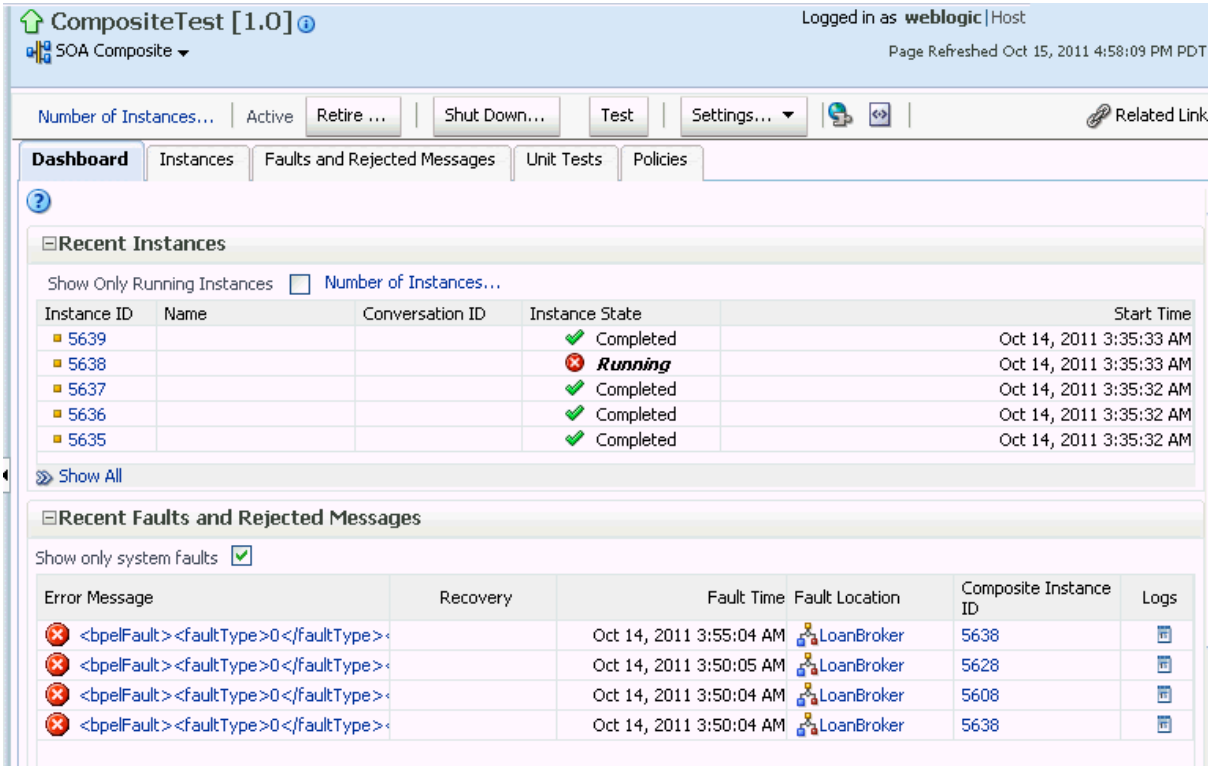
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**.	1. Under **soa-infra**, expand the partition.
2. Select the **Deployed Composites** tab.	2. Select a specific SOA composite application.
3. In the **Composite** section, select a specific SOA composite application.	

2. In the **Component Metrics** section, select the BPEL process service component.
3. Click **Dashboard**.

   The upper part of the Dashboard page displays the following details:

   - Recent instances of the BPEL process service component, including the instance ID, the state of the instance (for example, completed successfully or faulted), the start time, the last modification time, and the logs describing the instance.
   - Recent faults in the BPEL process service component, including the error message, whether you can recover from the fault, the time at which the fault occurred, the instance ID of the BPEL service component, the BPEL activity in which the fault occurred, and the logs describing the fault.
   - The average processing time for each activity in the BPEL process service component.



4. In the **Recent Instances** section, perform the following tasks:
   a. In the **Instance ID** column, click an instance ID for a service component to view its audit trail, process flow, sensor values, and faults.
   b. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.
   c. Click **Show All** below the section to access the Instances page of the service component.
5. In the **Recent Faults** section, perform the following tasks:
   a. In the **Error Message** column, click an error message to display complete information about the fault. If the fault is identified as recoverable, click the **Recover Now** link to perform fault recovery.
   b. In the **Recovery** column, click a fault identified as **Recoverable** to perform fault recovery at the component instance level.
   c. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.
   d. Click **Show All** below the section to access the Faults page of the service component.

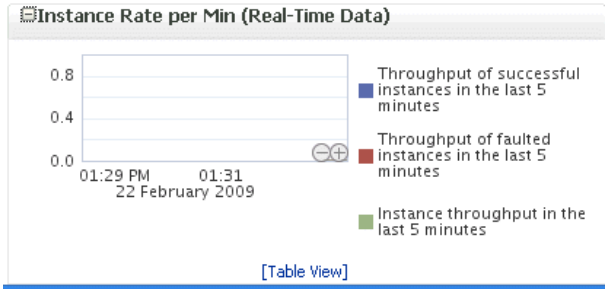
   The lower part of the Dashboard page displays the following details:

   - Details about the time distribution for activities, including the activity name, the total number of activities for all instances, and the average execution time.

Activity Time Distribution

Name	Count	Average Execution Time (ms)
assign (54)	2,201	0.001
receiveInput	2,201	0.001
replyOutput	2,201	0.006

- A graphical representation of the number of successful, faulted, and incoming (pending) instances of the BPEL process service component over a specific time range. Click **Table View** to display throughput details for the last five minutes, including the throughput for successful instances, the total faults throughput, and the instance throughput.



For more information, see Section 1.2.3, "Introduction to SOA Composite Application Instances" and *Oracle Fusion Middleware Administrator's Guide* for details about viewing and searching log files.

13.3 Monitoring BPEL Process Service Component Instances

You can monitor BPEL process service component instances. Each service component has its own unique instance ID. This ID is in addition to the instance ID of the overall SOA composite application of which this service component is a part.

To monitor BPEL process service component instances:

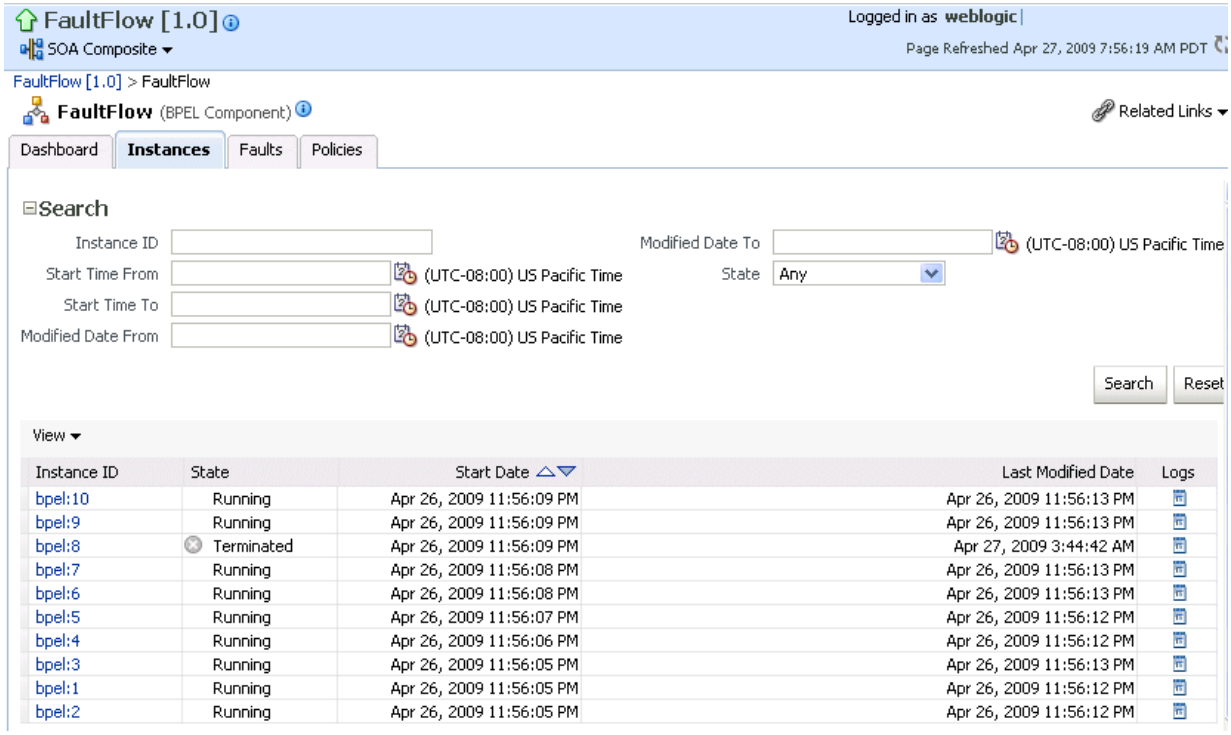
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**.	1. Under **soa-infra**, expand the partition.
2. Select the **Deployed Composites** tab.	2. Select a specific SOA composite application.
3. In the **Composite** section, select a specific SOA composite application.	

2. Select the BPEL process service component in the **Component Metrics** section.
3. Click **Instances**.

   The Instances page displays the following details:

   - A utility for searching for a specific BPEL service component instance by specifying criteria and clicking **Search**.
   - BPEL process service component instances, including the instance ID, instance state (for example, completed or faulted), instance start time, last instance modification time, and log files describing the instance.



4. In the **Instance ID** column, click an instance ID for a service component to view its audit trail, process flow, sensor values, and faults.
5. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

For more information, see Section 1.2.3, "Introduction to SOA Composite Application Instances."

13.4 Monitoring Sensor Data and Values in BPEL Process Service Components

You can view the fault, activity, and variable sensor data of a BPEL process service component. You design sensors in BPEL processes and trackable fields in Oracle JDeveloper. Sensors enable you to monitor BPEL process activities, variables, and faults during runtime.

To monitor sensor data and values in BPEL process service components:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**.	1. Under **soa-infra**, expand the partition.
2. Select the **Deployed Composites** tab.	2. Select a specific SOA composite application.
3. In the **Composite** section, select a specific SOA composite application.	

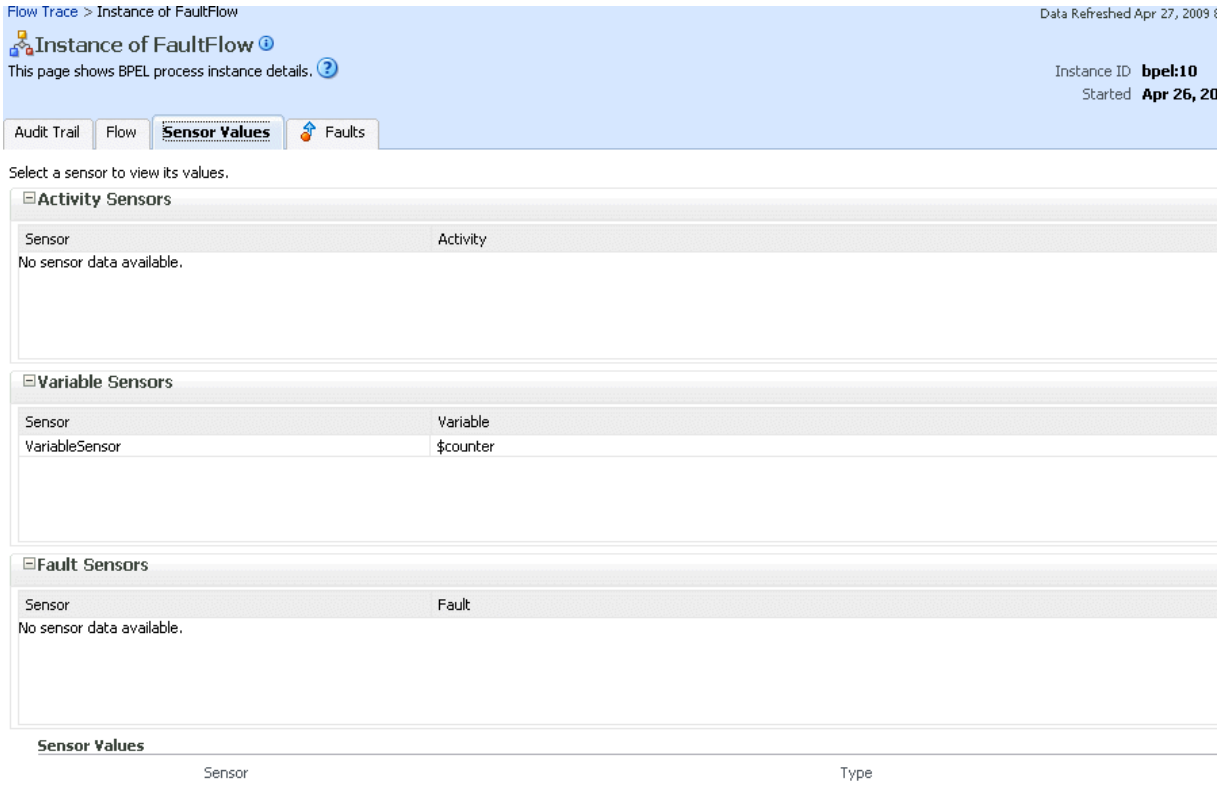
2. Use one of the following methods to select an instance of the application:
   - For recent instances of this application, click the instance number of an instance in the **Instance ID** column of the **Recent Instances** section.

- For all instances of this application, click the **Instances** tab, then click a specific instance in the **Instance ID** column.

The Flow Trace page appears.

3. Click a specific BPEL process service component in the **Instance** column of the **Trace** section.
4. Click the **Sensor Values** tab.
5. Select a sensor to view details.

   If you created JMS sensors in your BPEL process, JMS sensor values are not displayed in Oracle Enterprise Manager Fusion Middleware Control. Only sensor values in which the sensor action is to store the values in the database appear (for example, database sensor values).



For more information about sensors, see Chapter "Using Oracle BPEL Process Manager Sensors" of *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

13.4.1 Behavior of Activity Sensors in Compensate and CompensateScope Activities in BPEL 2.0

Assume you have an activity sensor in a BPEL 2.0 compensateScope activity with the evaluation time set to completion and have also configured the activity's variable sensor to point to a variable. You may expect that when the sensor is triggered, the compensation activity completes and the variable shows the state that it is in at the completion of the compensation activities.

However, note that the compensateScope activity (and also a compensate activity) ends immediately after passing control to the compensation activities. As a result, the

activity's variable sensor shows the state that it is in at the beginning of the compensation activities, rather than at the end.

This is the expected behavior.

As a workaround for obtaining the variable state upon completion of compensation, add an empty activity after the compensate or compensateScope activity and set the activity sensor on the empty activity with an evaluation time set to activation.

13.5 Monitoring BPEL Process Service Engine Instances and Faults

You can monitor instances and faults of all BPEL process service components running in the BPEL process service engine. These BPEL process service components can be part of separate SOA composite applications.

To monitor BPEL process service engine instances and faults:

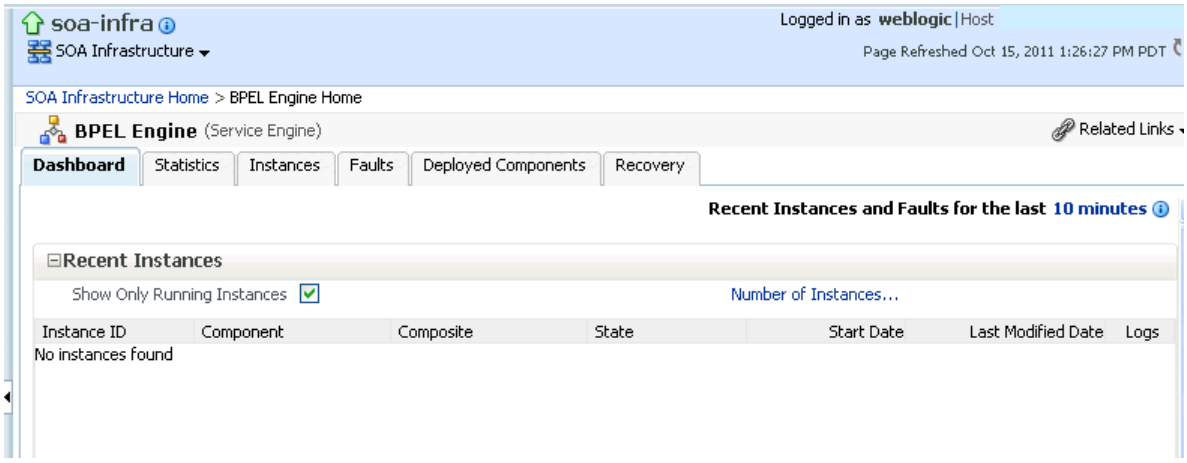
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **BPEL**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **BPEL**.

2. Click **Dashboard**.

   The upper part of the Dashboard page displays the following details:

   - A message indicating that the retrieval of recent instances and faults that are displayed on this page is restricted to the specified time period. This message is displayed if the **Restrict display of instances and faults to the last *time_period*** checkbox is selected on the SOA Infrastructure Common Properties page (it is selected by default). The default time period value is 24 hours, but you can change this value. If this checkbox is not selected, all instances and faults (including count metrics) for this service engine since the last purging are displayed.
   - Recent instances of all BPEL process service components running in the BPEL process service engine, including the instance ID of the service component, the service component name, the SOA composite application of which the service component is a part, the state of the instance (for example, completed successfully or faulted), the instance start time, the last modification time, and the logs describing the instance.



3. In the **Recent Instances** section, perform the following monitoring tasks:

   a. Click the **Number of Instances** link to display a message showing the numbers of running and total instances for this service engine.

      If you selected the **Restrict display of instances and faults to the last *time_period*** checkbox on the SOA Infrastructure Common Properties page and specified a time period or accepted the default value, the number of running and total instances for this service engine for that time period are displayed. If you did not select this checkbox, all instances and faults for this service engine since the last purging are displayed. Click **Recalculate** to recalculate the numbers.

   b. In the **Instance ID** column, click an instance ID for a service component to view its audit trail, process flow, sensor values, and faults.

   c. In the **Component** column, click a specific service component to access its home page.

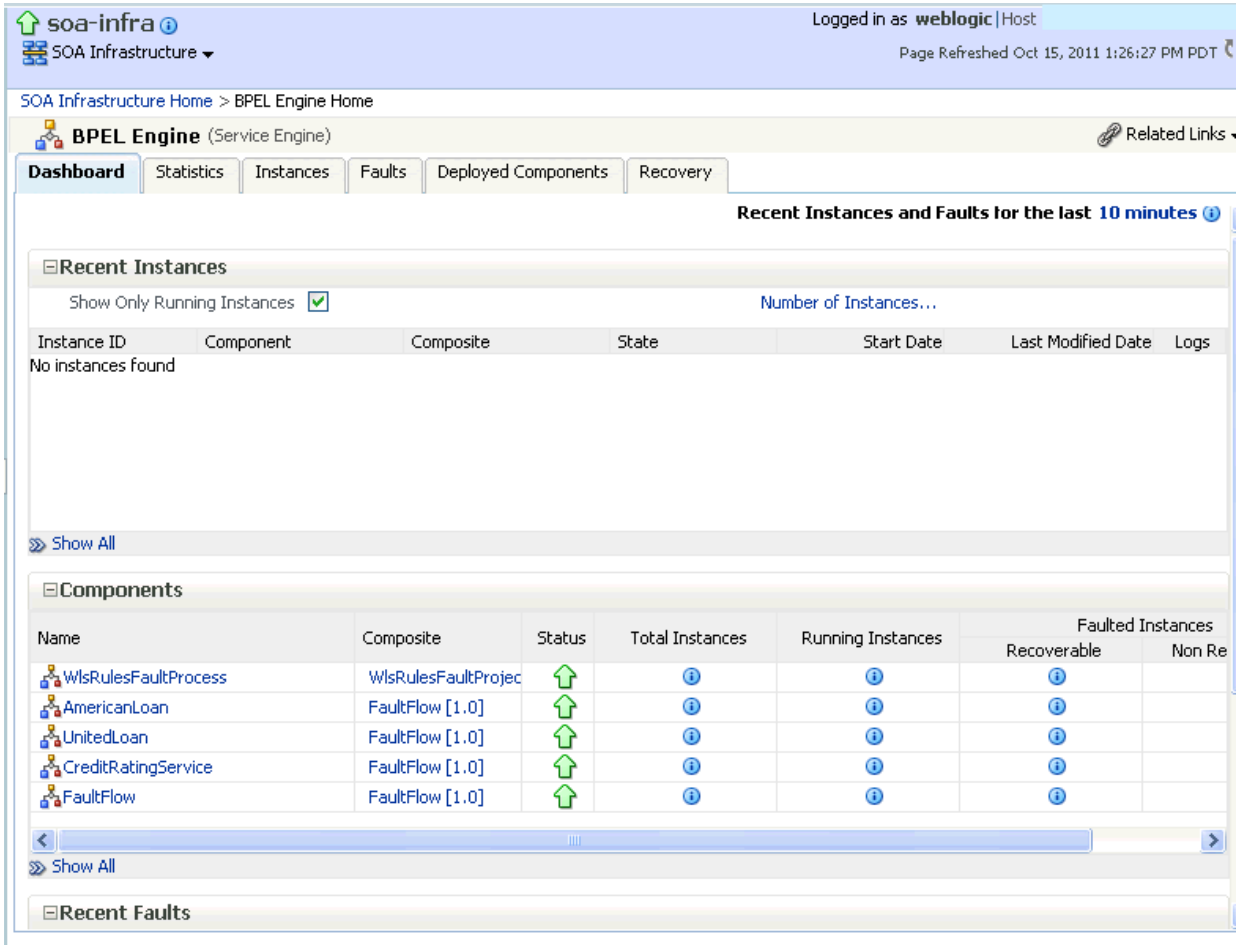
   d. In the **Composite** column, click a specific SOA composite application to access its home page.

   e. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

   f. Click **Show All** below the section to access the Instances page of the service engine.

   The lower part of the Dashboard page displays the following details:

   - The service components running in the service engine, the SOA composite applications of the service components, the state of the applications (for example, running), and the total, running, and faulted instances in the service engine.
   - The recent faults in the service engine, including the error message, whether you can recover from the fault, the time at which the fault occurred, the SOA composite application in which the fault occurred, the service component, the instance ID of the service component, the activity in which the fault occurred, and the log files describing the fault.



4. In the **Components** section, perform the following tasks:
   a. In the **Name** column, click a specific service component to access its home page.
   b. In the **Composite** column, click a specific SOA composite application to access its home page.
   c. Click **Show All** below the section to access the Deployed Components page of the service engine.
5. In the **Recent Faults** section, perform the following tasks:
   a. In the **Error Message** column, click an error message to display complete information about the fault. If the fault is identified as recoverable, click the **Recover Now** link to perform fault recovery.
   b. In the **Recovery** column, click a fault identified as **Recoverable** to perform fault recovery at the component instance level.
   c. In the **Composite** column, click a specific SOA composite application to access its home page.
   d. In the **Component** column, click a specific service component to access its home page.
   e. In the **Component Instance ID** column, click an instance ID for a service component to view its audit trail, process flow, sensor values, and faults.

f. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that fault.

For more information, see Section 1.2.4, "Introduction to Service Components and Service Component Instances."

13.6 Monitoring BPEL Process Service Engine Request and Thread Statistics

You can monitor request and thread statistics for all BPEL process service components running in the service engine.

To monitor BPEL process service engine request and thread statistics:

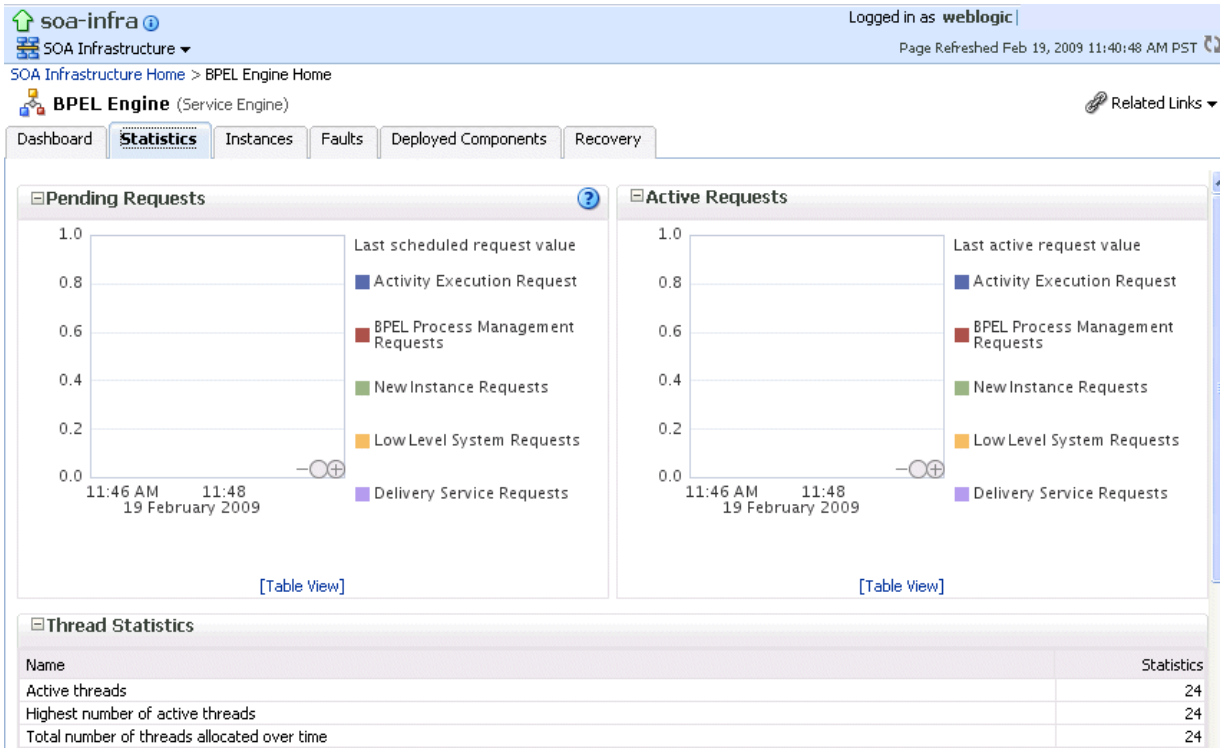
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **BPEL**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **BPEL**.

2. Click **Statistics**.

   The upper part of the Statistics page displays the following details. Click the **Help** icon for additional details.

   - Pending requests in the service engine
   - Active requests in the service engine
   - Thread statistics for the service engine



The lower part of the Statistics page displays details about the count and minimum, maximum, and average request processing times.

Request Breakdown

Name	Count	Min Request Processing Time (ms)	Max Request Processing Time (ms	Avg Request Processing Time (ms)
eng-composite-request	407	4.000	23,021.000	682.470
eng-single-request	620	0.000	23,021.000	553.500
load-workitem	212	2.000	435.000	34.910
load-wi-datasource	212	-1.000	27.000	1.200
eng-callback	212	0.000	408.000	31.950
initiate-correlation-set	161	-1.000	22.000	1.210
update-audit-trail	1	-1.000	-1.000	-1.000
eng-until	160	-1.000	1.000	0.040
sensor-send-activity-data	51	0.000	0.000	0.000
sensor-send-variable-data	52	0.000	0.000	0.000

13.7 Monitoring BPEL Process Service Engine Instances

You can monitor all BPEL process service component instances running in the service engine. These BPEL process service components can be part of separate SOA composite applications.

To monitor BPEL process service engine instances:

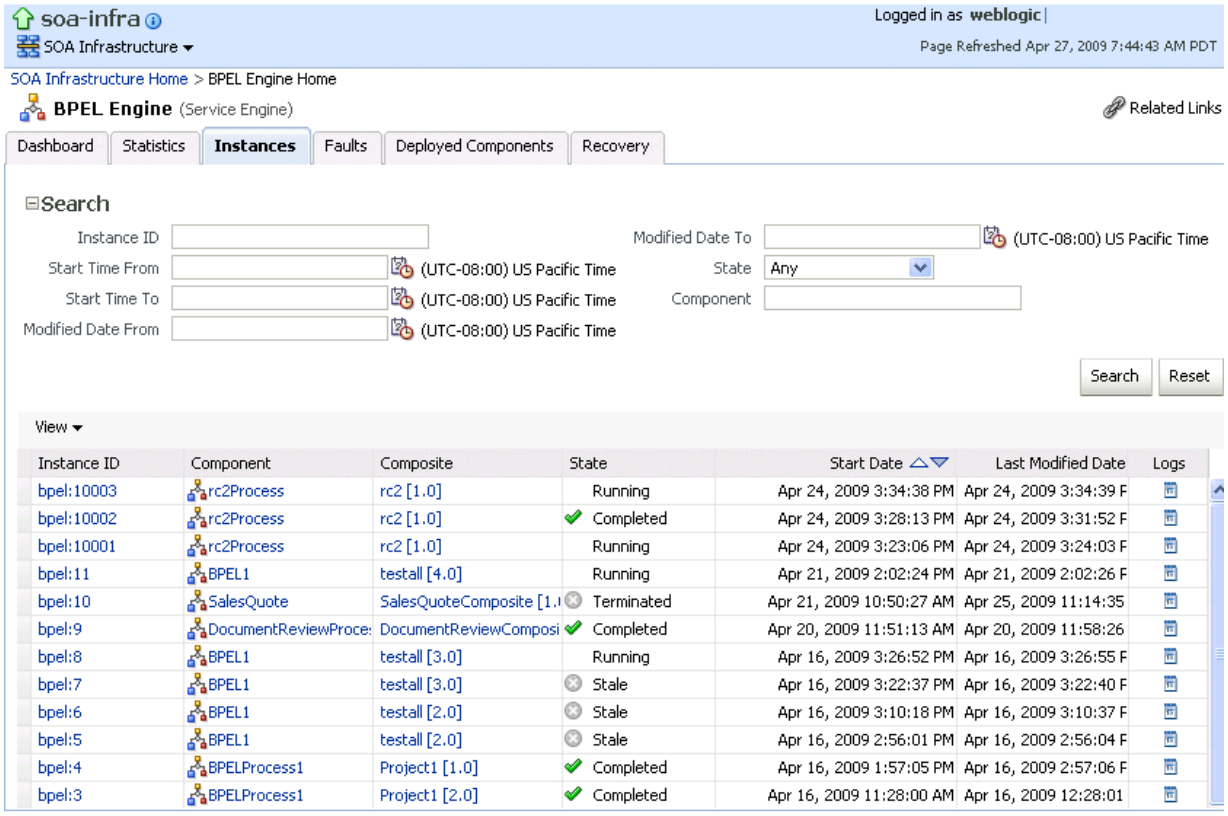
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **BPEL**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **BPEL**.

2. Click **Instances**.

   The Instances page displays the following details:

   - A utility for searching for a specific instance by specifying criteria and clicking **Search**.
   - Instances, including the instance ID of the service component, the service component name, the SOA composite application name, the state of the instance (for example, completed successfully, running, or faulted), the instance start time, the last modification time, and the log files describing the instance.



Note: @ Undeploying a SOA composite application instance does not cause the instance to complete. Undeployment marks running service component instances of that composite as stale. However, service component instances of that composite instance that are complete are not marked as stale.

---

3. In the **Instances** section, perform the following monitoring tasks:
   a. In the **Instance ID** column, click an instance ID for a service component to view its audit trail, process flow, sensor values, and faults.
   b. In the **Component** column, click a specific service component to access its home page.
   c. In the **Composite** column, click a specific SOA composite application to access its home page.
   d. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

For more information, see Section 1.2.4, "Introduction to Service Components and Service Component Instances."

13.8 Monitoring Deployed BPEL Process Service Components in the Service Engine

You can monitor all deployed SOA composite applications with BPEL process service components running in the service engine.

To monitor deployed BPEL processes in service engines:

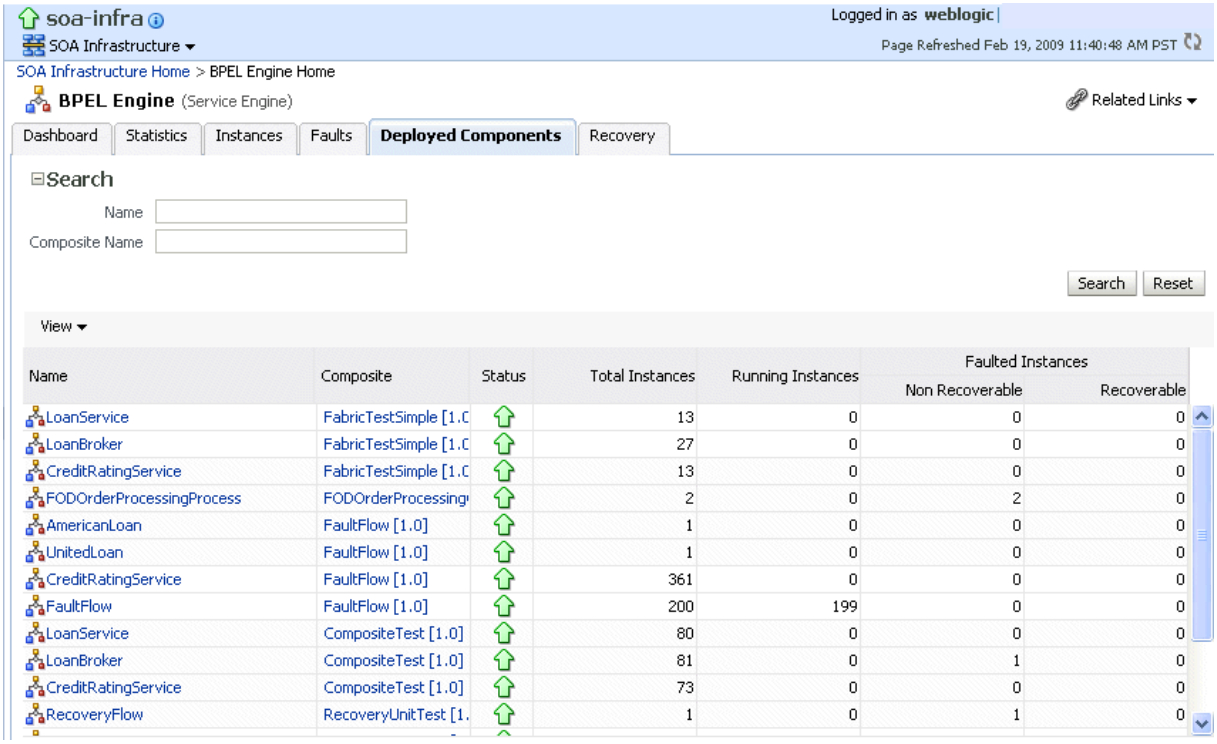
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **BPEL**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **BPEL**.

2. Click **Deployed Components**.

   The Deployed Components page displays the following details:

   - A utility for searching for a specific deployed SOA composite application by specifying criteria and clicking **Search**.
   - Details about deployed SOA composite applications with BPEL process service components running in this service engine, including the service component name, the SOA composite application, the current status, and the total, running, and faulted instances in the service engine.



3. In the **Name** column, click a specific service component to access its home page.
4. In the **Composite** column, click a specific SOA composite application to access its home page.

14

Managing BPEL Process Service Components and Engines

This chapter describes how to manage BPEL process service components and service engines, including recovering from service component and service engine faults, managing service component policies, performing BPEL process message recovery, and storing instance and callback message data in Oracle Coherence cache.

This chapter includes the following topics:

- Section 14.1, "Recovering from BPEL Process Service Component Faults"
- Section 14.2, "Managing BPEL Process Service Component Policies"
- Section 14.3, "Recovering from BPEL Process Service Engine Faults"
- Section 14.4, "Performing BPEL Process Service Engine Message Recovery"
- Section 14.5, "Storing Instance and Callback Message Data in Oracle Coherence Distributed Cache on Oracle Exalogic Platforms"

For more information, see the following sections:

- Section 1.2.4, "Introduction to Service Components and Service Component Instances"
- Section 1.2.6, "Introduction to Service Engines"

14.1 Recovering from BPEL Process Service Component Faults

You can monitor and perform individual and bulk fault recoveries for BPEL process service components that are identified as recoverable. For BPEL process faults to be identified as recoverable, there must be a fault policy defined that is bound to the fault (through the `fault-bindings.xml` file) and which triggers the action `ora-human-intervention`. However, without defining any fault policies, the fault takes its standard course as either a recoverable or nonrecoverable fault.

To recover from BPEL process service component faults:

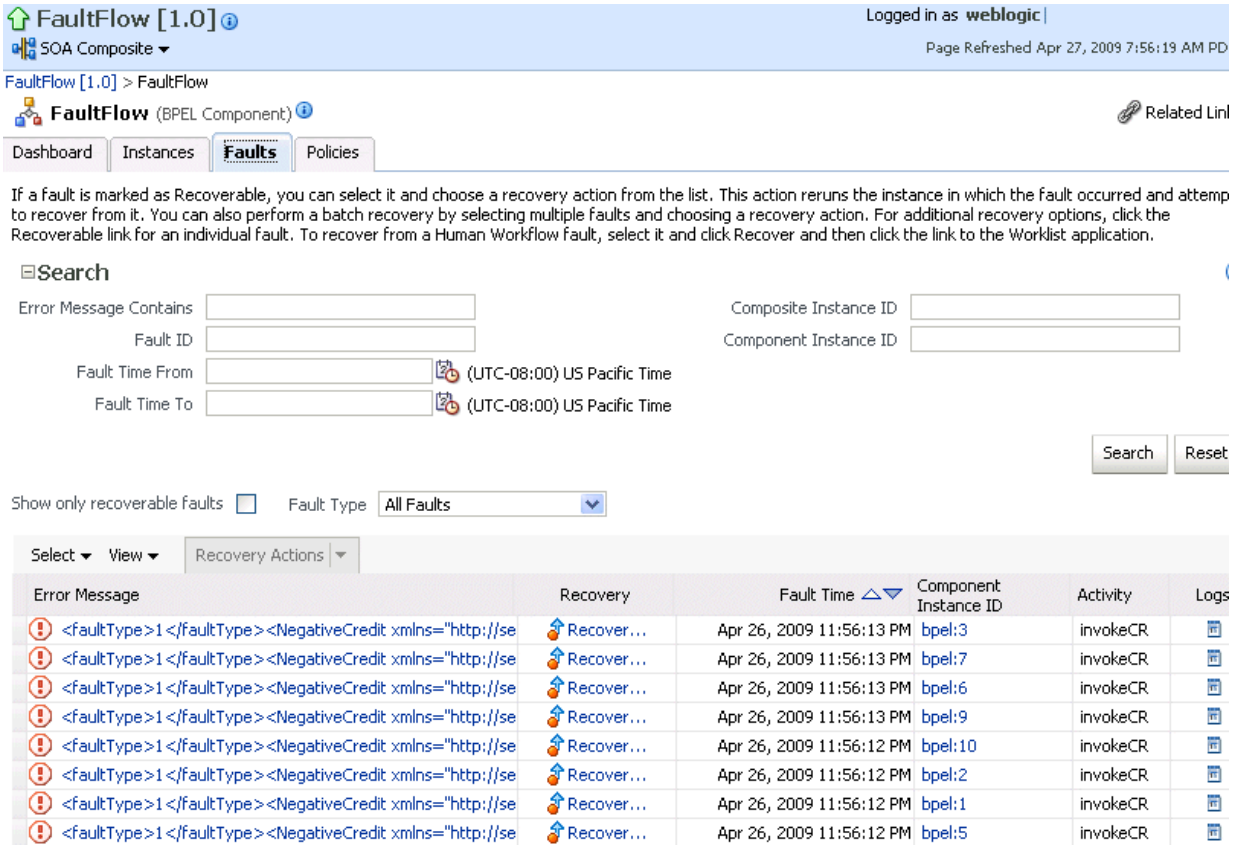
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**.	1. Under **soa-infra**, expand the partition.
2. Select the **Deployed Composites** tab.	2. Select a specific SOA composite application.
3. In the **Composite** section, select a specific SOA composite application.	

2. Select the BPEL process service component in the **Component Metrics** section.
3. Click **Faults**.

   The Faults page displays the following details:

   - A utility for searching for a specific fault by specifying criteria and clicking **Search**. Click the **Help** icon for details.
   - Faults that occurred in the service component, including the fault ID, error message, whether you can recover from the fault, time at which the fault occurred, service component instance ID, activity in which the fault occurred, and a link to a log file describing the fault.



   BPEL process service component faults identified as recoverable can be recovered.

4. Select faults for recovery using one of the following methods. Fault recovery selection at the BPEL process service component level equals the SOA Infrastructure level, SOA composite application level, and Oracle Mediator service component level.

For...	Then...
Single fault recovery	There are three options from which to choose for single-fault recovery: **1.** Click the row of the fault that has been identified as recoverable. With the row highlighted, select a specific action from the **Recovery Action** list, as described in Step 5. **2.** In the **Recovery** column, click the **Recover** link to access the Faults page of the instance audit trail to perform fault recovery. **3.** In the **Error Message** column, click the message of a fault that has been identified as recoverable. This displays complete fault details, including the fault ID, fault time, fault location, fault type, and error message text. A **Recover Now** option is displayed for recoverable faults. Click **Recover Now** to access the Faults page of the instance audit trail to perform fault recovery.
Bulk fault recovery	There are two options from which to choose for bulk-fault recovery: **1.** Use Shift+Click or Control+Click to select specific faults in the rows. or **2.** From the **Select** menu, choose **Select All Recoverable**. Then use Shift+Click or Control+Click to deselect the faults to *not* include in the recovery operation. Then: **3.** Select an action from the **Recovery Action** list, as described in Step 5. **Note:** Only the actions applicable to all selected faults are available.
Recovery of all faults	**1.** From the **Select** menu, choose **Select All Recoverable**. **2.** Select an action from the **Recovery Action** list, as described in Step 5. **Note:** Only the actions applicable to all selected faults are available.

Note: In most cases, fault policy actions are automatically executed. The only exception is if you defined a fault policy that uses the action `ora-human-intervention`. This action creates a recoverable fault that can be recovered from Oracle Enterprise Manager Fusion Middleware Control.

5. Select an action from the **Recovery Action** list.

Action	Description
Retry	Retries the instance directly. An example of a scenario in which to use this recovery action is when the fault occurred because the service provider was not reachable due to a network error. The network error is now resolved.
Abort	Terminates the entire instance.
Replay	Replays the entire scope activity again in which the fault occurred.
Rethrow	Rethrows the current fault. BPEL fault handlers (catch branches) are used to handle the fault. By default, all exceptions are caught by the fault management framework unless an explicit rethrow fault policy is provided.
Continue	Ignores the fault and continues processing (marks the faulted activity as a success).

6. Perform the following additional monitoring tasks from within the faults table:

a. Click the **Show only recoverable faults** checkbox to display only faults from which you can recover.

b. From the **Fault Type** list, select to display all faults, system faults, business faults, or Oracle Web Services Manager (OWSM) faults in the faults table. Click the **Help** icon for a description of these fault types.

c. From the **View** list, select **Columns** > **Fault ID** to display the fault IDs for each error message. The fault ID is automatically generated and uniquely identifies a fault. The fault ID is also displayed when you click an error message.

d. In the **Component Instance ID** column, click a specific service component ID to access task details about the instance (for example, the current state of a task). Rejected messages do not have a component instance ID.

e. In the **Logs** column, click a link to access the Log Messages page with filtered messages specific to that instance.

For more information, see the following documentation:

- Section 1.4.3.1, "Introduction to Fault Recovery"
- Section 8.4.1, "Examples of Fault Recovery for BPEL Processes"
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*
- *Oracle Fusion Middleware Administrator's Guide* for details about viewing and searching log files

14.2 Managing BPEL Process Service Component Policies

You can attach and detach policies to and from BPEL process service components in currently deployed SOA composite applications. Policies apply security to the delivery of messages. Oracle Fusion Middleware uses a policy-based model to manage web services.

> **Note:** Before attaching policies, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for definitions of available policies and details about which ones to use in your environment.

To manage BPEL process service component policies:

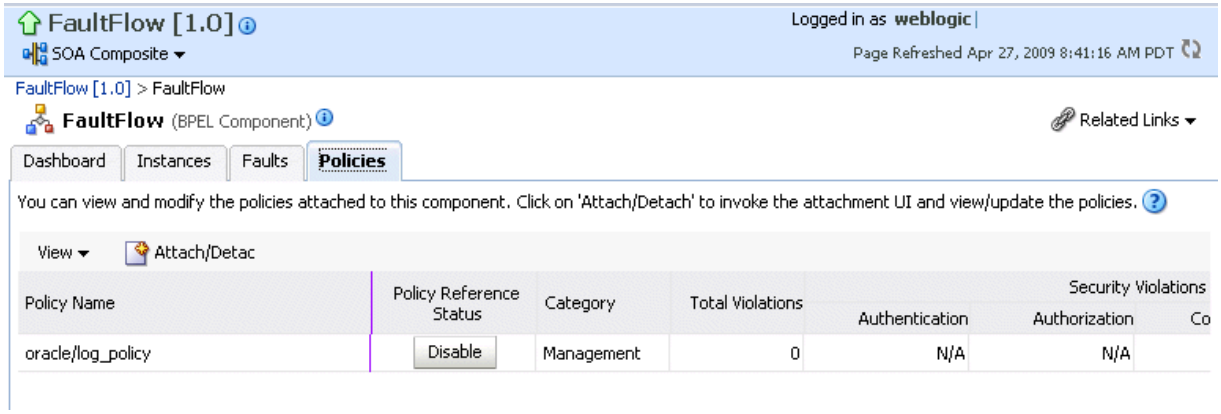
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, expand the partition. 2. Select a specific SOA composite application.

2. Select the BPEL process service component in the **Component Metrics** section.
3. Click **Policies**.

   The Policies page enables you to attach and detach policies to and from BPEL process service components. The **Policies** section displays the attached policy name, the policy reference status (enabled or disabled) that you can toggle, the category (Management, Reliable Messaging, MTOM Attachment, Security, or

WS-Addressing), the violations, and the authentication, authorization, confidentiality, and integrity failures since the SOA Infrastructure was last restarted.



4. Click **Attach/Detach**.

   If multiple components are available, you are prompted to select the service or component for which to perform the attachment or detachment.

5. Select the service or component to which to attach or detach a policy.

   This invokes a dialog for attaching or detaching policies.

   Policies currently attached appear in the **Attached Policies** section. Additional policies available for attachment appear in the **Available Policies** section.

6. Select to attach policies appropriate to your environment.

7. Click **Attach**.

8. When you are finished attaching policies, click **Validate**.

9. If an error message appears, make the necessary corrections until you no longer have any validation errors.

10. Click **OK**.

    The attached policy is displayed in the policies table.

For more information, see the following documentation:

- Section 1.4.3.2, "Introduction to Policies"
- Section 7.6, "Managing SOA Composite Application Policies" for the dialogs that are displayed during policy attachment
- *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for definitions of available policies and details about which ones to use for your environment

14.3 Recovering from BPEL Process Service Engine Faults

You can monitor and perform individual and bulk recoveries of faults occurring in BPEL process service engines that are identified as recoverable. All BPEL process service component faults, regardless of the SOA composite application instance of which they are a part, can be viewed in the BPEL process service engine. For BPEL process faults to be identified as recoverable, there must be a fault policy defined that is bound to the fault (through the `fault-bindings.xml` file) and which triggers the

action `ora-human-intervention`. However, without defining any fault policies, the fault takes its standard course as either a recoverable or nonrecoverable fault.

To recover from BPEL process service engine faults:

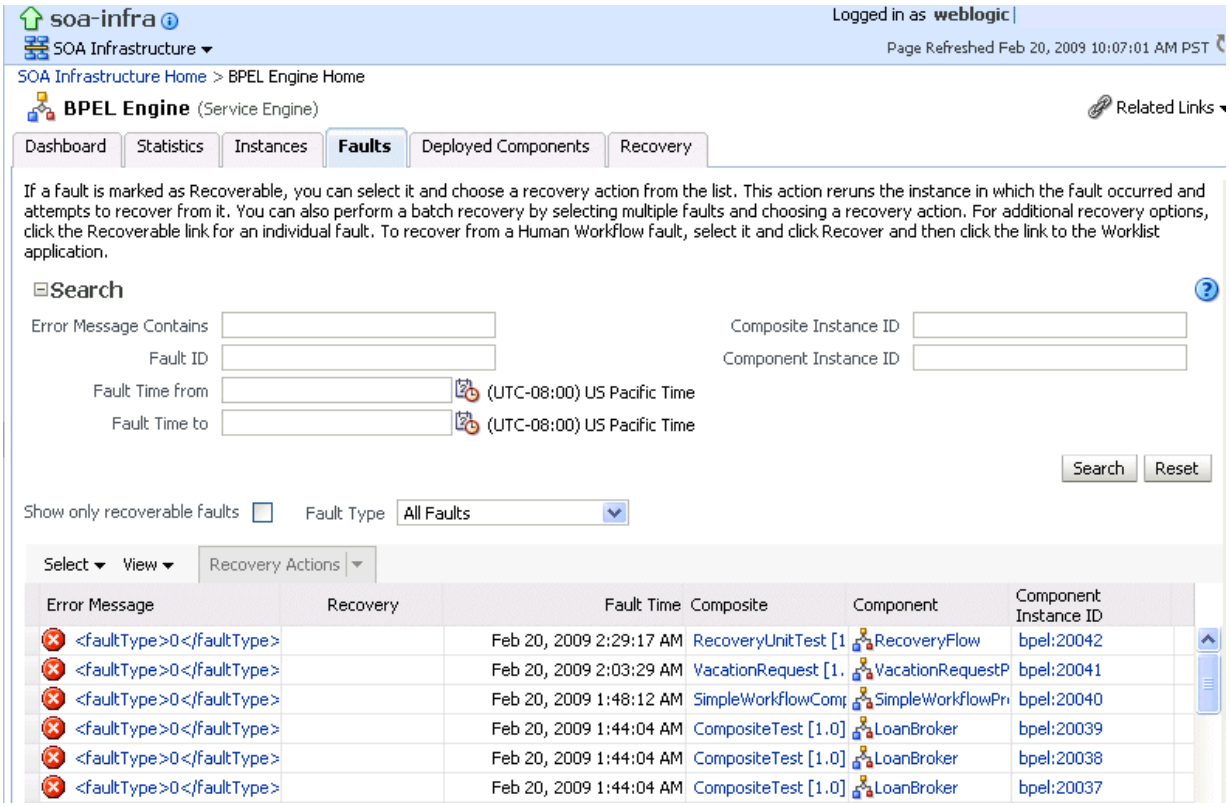
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **BPEL**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **BPEL**.

2. Click **Faults**.

   The Faults page displays the following details:

   - A utility for searching for a specific fault by specifying criteria and clicking **Search**. Click the **Help** icon for details.
   - Faults that occurred in the service engine, including the fault ID, the error message, whether you can recover from the fault, the time at which the fault occurred, the SOA composite application and service component in which the fault occurred, and the service component instance ID.



   BPEL process service engine faults identified as recoverable can be recovered.

3. Select faults for recovery using one of the following options. As with fault recovery at the SOA Infrastructure level, SOA composite application level, and Oracle Mediator service component level, you can perform single fault recovery, bulk fault recovery, and recovery of all faults. See Step 4 of Section 14.1, "Recovering from BPEL Process Service Component Faults" for instructions on selecting faults to perform these types of recovery.

> **Note:** In most cases, fault policy actions are automatically executed. The only exception is if you defined a fault policy that uses the action `ora-human-intervention`. This action creates a recoverable fault that can be recovered from Oracle Enterprise Manager Fusion Middleware Control.

4. Select an action from the **Recovery Action** list.

Action	Description
Retry	Retries the instance with an option to provide a retry success action. An example of a scenario in which to use this recovery action is when the fault occurred because the service provider was not reachable due to a network error. The network error is now resolved.
Abort	Terminates the entire instance.
Replay	Replays the entire scope activity again in which the fault occurred.
Rethrow	Rethrows the current fault. BPEL fault handlers (catch branches) are used to handle the fault. By default, all exceptions are caught by the fault management framework unless an explicit rethrow fault policy is provided.
Continue	Ignores the fault and continues processing (marks the faulted activity as a success).

5. Perform the following additional monitoring tasks from within the faults table:
   a. Click the **Show only recoverable faults** checkbox to only display faults from which you can recover.
   b. From the **Fault Type** list, select to display all faults, system faults, business faults, or OWSM faults in the faults table. Click the **Help** icon for a description of these fault types.
   c. From the **View** list, select **Columns** > **Fault ID** to display the fault IDs for each error message. The fault ID is automatically generated and uniquely identifies a fault. The fault ID is also displayed when you click an error message.
   d. In the **Composite** column, click a specific SOA composite application to access its home page.
   e. In the **Component** column, click a specific service component to access its home page.
   f. In the **Component Instance ID** column, click a specific service component ID to access task details about the instance (for example, the current state of a task). Rejected messages do not have a component instance ID.

For more information, see the following sections:

- Section 1.4.3.1, "Introduction to Fault Recovery" for conceptual details about faults
- Section 8.4.1, "Examples of Fault Recovery for BPEL Processes"

14.4 Performing BPEL Process Service Engine Message Recovery

You can perform a manual recovery of undelivered invoke or callback messages due to a transaction rollback in the process instance. Recovery of invoke messages applies to asynchronous BPEL processes only. Synchronous BPEL processes return an error to the

calling client and are not recoverable from the Recovery page. Recoverable activities are activities that failed and can be recovered. For example, if you are using the file adapter to initiate an asynchronous BPEL process and your system fails while the instance is processing, you can manually perform recovery when the server restarts to ensure that all message records are recovered.

You can also manage messages that have failed automatic recovery attempts by the BPEL process service engine. To ensure that automatic recovery of these messages is not attempted multiple times, these messages are placed in the exhausted state. You can then perform one of the following actions on these messages:

- Return them to the automatic recovery queue
- Never attempt a recovery on them again
- Attempt to recover them immediately

For example, assume you have a BPEL process that writes to a database adapter. If the database is down, these messages are sent to a recovery queue. Automatic recovery of these messages fails while the database is down. Such messages are marked with the exhausted state so that automatic recovery is not attempted on them again. When the database begins running again, you can reset these messages (return them to the automatic recovery queue) so that an automatic recovery is attempted on them again.

To perform BPEL process service engine message recovery:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **BPEL**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **BPEL**.

2. Click **Recovery**.

   The Recovery page displays the following details:

   - A **Refresh Alarm Table** button for resynchronizing lost, in-memory, Quartz-scheduled jobs in the database. For example, assume a timer on a wait activity or on an onAlarm branch of a pick activity was initiated, but the transaction was rolled back. You can resynchronize these jobs with the BPEL instances residing in the wait activity/onAlarm branch in the database.
   - A utility for searching for a specific message failure by specifying criteria and clicking **Search**. Click the **Help** icon for details.

     You can enter the execution context ID (ECID) value in the **ECID** field. The ECID value enables you to track a message flow that crosses instances of different composite applications. If there are BPEL process messages requiring recovery and the **AuditConfig** property in the System MBean Browser is set to **All** (the default value), the following message is displayed in the **Trace** table of the Flow Trace page:

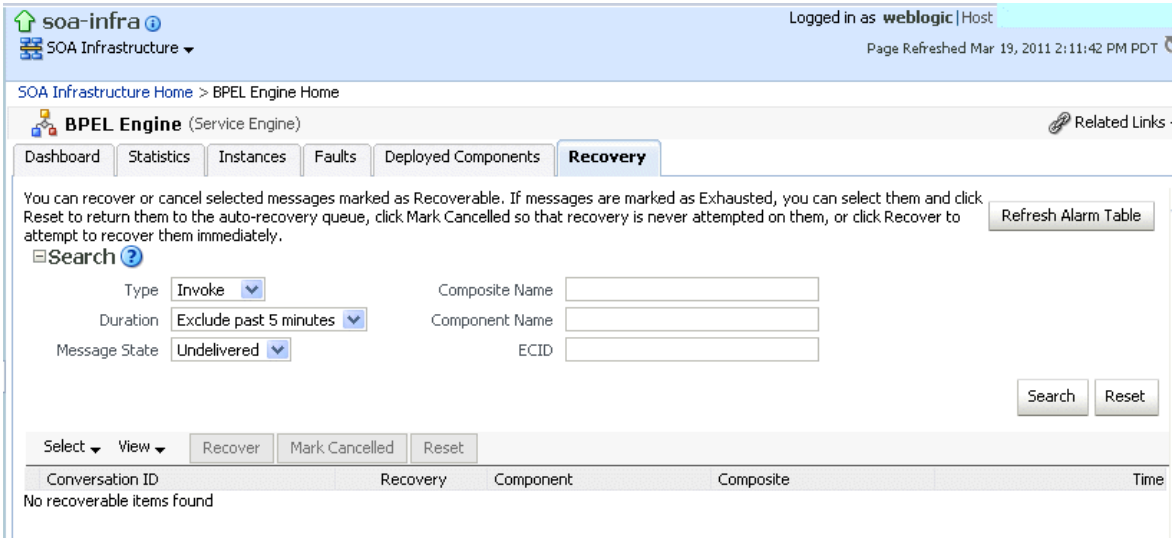
     ```
     BPEL Message Recovery Required
     ```

     Clicking **Show Details** or the recovery icon that appears next to this message displays a Warning dialog with information about the number of invoke, callback, and activity recoverable message types and the ECID value. You can copy the ECID value from the Warning dialog, paste it into the **ECID** field, and select the recoverable message type from the **Type** list as part of creating your search criteria on the Recovery page.

For more information, see Section 13.1, "Monitoring the Audit Trail and Process Flow of a BPEL Process Service Component."

Note: Oracle recommends that you add an index on the `DLV_MESSAGE.ECID` column of the `DLV_MESSAGE` table to improve SQL query performance when searching messages for a specific ECID value. This is because if there are too many entries in the `DLV_MESSAGE` table, the search query may be slow and may also overload the database. For information on adding an index, see Chapter "Creating Indexes" of the *Oracle Database Administrator's Guide*.

- Message failures in the service engine, including the conversation ID, whether you can recover from the message failure, the service component and composite application in which the failure occurred, and the time at which the fault occurred. Depending on the state, you can recover these messages immediately, cancel these messages, or reset these messages for automatic recovery.



Notes:

- You can recover callback messages in resolved and undelivered states. These messages can be displayed for recovery when you execute search criteria in which you select **Callback** from the **Type** list and either **Resolved** or **Undelivered** from the **Message State** list. When a callback message first enters the BPEL process service engine, its state is undelivered. When this message is resolved to the target BPEL process instance either through matching a conversation ID or a correlation, the state is switched to resolved. In both of these states, the messages have not yet been consumed. Messages in these two states can be recovered (redelivered into the BPEL process service engine for consumption). In other situations, the callback messages can become stranded in both of these states. Messages in these states can also be recovered. However, there is no guarantee that stranded callback messages always remain in an undelivered state.
- If you select **Invoke** from the **Type** list and **Undelivered** from the **Message State** list, and then click **Recover**, a recovery is performed. However, the **Last Modified Date** column remains empty for this instance on the Dashboard page of the Oracle BPEL Process Manager service component or service engine. This is the expected behavior. The last modified date is not displayed because the initial Oracle BPEL Process Manager instance (for example, **bpel:70004**) is created by the first invocation (that is, it is created, but has not yet been modified). The recovery of the undelivered invocation message always creates a *new* instance (for example, **bpel:70005**). The previously created instance (**bpel:70004**) is not used and remains permanently in the same status (the **Last Modified Date** column is empty). This information is provided for auditing purposes only.
- The **Message States** list is applicable only to callback and invoke message type recovery, and not for activity message type recovery.

3. Select a fault in the table.
4. Select one of the following options:

Action	Description
Recover	Retries the message in which the fault occurred. If you select messages in the exhausted state and click this button, an attempt is made to recover them immediately. Should this recovery attempt also fail, the message is returned to the exhausted state. You must then select the message and click **Reset** to return the message to the automatic recovery queue. If an asynchronous BPEL process encounters a transaction rollback scenario because of any underlying exception error, it rolls back to the last dehydration activity. If this is a new instance, and a receive activity was the first dehydration activity, the BPEL process service engine creates a recoverable invoke. When you click **Recover** to recover the invoke, the service engine creates a new instance. This instance may run to completion with no exception error. However, you continue to see the older instance identified as faulted.

Action	Description
Mark Cancelled	Marks the message so it is never delivered. If you select messages in the exhausted state and click this button, recovery is never attempted on them. **Note**: The **Recover** and **Mark Cancelled** buttons are enabled in the following situations: ▪ For users with the administrator or operator role. These buttons are disabled for users with the monitor role. ▪ In the context of a recoverable message. You must select a recoverable message for these buttons to be enabled. For more information about roles, see Appendix C.1, "Roles and Privileges."
Reset	Select to reset exhausted messages to the undelivered state. This returns the message to the automatic recovery queue. The messages that are displayed in the exhausted state disappear from the messages table. If you select **Undelivered** from the **Message State** list and click **Search**, these messages are displayed. Callback messages in the exhausted state can also be reset to the resolved state and still remain recoverable.

Once a message is submitted for recovery, the BPEL process service engine may take time to complete the action. This typically takes less than several seconds. During this time, the message remains visible in the Recovery page. Duplicate attempts to recover the same message in that period are ignored. Refresh the page every few seconds to receive the latest recovery status.

Note: If you define a fault policy in a BPEL process with an `ora-retry` action and a fault occurs, the BPEL process attempts to recover from the fault the number of times you specified with the `retryCount` parameter. After this period, the process continues to be in a running state. The status of an activity in the process that has not completed (such as an invoke or receive) shows as pending a manual recovery. This is the expected behavior.

For information about configuring the maximum number of times to attempt an invoke and callback message recovery, see Section 12.4, "Configuring Automatic Recovery Attempts for Invoke and Callback Messages."

14.5 Storing Instance and Callback Message Data in Oracle Coherence Distributed Cache on Oracle Exalogic Platforms

With BPEL processes, a potential performance issue is the number of database interactions required per instance. This factor is the main reason for synchronous transient flows outperforming asynchronous durable flows. You can design around this issue by utilizing synchronous transient flows in situations where low response times are required. However, you may be unable to design this type of flow for business reasons.

If you are running Oracle SOA Suite 11*g* Release 1 11.1.1.6 on an Oracle Exalogic platform, you can use the distributed cache feature of Oracle Coherence to store instance and callback message data from BPEL processes. This eliminates database reads, thereby reducing the number of database interactions.

Oracle Coherence is a component of Oracle Fusion Middleware that enables organizations to scale mission-critical applications by providing access to frequently used data. Oracle Coherence includes a distributed cache feature that provides scalability for both read and write access. Data is automatically, dynamically, and transparently partitioned across nodes. The distribution algorithm minimizes network traffic and avoids service pauses by incrementally shifting data.

Oracle Exalogic is an integrated hardware and software system designed to provide a platform for a range of application types and varied workloads. Oracle Exalogic is intended for large-scale, performance-sensitive, mission-critical application deployments.

Note: If your environment is not using Oracle Exalogic, Oracle Coherence distributed cache is not available.

The potential performance gains of using a distributed cache for BPEL processes are as follows:

- Eliminates the read required for messages (invoke and callback) from the database after initial delivery
- Eliminates the read required for cube instances after a dehydration point

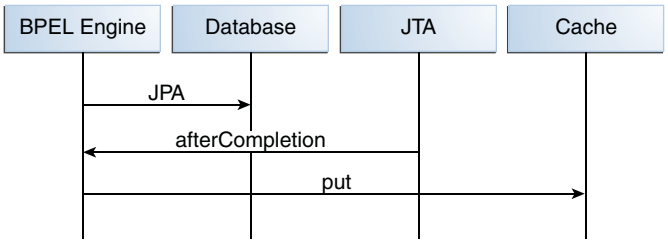
For more information about Oracle Coherence, see the *Oracle Coherence Getting Started Guide* and *Oracle Coherence Developer's Guide*.

For more information about Oracle Exalogic, see the *Oracle Exalogic Machine Owner's Guide*.

14.5.1 Introduction to the Oracle Coherence Caching Architecture

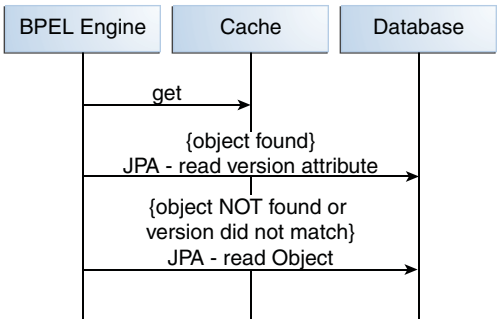
During dehydration, instance objects are stored in the database using the Java Persistence API (JPA) in a container-managed Enterprise JavaBeans (EJB) transaction. The BPEL process service engine registers the transaction `afterCompletion` listener for post-transaction processing. Instance objects modified during a transaction are tracked and made available to the `afterCompletion` listener, which updates the cache. Figure 14–1 provides details about the dehydration process.

Figure 14–1 Dehydration Process



During rehydration, instance objects are read from cache. JTA implementations do not provide XA guarantees for transaction completion notification, and cache eviction may delete the object from cache. JTA implementations account for these two scenarios and address the issues of cache not returning an object or returning an older version of the object. Figure 14–2 provides details about the rehydration process.

Figure 14–2 Rehydration Process



If the instance object is not found in cache, it is loaded from the database. In this scenario, cache does not provide any benefits. However, if you configure the correct amount of cache for your environment, this is not common. The correct cache size varies based on your environment. Therefore, Oracle Coherence configuration must be available and you should be able to configure it.

To identify object staleness, the `CACHE_VERSION` column in the database table is queried and compared to the corresponding `cacheVersion` object attribute. If the comparison fails, instance objects are loaded from the database. In this scenario, cache provides no benefit. However, this is a rare occurrence.

Cache lookup usually provides a valid object. In this scenario, performance gain for dehydration and rehydration using cache over direct writes (the default) equals the following:

```
(database read time + relational to object mapping) minus (Object serialization +
reading from serialized form + Coherence network overhead + query to
database for reading CACHE_VERSION)
```

It also reduces activity on the database server.

If Oracle Coherence cache is not available due to a network issue, the BPEL process service engine continues to work. If there are no errors, business process instances continue to progress. The BPEL process service engine checks for the availability of cache, resuming its usage when it is back online. While cache is unavailable, warning messages are logged and you may notice a decrease in performance.

14.5.2 Running with Default SOA Cluster Nodes and Coherence Cache Grid Nodes

BPEL process caches are not created on an Oracle SOA Suite cluster node. You must start the BPEL cache servers, which host the BPEL caches, by following the instructions in Section 14.5.5, "Starting the BPEL Process Cache Servers." Start at least four servers to observe an increase in performance.

There is no requirement for ordering of an Oracle SOA Suite cluster and BPEL cache servers. The BPEL process service engine continues to function without BPEL cache servers, even when the **qualityOfService** property is set to **CacheEnabled** in Oracle Enterprise Manager Fusion Middleware Control.

For more information about Oracle Coherence, see the *Oracle Coherence Getting Started Guide* and *Oracle Coherence Developer's Guide*.

14.5.3 Configuring Oracle Coherence Caching

The System MBean Browser property **QualityOfService** enables you to configure Oracle Coherence for dehydration. You must configure this property on all nodes in the SOA cluster.

To configure Oracle Coherence for dehydration:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **SOA Administration** > **BPEL Properties**.	1. Right-click **soa-infra**. 2. Select **SOA Administration** > **BPEL Properties**.

   The BPEL Service Engine Properties page is displayed.

2. Click **More BPEL Configuration Properties**.
3. In the **Attributes** tab, click **QualityOfService**.
4. In the **Value** field, enter a value appropriate to your environment. This change does not require a SOA Infrastructure restart.

Table 14–1 qualityOfService Values

Value	Description
DirectWrite	No cache is used for dehydration and rehydration. Read and write operations are done to the database. This is the default setting.
CacheEnabled	During dehydration, the instance data is stored in the database using an XA data source connection; the placement of objects into cache is part of post-transaction processing. During rehydration, data is fetched from the cache. If the data is not found (for example, the BPEL process cache servers are not available) or the version is stale, data is read from the database.

5. Click **Apply**.

14.5.4 Configuring the Storage of Multiple Audit Trail Messages in One Transaction

For asynchronous flows, performance gains can be achieved by storing multiple audit messages in one transaction. To improve performance, you can store multiple audit trail messages (across instances) in a single transaction by setting the System MBean browser property **AsynchAuditBatchSize** in Oracle Enterprise Manager Fusion Middleware Control.

Setting this property to an appropriate value reduces audit trail transaction commits. Instead, a commit is only performed when a specified limit is reached.

To configure the storage of multiple audit messages in one transaction:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **SOA Administration** > **BPEL Properties**.	1. Right-click **soa-infra**. 2. Select **SOA Administration** > **BPEL Properties**.

The BPEL Service Engine Properties page is displayed.

2. Click **More BPEL Configuration Properties**.
3. In the **Attributes** tab, click **AsynchAuditBatchSize**.
4. In the **Value** field, enter a value appropriate to your environment. The default value of `-1` indicates that there is no audit trail message batching. Each audit message is persisted in its own transaction.

   The recommended value range is `5` to `25`. For example, if you set this property to `8`, this indicates that when eight audit trail messages have accumulated, a transaction is created with these messages and committed to the dehydration store.

   This parameter only impacts Oracle Exalogic environments. For other environments, it is not operational.

   This change does not require a SOA Infrastructure restart.
5. Click **Apply**.

14.5.5 Starting the BPEL Process Cache Servers

You must run the `start-bpel-cache.sh`. script to start the BPEL process cache servers. This script must be run on UNIX platforms. This script is independent of the SOA cluster. It can be on the same node as the SOA cluster nodes or on a completely different host. The only requirement is network connectivity. The Oracle SOA Suite nodes must be reachable from the host on which the BPEL process cache servers are installed.

This script joins an Oracle SOA Suite cluster with a multicast, default address and port. These values match with the corresponding values in the `$FMW_HOME/user_projects/domains/`*`domain_name`*`/bin/setDomainEnv.sh` or `setDomainEnv.bat` file.

If you choose multicast for a cluster, but use a different address and port, you can override it in the `bpelCacheEnv.sh` file by using an environment variable or setting a shell variable. Use the same values for SOA managed servers (`setDomainEnv.sh` or `setDomainEnv.bat`).

The default cache configuration for the Oracle SOA Suite cluster must be unicast, and not multicast. For more information about this recommended cache configuration for Oracle SOA Suite clusters for Oracle Coherence, see the *Oracle Fusion Middleware High Availability Guide* or *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite* for details.

To start the BPEL process cache servers:

1. Go to the `$FMW_HOME/SOA_ORACLE_HOME/bin` directory.
2. Open the `start-bpel-cache.sh` file.
3. Follow the instructions inside this file to create the `bpelCacheEnv.sh` file and configure various environment variables.

   Environment/shell variable names and value formats are described in the initial notes section of the `start-bpel-cache.sh` file.
4. Ensure that you first set the **QualityOfService** property to **CacheEnabled** in Oracle Enterprise Manager Fusion Middleware Control, as described in Section 14.5.3, "Configuring Oracle Coherence Caching."
5. Go to the `$FMW_HOME/SOA_ORACLE_HOME/bin` directory.

6. Run the following script:

```
start-bpel-cache.sh
```

Part VI

Administering Oracle Mediator Service Components and Engines

This part describes how to administer Oracle Mediator service components and engines.

This part includes the following chapters:

- Chapter 15, "Configuring Oracle Mediator Service Components and Engines"
- Chapter 16, "Monitoring Oracle Mediator Service Components and Engines"
- Chapter 17, "Managing Oracle Mediator Service Components and Engines"
- Chapter 18, "Managing Cross-References"

15

Configuring Oracle Mediator Service Components and Engines

This chapter describes how to configure runtime properties for Oracle Mediator service components and service engines using Oracle Enterprise Manager Fusion Middleware Control. You can also configure advanced and custom properties for Mediator.

This chapter includes the following topics:

- Section 15.1, "Configuring Oracle Mediator Service Engine Properties"
- Section 15.2, "Configuring Resequenced Messages"

15.1 Configuring Oracle Mediator Service Engine Properties

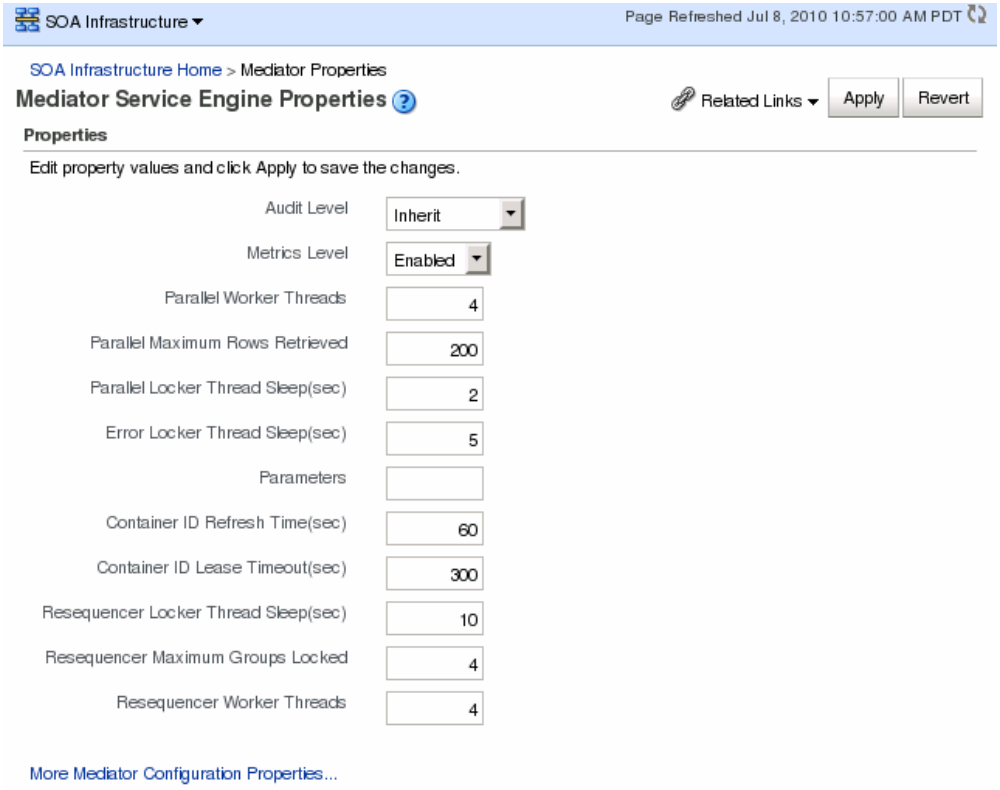
Mediator service engine properties are used by the Mediator service engine during processing of Mediator service components.

To configure Mediator service engine properties:

1. Access the Mediator Service Engine Properties page using one of the following methods:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **SOA Administration** > **Mediator Properties**.	1. Right-click **soa-infra**. 2. Select **SOA Administration** > **Mediator Properties**.

The Mediator Service Engine Properties page displays a list of Mediator properties, as shown in the following figure.



2. Make changes to the service engine properties that are appropriate to your environment.

   The properties are described in Table 15–1 below.

Table 15–1 Mediator Service Engine Properties

Property	Description
Audit Level	The Mediator-specific audit level. The value of this property overrides the value of the global SOA Infrastructure audit level property. The possible values of this property are: ■ **Off**: Switches off auditing for Mediator. Composite instance tracking and payload tracking information are not collected. ■ **Inherit**: The level of audit is the same as the SOA infrastructure. This setting enables the Mediator audit level to automatically change, when the global setting is changed. Setting a different audit level tracking, for this page, overrides the tracking set at the SOA Infrastructure level. ■ **Production**: All events are logged. All audit details, except the details of assign activities, are logged. Instance tracking information is collected, but payload details are not captured and these details are not available in the flow audit trails. This level is optimal for most typical operations and testing. ■ **Development**: All events and all audit details are logged. In addition, payload details are captured and are available in the flow audit trails. This level is useful for debugging purposes, but may impact performance. The default value of this flag is **Inherit**. **Notes:** ■ You do not need to restart the server after changing these properties. ■ Audit levels were known as instance tracking levels in the Oracle Application Server 10*g* releases.
Metrics Level	The Mediator-specific property for configuring the Dynamic Monitoring Service (DMS) metrics level. DMS metrics are used to measure the performance of application components. The possible values of this property are: ■ **Enabled**: Enables DMS metrics tracking ■ **Disabled**: Disables DMS metrics tracking
Parallel Worker Threads	The number of parallel dispatchers for message processing. Increase this parameter to increase the number of outbound threads for parallel processing.
Parallel Maximum Rows Retrieved	The number of rows retrieved per iteration for parallel processing. You can set the value of this parameter to 50 to 100 times the **Parallel Worker Threads** property, depending on the memory consumption limit. **Note**: A large value for this property can result in memory exhaustion.
Parallel Locker Thread Sleep (sec)	The idle time between two successive iterations for retrieving rows, when there is no message for parallel processing. The time is measured in seconds.
Error Locker Thread Sleep (sec)	The idle time between two successive iterations for retrieving errored messages when there is no errored message from parallel processing. The time is measured in seconds.You can configure the error/retry infrastructure by setting this property.
Parameters	Custom configuration properties. For an example, see Section 15.2, "Configuring Resequenced Messages."

Table 15–1 (Cont.) Mediator Service Engine Properties

Property	Description
Container ID Refresh Time (sec) **Container ID Lease Timeout (sec)**	Heartbeat infrastructure properties. The heartbeat infrastructure is a part of the Mediator service engine and detects the absence of a Mediator service engine instance due to failure or shutdown of a node. The heartbeat infrastructure creates a unique identifier for each instance of the Mediator service engine and performs the necessary housekeeping tasks, if a Mediator service engine fails. The heartbeat infrastructure consists of a heartbeat thread. The heartbeat thread periodically updates the time stamp associated with each Mediator service engine's unique identifier. By updating the time stamp associated with it, a Mediator service engine announces its presence to the other Mediator service engines. The heartbeat thread also checks if there are unique identifiers that have not been updated for a particular period of time. You can configure the heartbeat framework by setting the following parameters: ■ **Container ID Refresh Time (sec)**: The time interval at which the heartbeat thread periodically updates the time stamp associated with each Mediator service engine's unique identifier. **Notes:** The default value is `60` seconds. In case of unplanned outages, you must wait the amount of time specified for this property after restarting the server. ■ **Container ID Lease Timeout (sec)**: The time interval at which the heartbeat thread periodically checks if there are unique identifiers that have not been updated for a particular period of time. A node is considered to be timed out if it has not refreshed itself for a duration longer than the amount of time specified here. By configuring these parameters, you can specify the period used by the heartbeat thread to detect the failure of a Mediator service engine.
Resequencer Locker Thread Sleep (sec)	The sleep interval for the locker threads in seconds. When the resequencer is unable to find a group with messages that can be processed, the locker thread sleeps for the specified duration. The locker thread does not sleep between each iteration of a database seek, as long as it finds groups with messages that can be processed. The default value is 10.
Resequencer Maximum Groups Locked	The number of groups to be retrieved for processing in a single iteration of a database seek. Once retrieved, the groups are assigned to worker threads for processing. The default value is 4.
Resequencer Worker Threads	The number of worker threads (dispatchers) for processing resequencing groups in parallel. Each worker thread is assigned a group, and then processes messages for the group in sequence. When there are a large number of groups waiting for messages to be processed, increasing this parameter can improve performance.The default value is 4.

3. To configure advanced Mediator properties in the System MBean Browser, click **More Mediator Configuration Properties**.

You can also access Mediator properties in the System MBean Browser through the navigator, as described below.

To access System MBean Browser properties:

1. From the **SOA Infrastructure** menu, point to **Administration** and then select **System MBean Browser**.

The System MBean Browser page appears.

2. In the System MBean Browser navigation pane on the left, expand **oracle.as.soainfra.config**, expand **Server:** ***server_name***, expand **MediatorConfig**, and then select **mediator**.

   The properties of the MBean appear in the right pane.

3. To change the value of a property, modify its **Value** field and then click **Apply**.

> **Note:** Not all values can be modified. The System MBean properties generally correspond to the properties listed inTable 15–1, with some additional read-only properties for the MBean. An additional property, **ContainerInitRetries**, is not listed in Table 15–1. This property indicates the number of attempts that are made to create an Oracle Mediator service engine instance ID.

15.2 Configuring Resequenced Messages

For Mediator service components to resequence messages, you must configure the following:

- The worker thread count
- The maximum number of groups that can be locked by a thread
- The sleep interval

If the Mediator service component is configured to use best effort resequencing and the messages to process in each batch are based on a time window rather than a maximum number of rows, you can also configure the buffer window.

To configure resequenced messages:

1. Access the Mediator Service Engine Properties page or the System MBean Browser using one of the methods described in Section 15.1, "Configuring Oracle Mediator Service Engine Properties."
2. Enter a value for the following Mediator properties:
   - **Resequencer Worker Threads**: The number of threads used by resequencers.
   - **Resequencer Maximum Groups Locked**: The maximum number of group rows retrieved for each locking cycle.
   - **Resequencer Locker Thread Sleep**: The length of time in seconds for the resequencer locker to sleep when there are no messages in the database.

   > **Note:** For more information about these properties, see Table 15–1.

3. To configure the buffer window for the time window in best effort resequencing, enter the following for the **Parameters** property value:

   ```
   buffer.window=x
   ```

   Where *x* is the percentage of the configured time window to add to the buffer. For example, `buffer.window=20` means that 20% of the length of the time window is added as a buffer.

4. Click **Apply**.

16

Monitoring Oracle Mediator Service Components and Engines

This chapter describes how to monitor runtime instance information, component states, and faults for Oracle Mediator service components and engines using Oracle Enterprise Manager Fusion Middleware Control.

This chapter includes the following topics:

- Section 16.1, "Introduction to the Oracle Mediator Dashboard Pages"
- Section 16.2, "Monitoring Oracle Mediator Service Component Instances and Faults"
- Section 16.3, "Monitoring Oracle Mediator Service Engine Instances and Faults"
- Section 16.4, "Monitoring Resequenced Messages"

For more information, see the following sections:

- Section 1.2.4, "Introduction to Service Components and Service Component Instances"
- Section 1.2.6, "Introduction to Service Engines"

16.1 Introduction to the Oracle Mediator Dashboard Pages

The Mediator Component Dashboard lets you view information about the current state of the selected Mediator service component, and to manage faulted instances and messages. The Mediator Engine Dashboard lets you view the same information, but for all Mediator service components running on the Mediator service engine.

Each section of the dashboard includes links to further information that can help you monitor the status of your environment. The following sections describe the instance information in the different sections of the Dashboard page.

Note: When viewing instances for Mediator service components that use message resequencing, additional fields appear on some of the tabbed pages. These fields provide additional information about the resequencing groups.

16.1.1 Recent Instances Section

This section describes the types of information you can view for recent Mediator instances. You can select whether to view all instances or only running instances.

- **Instance ID**: The unique ID of a specific Mediator instance.

- **Component**: The name of the Mediator service component for which the instance was initiated (Mediator Engine Dashboard only).
- **Composite**: The name of the composite application in which the instance was initiated (Mediator Engine Dashboard only).
- **State**: The state of the specific Mediator instance. The possible states are described below.
  - **Stale**: The composite for which this Mediator instance was created is undeployed.
  - **Terminated by User**: The instance was terminated manually through Oracle Enterprise Manager Fusion Middleware Control, or automatically by a fault policy.
  - **Faulted**: The instance is faulted and cannot be recovered.
  - **Completed successfully**: Everything is fine with this instance and it ran successfully.
  - **Recovery required**: The instance is faulted and can be recovered manually through Oracle Enterprise Manager Fusion Middleware Control.

    For more information about recovering a fault, see Section 17.3, "Recovering From Oracle Mediator Service Engine Faults".
  - **Running**: One or more routing rules of the Mediator instance are still running.
- **Start Date**: The date when the Mediator instance was started.
- **Last Modified Date**: The date when the Mediator instance was last modified.
- **Source**: The operation or event that triggered the Mediator instance.
- **Logs**: The location of the log file containing the log messages related to the Mediator instance.

16.1.2 Components Section

This section appears on the Mediator Engine Dashboard only, and displays the following information about the Mediator service components running on the Mediator service engine.

- **Name** : The name of each Mediator service component running on the service engine.
- **Composite**: The name of each composite that includes a Mediator service component running on the service engine.
- **Status**: An indicator of whether the Mediator service component is up or down.
- **Total Instances**: The number of total instances processed or being processed for the service component.
- **Running Instances**: The number of instance currently being processed for the service component.
- **Recoverable Instances**: The number of faulted instances that are recoverable.
- **Non Recoverable Instances**: The number of faulted instances that are not recoverable.

16.1.3 Recent Faults Section

This section provides the following information about the recent faults that occurred while the Mediator service component or service engine was running:

- **Error Message**: The detailed error message associated with the faulted instance.
- **Recovery**: An indicator of whether the fault is recoverable or not. If a fault is marked as recoverable on the Mediator Component Dashboard, you can select it and choose a recovery action from the **Recovery Actions** list. You can also click **Recover** for that fault to access more recovery options at the service component instance level.
- **Fault Time**: The time when the fault occurred in the instance.
- **Component**: The name of the Mediator service component in which the fault occurred (on the Mediator Engine Dashboard only).
- **Component Instance ID**: The unique ID of the Mediator service component instance.
- **Case**: The routing source case where the fault occurred.
- **Logs**: The log file containing the log message related to the fault. Click this link to see more details about the fault and potential causes of the fault.

For more details about the information available on the Faults page, see Section 17.3, "Recovering From Oracle Mediator Service Engine Faults".

16.1.4 Routing Statistics Section

This section is on the Mediator Component Dashboard only, and provides the following information about the routing data of a source operation or subscribed event.

- **Number of Successfully Processed Messages**: The number of messages successfully processed for the selected route source.
- **Number of Faulted Messages**: The number of messages that threw faults for the selected route source.
- **Number of Incoming Messages**: The total number of incoming messages for the selected route source.
- **Average Processing Time for Successful Messages**: The average time taken to process each successful message for the selected route source.
- **Average Processing Time for Faulted Massages** -The average time taken to process each faulted message.

The **Route Target** subsection in the **Routing Statistics** section displays statistics of the target routes for the Mediator service component. This section provides the following information about an Mediator service component instance:

- **Name**: The name of the route target of the Mediator service component.
- **Error**: The number of errors that occurred during routing.
- **Average Processing Time**: The average processing time for the instances of the Mediator service component. This field has two subfields, **Success** and **Failure**. The Success subfield shows the average processing time for the instances of the Mediator service component that were processed successfully. The Failure subfield shows the average processing time for the instances of the specific Mediator service component that failed to process successfully.

- **Average Invocation Time**: The average invocation time for the instances of the Mediator service component.

16.1.5 Instance Rate Per Min Section

This section provides information about the execution rate of the Mediator instances per minute. This section displays a graph that shows real-time data for successful, faulted, and incoming instances in the last five minutes.

You can view the instance rate for the last five minutes in tabular form by clicking **Table View**.

16.2 Monitoring Oracle Mediator Service Component Instances and Faults

You can perform any of the following component-level monitoring tasks from the Mediator Component home page on Oracle Enterprise Manager Fusion Middleware Control:

- Monitoring Oracle Mediator Service Component Recent Instances and Faults
- Monitoring Oracle Mediator Service Component Instances
- Monitoring Oracle Mediator Service Component Faults
- Monitoring Oracle Mediator Routing Statistics
- Monitoring Audit Trail and Fault Details for an Oracle Mediator Component Instance

16.2.1 Monitoring Oracle Mediator Service Component Recent Instances and Faults

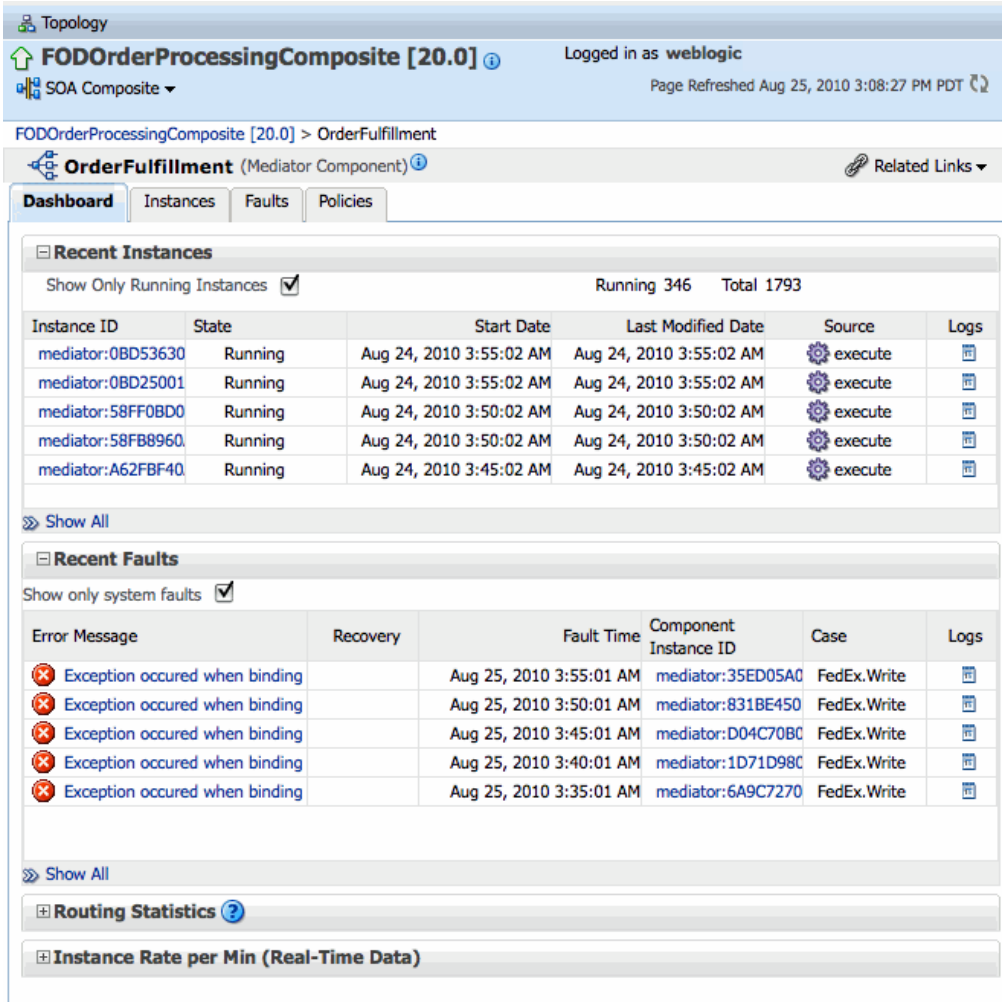
The Mediator Component Dashboard displays information about recent instances and faults. Each service component in a SOA composite application has its own instance ID. These IDs are different from the overall instance ID of the SOA composite application of which each service component is a part.

To monitor Mediator service component instances and faults:

1. Access the SOA Composite home Page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Expand **soa-infra**, and then select a specific SOA composite application.

2. In the **Component Metrics** section, select the Mediator service component.
3. Click the **Dashboard** tab.



The Dashboard page displays the information described in Section 16.1, "Introduction to the Oracle Mediator Dashboard Pages."

4. In the **Recent Instances** section, perform any of the following tasks:

   a. In the **Instance ID** column, click an instance ID to view its instance details, including an audit trail and faults.

   b. In the **Logs** column, click a log icon to access the Log Messages page with messages specific to that instance.

   c. Click **Show Only Running Instances** to view only those instances that are in the running state.

   d. Click **Show All** below the section to access the Instances page of the service component and view all instances of the service component.

5. In the **Recent Faults** section, perform any of the following tasks:

   a. In the **Error Message** column, click an error message to display complete information about the fault. If the fault is identified as recoverable, click **Recover Now** to perform fault recovery.

   b. In the **Recovery** column, click a fault identified as **Recoverable** to perform fault recovery at the service component instance level.

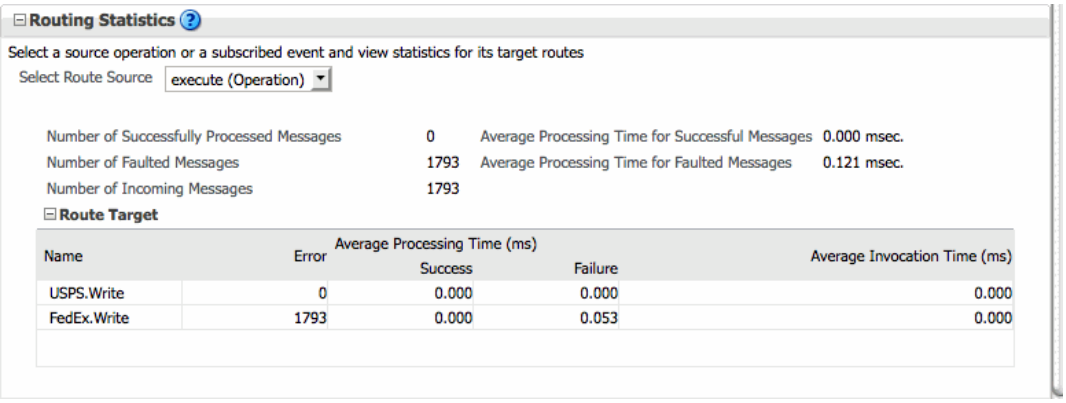
c. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

d. Click **Show only system faults** to display only the faults that were caused by a system error. System faults are related to system failure issues, such as a database or network being inaccessible.

e. Click **Show All** below the section to access the Faults page of the service component and view all component faults.

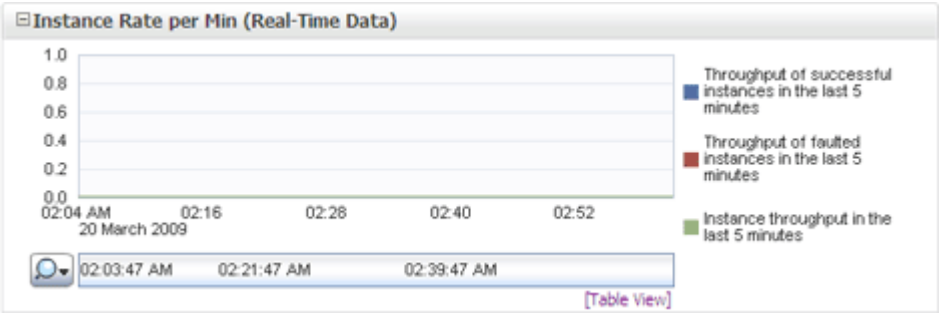
6. Expand the **Routing Statistics** section and select an operation or event in the **Select Route Source** field.

   View the statistics for the selected operation or event. To view additional information, expand the **Route Target** section.



7. Expand the **Instance Rate per Min (Real-Time Data)** section to view a graphical representation of the number of successful, faulted, and incoming (pending) instances of the Mediator service component over a specific time range.

   Click **Table View** to display the same information in a tabular format.



For more information, see Section 1.2.3, "Introduction to SOA Composite Application Instances". For information about viewing and searching log files, see *Oracle Fusion Middleware Administrator's Guide*.

16.2.2 Monitoring Oracle Mediator Service Component Instances

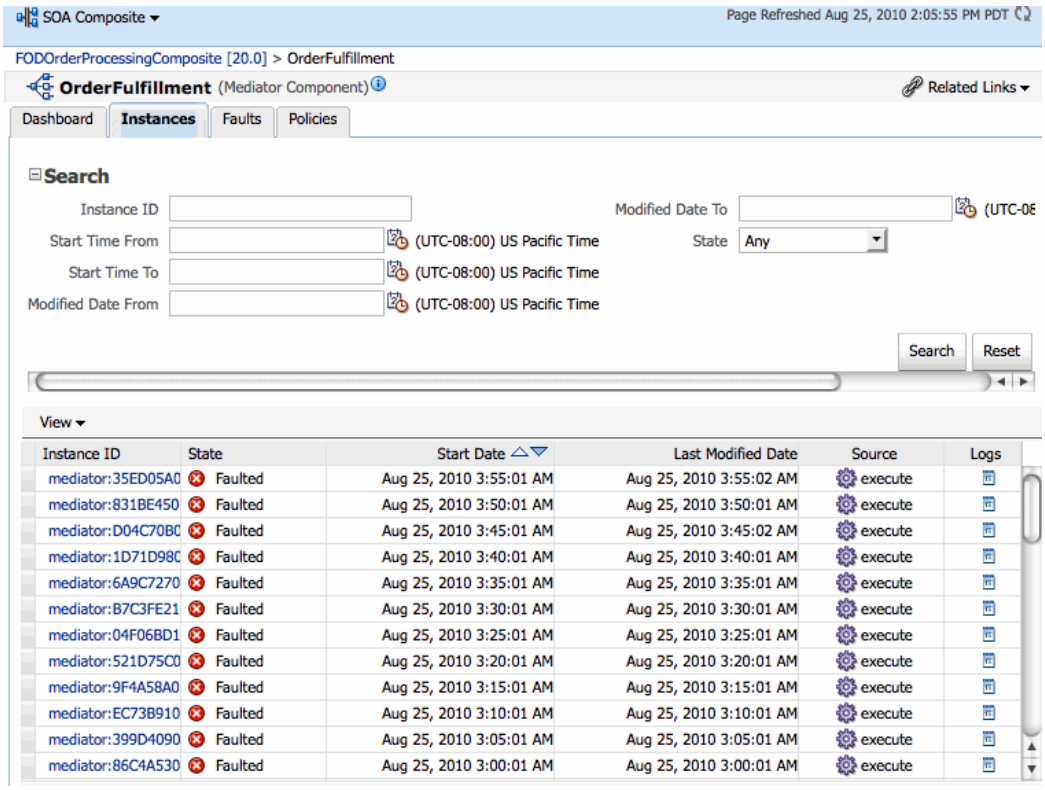
The Instances page on the Mediator Component home page lets you search for a Mediator service component instance and view information about an instance based on the criteria specified.

To monitor the instance statistics of a Mediator service component:

1. Access the SOA Composite home page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Expand **soa-infra**, and then select a specific SOA composite application.

2. In the **Component Metrics** section, select the Mediator service component.
3. Click the **Instances** tab.



4. To perform a search, enter criteria into any of the available fields, and then click **Search**. For the date and time fields, you can click the calendar icon next to the field to select a date or time.

   For information about the fields displayed on the Instances page, see Section 16.1.1, "Recent Instances Section."

16.2.3 Monitoring Oracle Mediator Service Component Faults

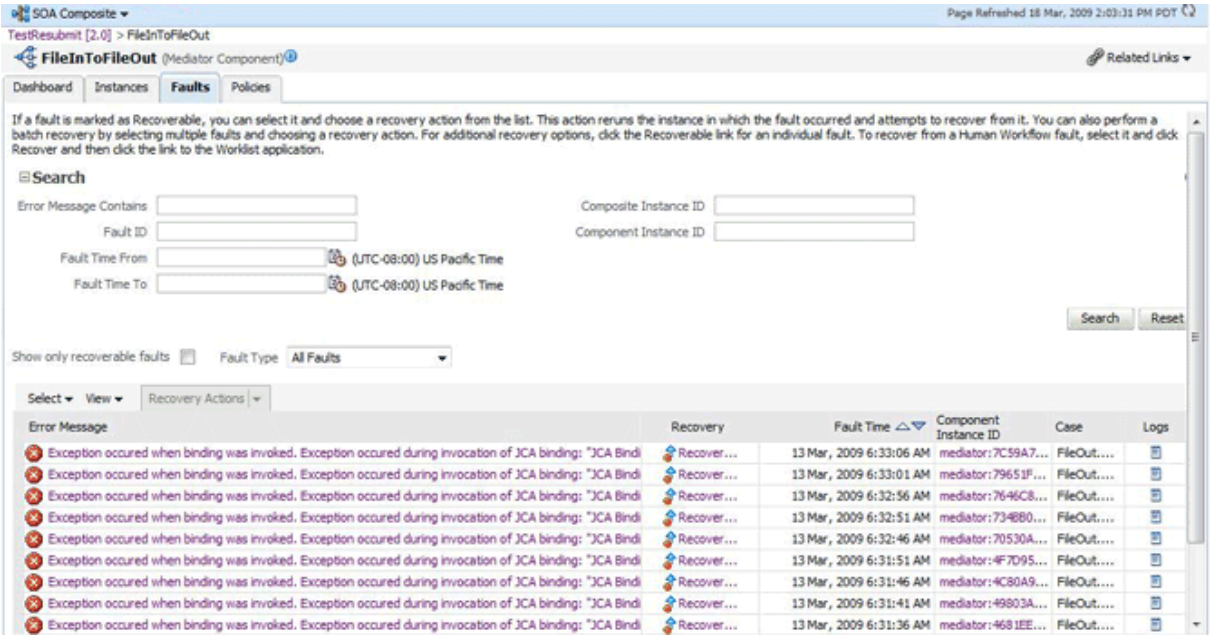
The Faults page of the Mediator Component home page lets you search for faults based on the specified criteria, to view information about faulted instances, and to recover or terminate multiple faults.

To monitor the faults for a Mediator service component:

1. Access the SOA Composite home page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Expand **soa-infra**, and then select a specific SOA composite application.

2. In the **Component Metrics** section, select the Mediator service component.
3. Click the **Faults** tab.



4. To perform a search, enter criteria into any of the available fields, and then click **Search**. For the date and time fields, you can click the calendar icon next to the field to select a date or time.

   For information about the fields displayed on the Faults page, see the Section 16.1.3, "Recent Faults Section." For information on handling faults, see Section 17.1, "Recovering From Oracle Mediator Service Component Faults."

16.2.4 Monitoring Oracle Mediator Routing Statistics

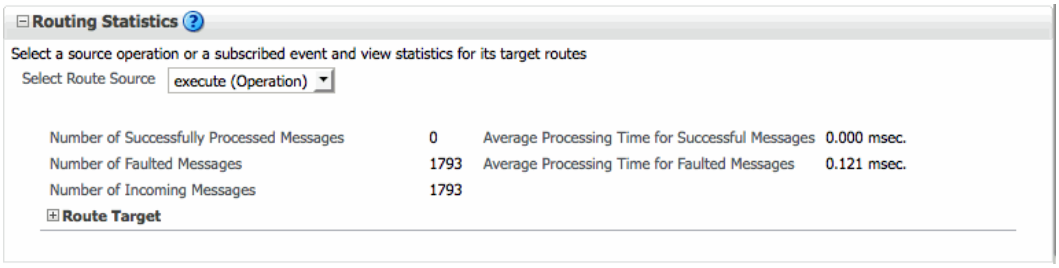
The **Routing Statistics** section of the Dashboard page in the Mediator Component home page displays the routing data of a source operation or subscribed event.

To monitor the routing statistics of a Mediator service component:

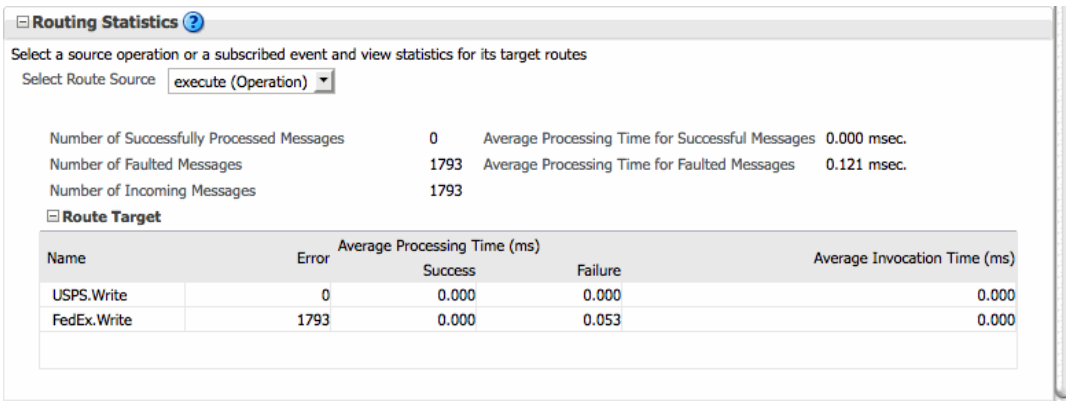
1. Access the SOA Composite home page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Expand **soa-infra**, and then select a specific SOA composite application.

2. In the **Component Metrics** section, select the Mediator service component.
3. Click the **Dashboard** tab.
4. In the **Routing Statistics** section, select a routing source from the **Select Route Source** list.



5. Expand the **Route Target** table.



6. View the routing statistics for all targets in the **Route Target** table.

   For information about the fields displayed in the **Routing Statistics** section, see Section 16.1.4, "Routing Statistics Section."

16.2.5 Monitoring Audit Trail and Fault Details for an Oracle Mediator Component Instance

You can click the links on the Mediator Component Dashboard, Instances, and Faults pages to view audit trail and fault information for the instance. You can also access this information from the Mediator Engine Dashboard, Instances, and Faults pages.

To view audit trail and fault details:

1. Access the SOA Composite home page through one of the following options:

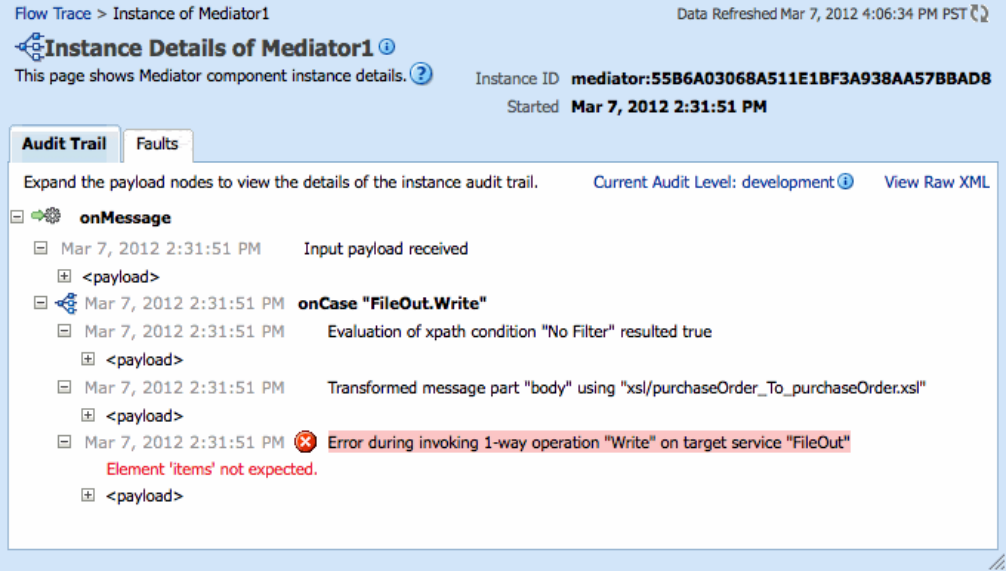
From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Expand **soa-infra**, and then select a specific SOA composite application.

2. In the **Component Metrics** section, select the Mediator service component.
3. Click the **Dashboard** tab, or click the **Instances** or **Faults** tab and perform a search.

4. Select the instance ID of the instance you want to view. On the Dashboard tab, you can click the instance ID from the Recent Instances or the Recent Faults section.

   The Mediator Instance Details dialog appears with the audit trail displayed.

Figure 16–1 Mediator Component Instance Audit Trail



5. To view a list of faults, click the **Faults** tab.
6. To view detailed information about a specific fault, click the fault in the Message column.

Figure 16–2 Mediator Component Instance Fault Details

16.3 Monitoring Oracle Mediator Service Engine Instances and Faults

You can perform the following monitoring tasks at the Mediator service engine level from the Dashboard tab of the Mediator Engine home page in Oracle Enterprise Manager Middleware Control.

- Monitoring Oracle Mediator Service Engine Recent Instances and Faults
- Monitoring Oracle Mediator Service Engine Instances
- Monitoring Request Breakdown Statistics
- Monitoring Deployed Oracle Mediator Service Components in the Service Engine

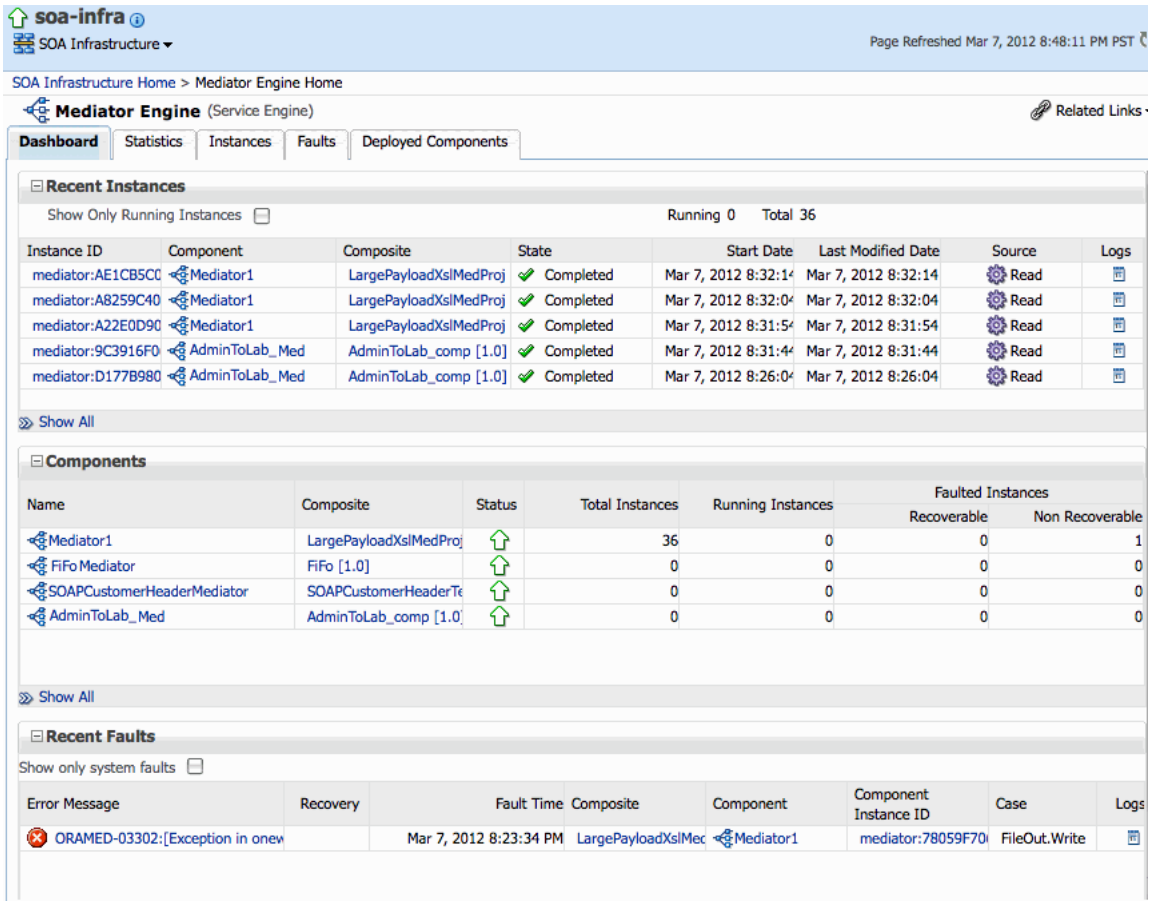
16.3.1 Monitoring Oracle Mediator Service Engine Recent Instances and Faults

To monitor Mediator service engine instances and faults:

1. Access the Mediator Engine home page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** and then select **Mediator**.	1. Right-click **soainfra**, point to **Service Engines** and then select **Mediator**.

2. Click the **Dashboard** tab.



The Dashboard page displays the information described in Section 16.1, "Introduction to the Oracle Mediator Dashboard Pages."

3. In the **Recent Instances** section, perform any of the following tasks:

   a. In the **Instance ID** column, click an instance ID to view its instance details, including an audit trail and faults.

b. In the **Component** column, click a service component name to view the home page for that component.

c. In the **Composite** column, click a composite name to view the home page for the SOA composite.

d. In the **Logs** column, click a log icon to access the Log Messages page with messages specific to that instance.

e. Click **Show Only Running Instances** to view only those instances that are in the running state.

f. Click **Show All** below the section to access the Instances page of the service engine and view all instances of the service engine.

4. In the **Components** section, perform any of the following tasks:

   a. In the **Name** column, click a service component name to view the Mediator Component Dashboard for that service component.

   b. In the **Composite** column, click the name of a SOA composite to view the home page for that composite.

   c. Click **Show All** below the section to access the Deployed Components page of the service engine and view all deployed Mediator components.

5. In the **Recent Faults** section, perform any of the following tasks:

   a. In the **Error Message** column, click an error message to display complete information about the fault. If the fault is identified as recoverable, click **Recover Now** to perform fault recovery.

   b. In the **Recovery** column, click a fault identified as **Recoverable** to perform fault recovery at the service engine level.

   c. In the **Composite** column, click a composite name to view the home page for the SOA composite.

   d. In the **Component** column, click a service component name to view the home page for that component.

   e. In the **Component Instance ID** column, click an instance ID to view its instance details, including an audit trail and faults.

   f. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

   g. Click **Show only system faults** to display only the faults that were caused by a system error. System faults are related to system failure issues, such as a database or network being inaccessible.

   h. Click **Show All** below the section to access the Faults page of the service engine and view all service engine faults.

For more information, see Section 1.2.3, "Introduction to SOA Composite Application Instances". For information about viewing and searching log files, see *Oracle Fusion Middleware Administrator's Guide*.

16.3.2 Monitoring Oracle Mediator Service Engine Instances

You can monitor all Mediator service component instances running in the service engine from the Mediator Engine home page. These service components can be part of several SOA composite applications.

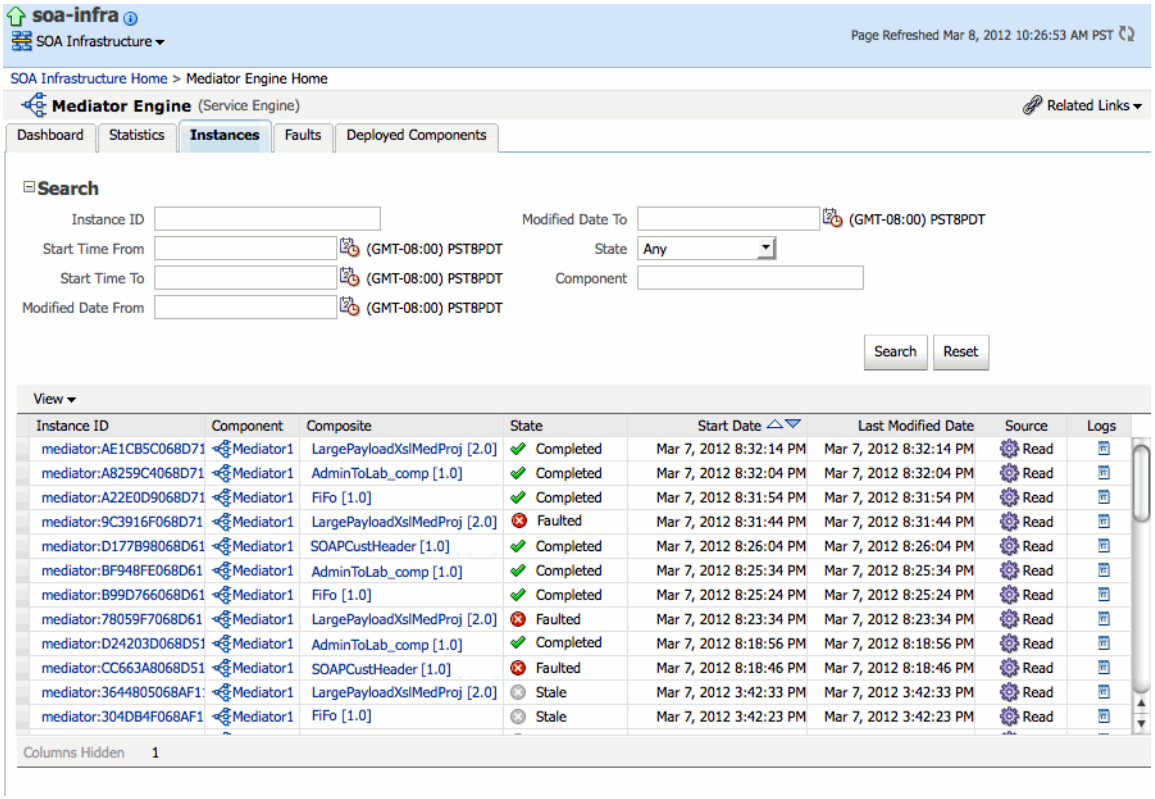
To monitor Mediator service engine instances:

1. Access the Mediator Engine home page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **Mediator**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **Mediator**.

2. Click the **Instances** tab.

   The Instances page includes two sections. The upper section lets you search for the instances to view and the lower section displays information about the instances returned by the search.



> **Note:** Undeploying a SOA composite application instance does not cause the instance to complete. Undeployment marks running service component instances of that composite as stale. However, service component instances of that composite instance that are complete are not marked as stale.

3. In the **Search** section, enter any of the search criteria and click **Search**. To clear the search fields, click **Reset**.
4. In the **Instances** section, perform any of the following monitoring tasks:
   a. In the **Instance ID** column, click an instance ID for a service component to view its audit trail, process flow, and faults.
   b. In the **Component** column, click a specific service component to access its home page.

c. In the **Composite** column, click a specific SOA composite application to access its home page.

d. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

For more information, see Section 1.2.4, "Introduction to Service Components and Service Component Instances."

16.3.3 Monitoring Request Breakdown Statistics

You can assess the efficiency level of the Mediator service engine by monitoring the request breakdown statistics.

To monitor the request breakdown statistics of the currently deployed Mediator service components:

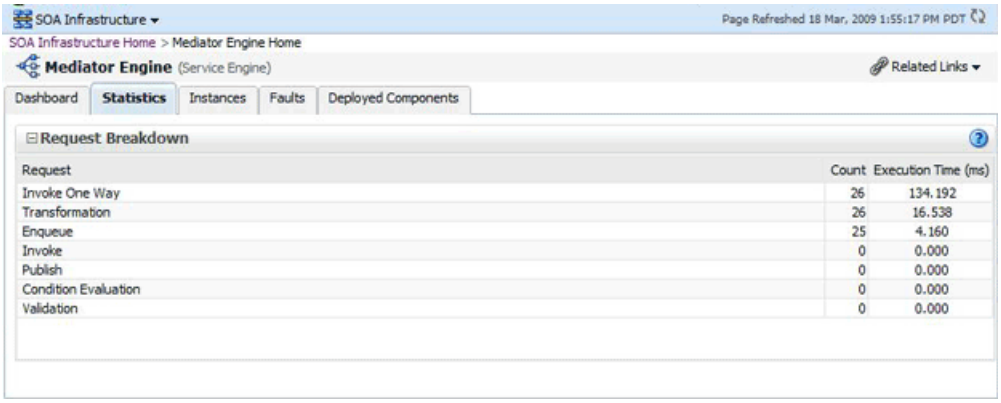
1. Access the Mediator Engine home page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **Mediator**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **Mediator**.

   The Mediator Engine home page appears.

2. Click the **Statistics** tab.

3. View the request breakdown statistics.



   The **Request Breakdown** section provides information about the count and the average time taken for processing the following actions:

   - **Invoke One Way**: One-way invocations from the Mediator service engine.
   - **Transformation**: Transforming messages in the Mediator service engine.
   - **Enqueue**: Dehydrating messages for parallel routing rules.

     Note: Dehydrating of messages means storing the incoming messages in the database for parallel routing rules for processing later by worker threads.

   - **Invoke**: Request-response invocations from the Mediator Service Engine.

- **Publish**: Publishing events from the Mediator service engine.
- **Condition Evaluation**: Filtering conditions for evaluation by the Mediator service engine.
- **Validation**: Message validations by the Mediator service engine.

16.3.4 Monitoring Deployed Oracle Mediator Service Components in the Service Engine

You can monitor all deployed SOA composite applications with Mediator service components running in the service engine.

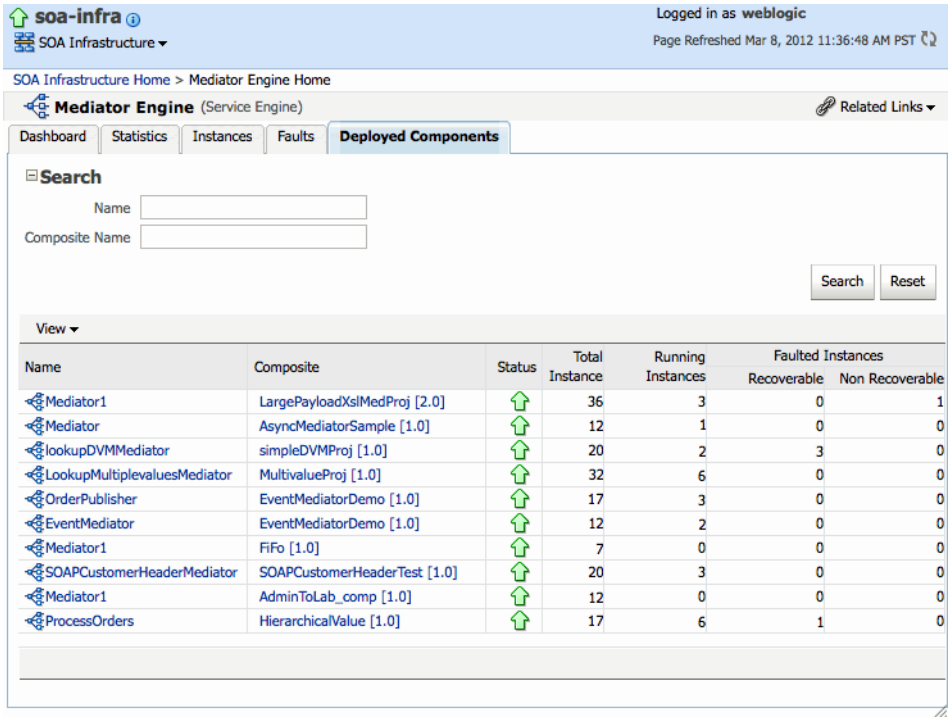
To monitor deployed Mediator components in service engines:

1. Access the Mediator Engine home page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **Mediator**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **Mediator**.

2. Click the **Deployed Components** tab.

   The Deployed Components page includes two sections. The upper section lets you search for the components to view and the lower section displays information about the components returned by the search.



3. In the **Name** column, click a specific service component to access its home page.
4. In the **Composite** column, click a specific SOA composite application to access its home page.

16.4 Monitoring Resequenced Messages

You can monitor the resequenced messages from the Mediator Component home page and the Mediator Instance Dialog. The following topics describe the monitoring processes:

- Section 16.4.1, "Monitoring Resequenced Messages from the Mediator Service Component Home Page"
- Section 16.4.2, "Monitoring Resequenced Messages from the Mediator Instance Dialog"

16.4.1 Monitoring Resequenced Messages from the Mediator Service Component Home Page

On the Mediator Component home page, the Dashboard, Instances, and Faults pages include additional fields for monitoring components that use the resequencing feature. These additional fields only appear when the Mediator service component you are monitoring includes a resequencer.

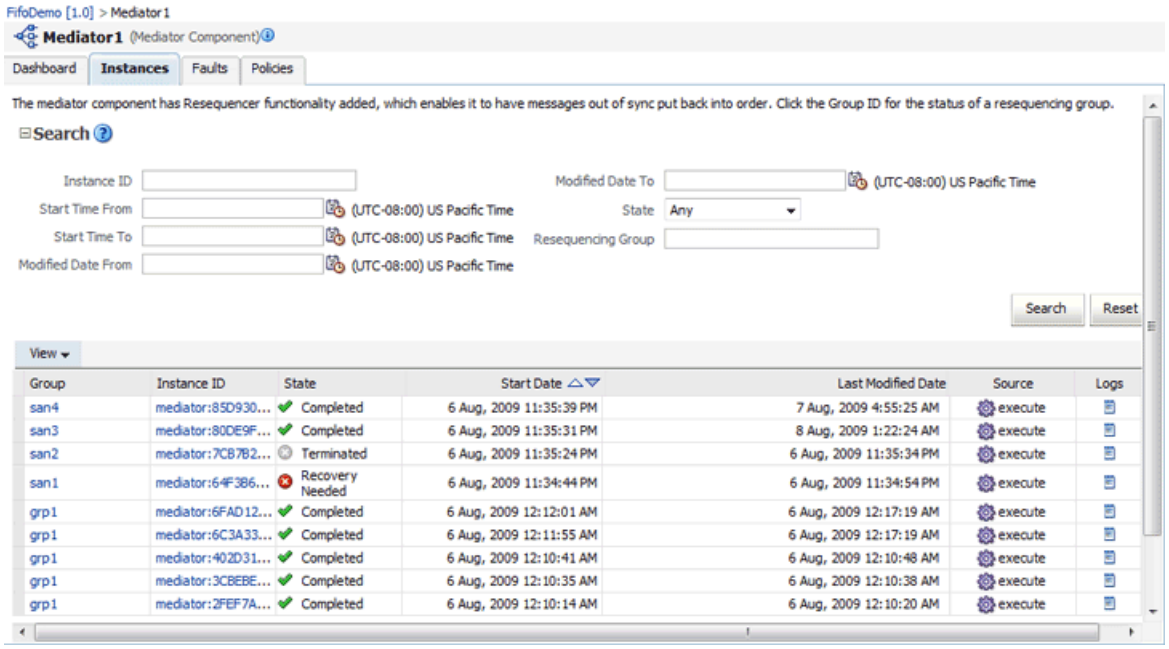
16.4.1.1 Dashboard Page

The Dashboard page of the Mediator Component home page has one additional column, **Group**, in the **Recent Instances** and **Recent Faults** tables. This field contains the group ID of the resequenced messages. If you click the group ID for a row, additional information about the group appears. For more information, see Section 16.4.1.4, "Mediator Resequencing Group Dialog." For more information about the Dashboard page, see Section 16.1, "Introduction to the Oracle Mediator Dashboard Pages."

16.4.1.2 Instances Page

As shown in Figure 16–3, the Instances page of the Mediator Component home page provides an additional search field, **Resequencing Group**, so you can search for a Mediator component instance based on group criteria. The **Instances** section, which displays information about Mediator service component instances that match the search criteria, has one additional column, **Group**. This column contains the group ID of the resequenced messages. If you click the group ID for a row, additional information about the group appears. For more information, see Section 16.4.1.4, "Mediator Resequencing Group Dialog." For more information about the Instances page, see Section 16.2.2, "Monitoring Oracle Mediator Service Component Instances."

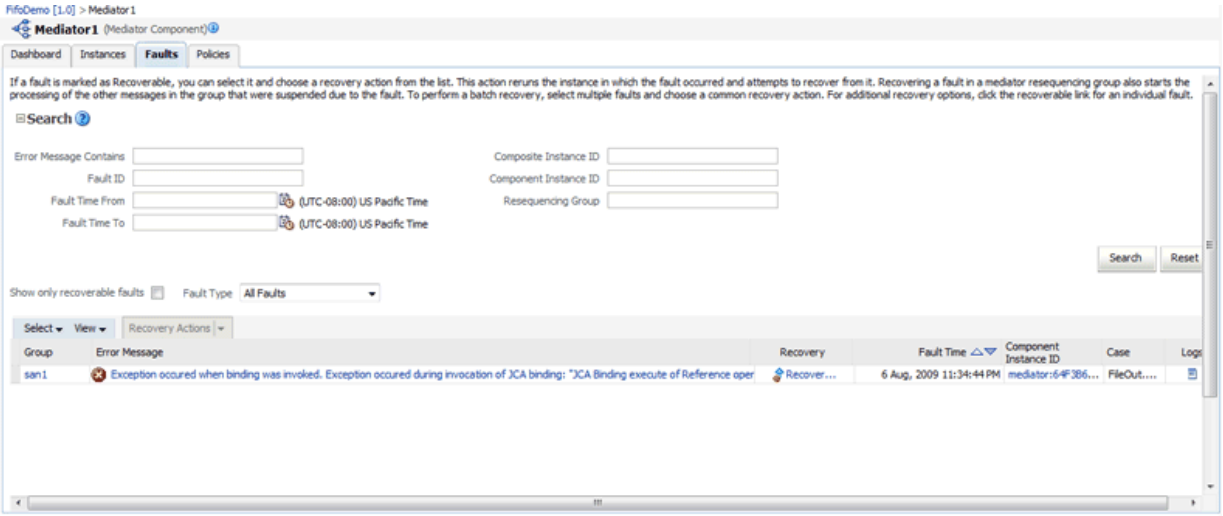
Figure 16–3 Instances Page for Resequenced Messages



16.4.1.3 Faults Page

As shown in Figure 16–4, the Faults page of the Mediator Component home page provides an additional search field, **Resequencing Group**, so you can search for Mediator service component instances based on group criteria. The **Faults** section, which displays information about Mediator component instances that match the search criteria, provides one additional column, **Group**. This column contains the group ID of the resequenced messages. If you click the group ID for a row, additional information about the group appears. For more information, see Section 16.4.1.4, "Mediator Resequencing Group Dialog." For more information about the Faults page, see Section 17.3, "Recovering From Oracle Mediator Service Engine Faults."

Figure 16–4 Faults Page for Resequenced Messages



16.4.1.4 Mediator Resequencing Group Dialog

If all the messages in a group are processed without any fault or time out interval, the Mediator Resequencing Group dialog displays a message indicating that the group is working. Figure 16–5 illustrates the dialog.

Figure 16–5 Mediator Resequencing Group Dialog

The Mediator Resequencing Group dialog provides the following information about a group:

- Whether the group is timed-out or faulted
- The blocking message in the group
- The next message to be processed after the group is unlocked
- The time after which the processing of the messages in the group stopped
- The instruction text to unlock the group

When processing of messages in a group is suspended due to a fault or a timeout, the Mediator Resequencing Group dialog provides information about the suspended group. The processing of messages in a group can be suspended for the following reasons:

- Group Is Faulted
- Group Is Timed-out
- Group Is Faulted Due to a System Error

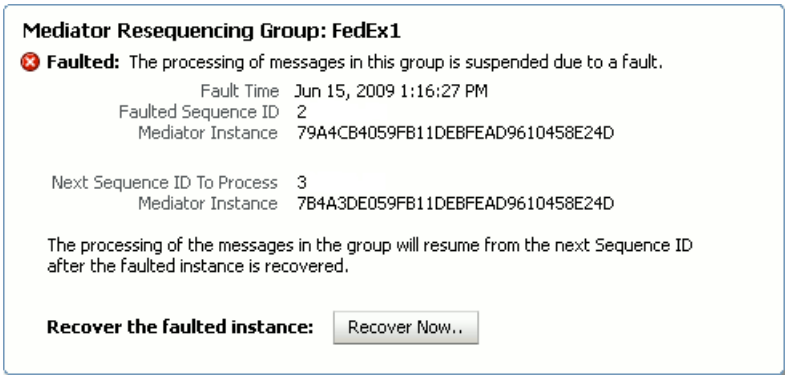
Group Is Faulted

A group is in the faulted state when one of its messages throws an error while it is processed, as shown in Figure 16–6. The following information is displayed for a faulted group:

- The sequence ID of the faulted message
- The sequence ID of the next message to processed, along with its Mediator Instance ID

Click **Recover Now** to access the Faults page of the Mediator Instance Details page to perform single-fault recovery.

Figure 16–6 Group Is Faulted



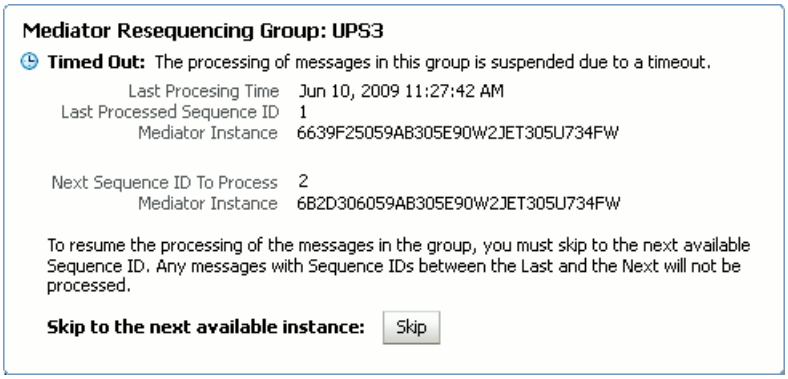
Group Is Timed-out

A group is in the timed-out state when processing of the group stops while waiting for an expected message, as shown in Figure 16–7. The following information is displayed for a timed-out group:

- The sequence ID of the last processed message
- The sequence ID of the next message to be processed, along with its Mediator Instance ID

Click **Skip** to unlock the group and start processing the next available instances in the group.

Figure 16–7 Group Is Timed Out



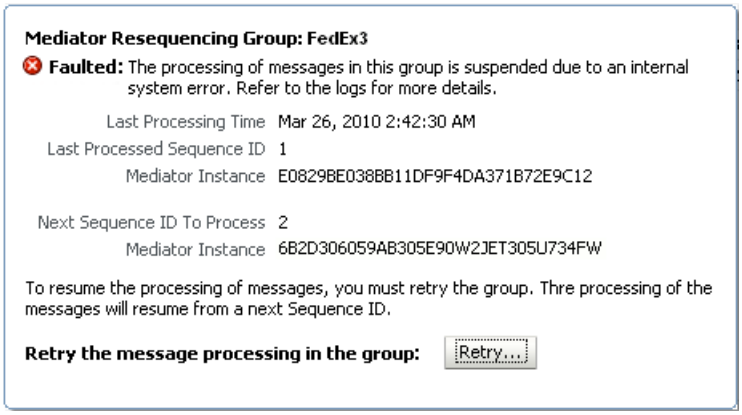
Group Is Faulted Due to a System Error

A group can be in a faulted state when processing of the group is suspended due to a system error, as shown in Figure 16–8. This is a special case for a Faulted Group; the group is not timed out and there is no faulted message for the group. The groups in this state can be described as Errored Groups. The following information is displayed for an Errored Group:

- The sequence ID of the last processed message
- The sequence ID of the next message to be processed, along with its Mediator Instance ID

Click **Retry** to unlock the group and start processing the next instances in the group.

Figure 16–8 Group Is Faulted Due to a System Error



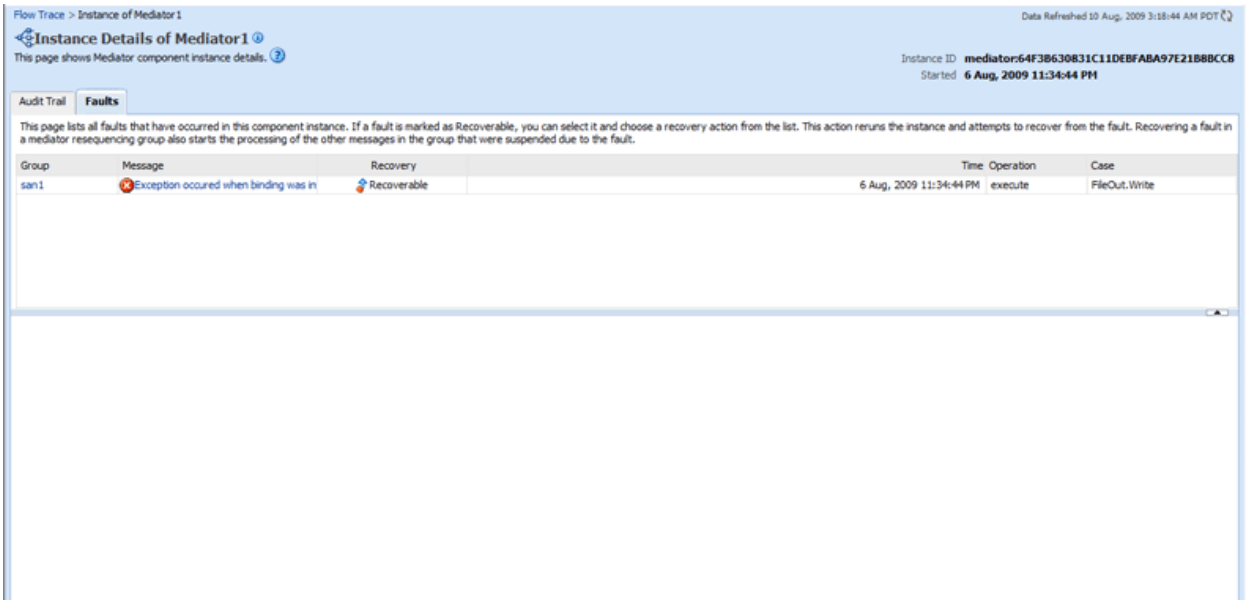
16.4.2 Monitoring Resequenced Messages from the Mediator Instance Dialog

The Faults page of the Mediator Instance Details dialog provides the following resequencer-specific information:

- Group ID of the resequenced message
- Operation performed on the resequenced message (Figure 16–9)

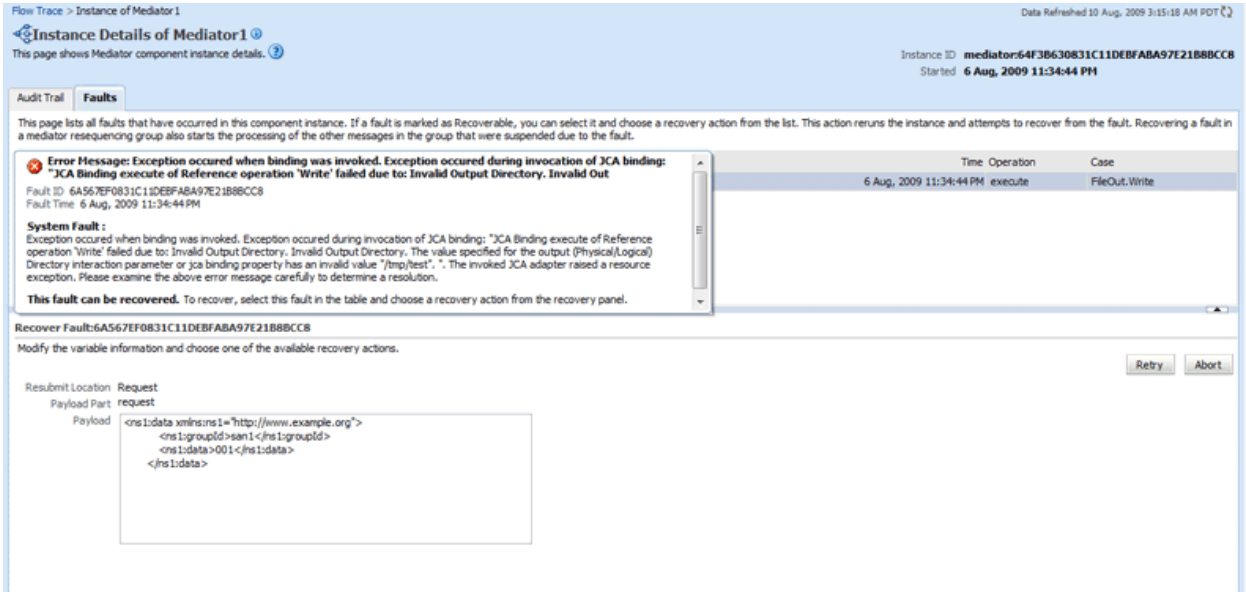
  For more information about the Mediator Instance Details dialog, see Figure 16.2.5.

Figure 16–9 Resequenced Message Faults on the Mediator Instance Details Dialog



If you click a fault message, more information about the fault appears, as shown in Figure 16–10. If you click **Retry**, the Mediator service engine tries to reprocess the message. If you click **Abort**, the Mediator service engine terminates the faulted message, unlocks the group, and then resumes processing from the next message in the sequence.

Figure 16–10 Fault Message Details for Resequenced Messages



17

Managing Oracle Mediator Service Components and Engines

This chapter describes how to manage Oracle Mediator service components and engines using Oracle Enterprise Manager Fusion Middleware Control. It provides instructions for viewing recovering faults, and attaching and detaching Mediator policies.

This chapter includes the following topics:

- Section 17.1, "Recovering From Oracle Mediator Service Component Faults"
- Section 17.2, "Managing Oracle Mediator Policies"
- Section 17.3, "Recovering From Oracle Mediator Service Engine Faults"

17.1 Recovering From Oracle Mediator Service Component Faults

You can monitor Mediator service component faults and perform individual and bulk recoveries for faults that are identified as recoverable.

To manage Mediator service component faults:

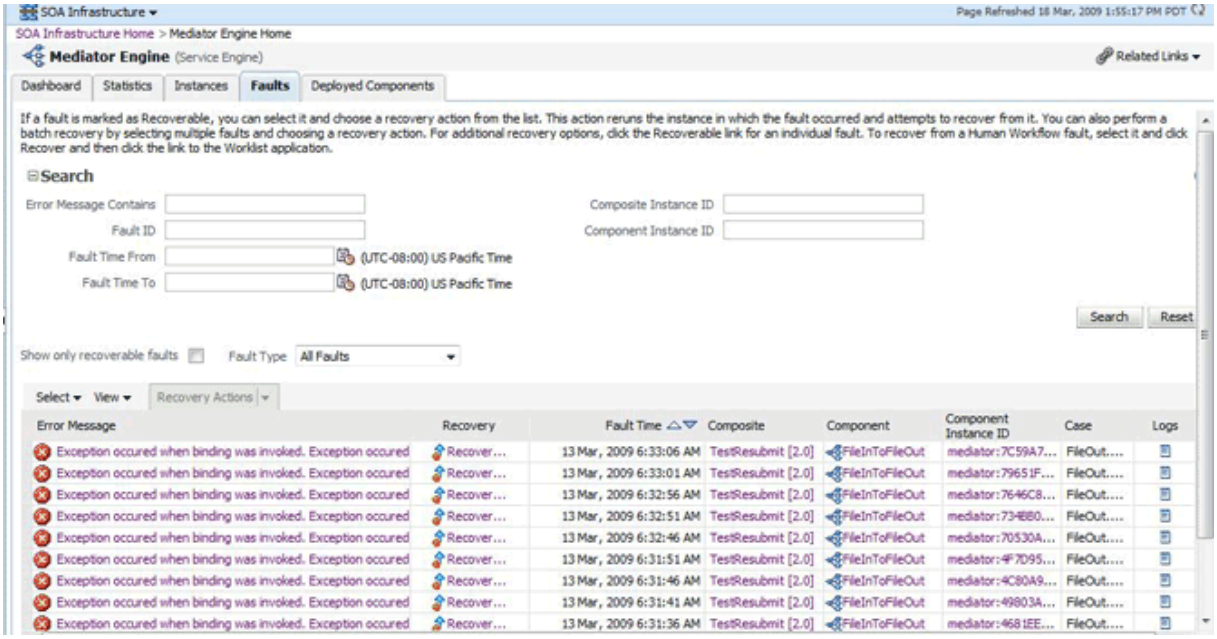
1. Access the SOA Component home page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Expand **soa-infra**, and then select a specific SOA composite application.

2. Select the Mediator service component in the **Component Metrics** section.
3. Click **Faults**.

   The Faults page displays the following information:

   - A utility that you can use to search for a specific fault by specifying criteria and clicking **Search**. Click the **Help** icon for more information.
   - Faults that occurred in the service component, including the error message, whether you can recover from the fault, the time at which the fault occurred, service component instance ID, activity in which the fault occurred, and a link to a log file describing the fault.



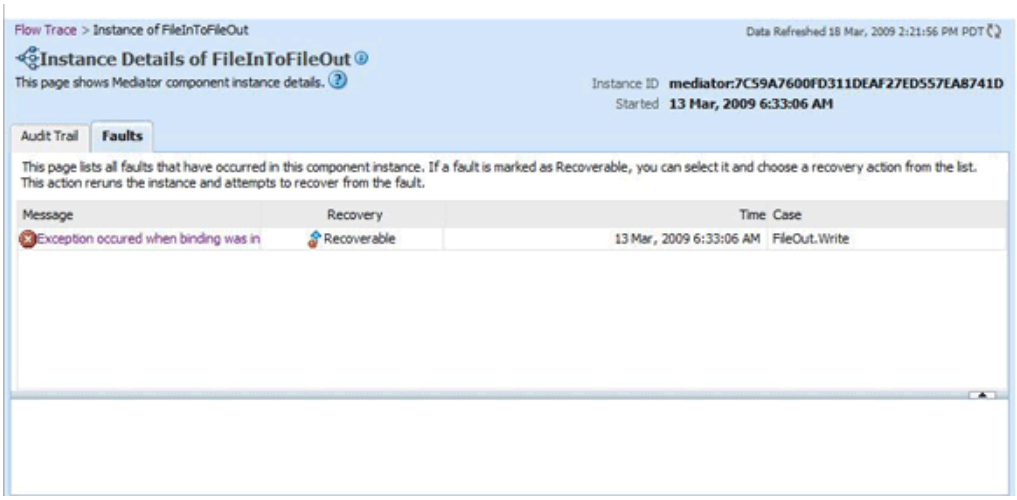
Mediator service component faults that are marked as recoverable can be recovered.

4. Select faults for recovery using one of the following methods.

For...	Then...
Single fault recovery	There are three options from which to choose for single-fault recovery. Perform any of the following actions to recover a single fault. ▪ Click the row of the fault that has been identified as recoverable. With the row highlighted, select a specific action from the **Recovery Action** list, as described in Step 5. ▪ In the **Recovery** column, click the **Recover** link to access the Faults page of the instance audit trail to perform fault recovery. Figure 17–1 illustrates the fault page. ▪ In the **Error Message** column, click the message of a fault that has been identified as recoverable. This displays complete fault details, including the fault ID, fault time, fault location, fault type, and error message text. Click **Recover Now** to access the Faults page of the instance audit trail to perform fault recovery. If there is no **Recover Now** option, the fault is not recoverable.
Bulk fault recovery	There are two options from which to choose for bulk-fault recovery. Perform the following steps to recover multiple faults. 1. Do one of the following: ▪ Use Shift+Click or Control+Click to select specific faults in the rows. ▪ From the **Select** menu, choose **Select All Recoverable**. Then use Shift+Click or Control+Click to deselect the faults to *not* include in the recovery operation. 2. Select an action from the **Recovery Action** list, as described in Step 5. **Note:** Only the actions applicable to all selected faults are available.

For...	Then...
Recovery of all faults	Perform the following steps to recover all faults. 1. From the **Select** menu, choose **Select All Recoverable**. 2. Select an action from the **Recovery Action** list, as described in Step 5. **Note:** Only the actions applicable to all selected faults are available.

Figure 17–1 Fault Tab on the Mediator Instance Detail Dialog



5. If you did not already select a recovery method in the above step, select one of the following actions from the **Recovery Action** list.

Action	Description
Retry	Retries the instance directly. An example of a scenario in which to use this recovery action is when the fault occurred because the service provider was not reachable due to a network error. The network error is now resolved.
Abort	Terminates the entire instance.
Replay	Replays the entire scope activity again in which the fault occurred.
Rethrow	Rethrows the current fault. BPEL fault handlers (catch branches) are used to handle the fault. By default, all exceptions are caught by the fault management framework unless an explicit rethrow fault policy is provided.
Continue	Ignores the fault and continues processing (marks the faulted activity as a success).

For more information, see the following documentation:

- Section 1.4.3.1, "Introduction to Fault Recovery"
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*
- *Oracle Fusion Middleware Administrator's Guide* for details about viewing and searching log files

17.2 Managing Oracle Mediator Policies

Oracle Fusion Middleware uses a policy-based model to manage web services. Policies apply behavioral requirements and security to the delivery of messages. You can view, attach, and detach policies for Mediator service components in currently deployed SOA composite applications.

The Policies page of the Mediator Component Home page provides the following information about a Mediator component instance:

- **Policy Name**: The name of the policy.
- **Policy Reference Status**: A toggle button that allows you to enable or disable a policy. Disabling a policy temporarily turns it off without detaching it. If the button says **Disable**, the policy is enabled; if it says **Enable**, the policy is disabled.
- **Category**: The category of the policy. It has the following values: Management, Reliable Messaging, MTOM Attachments, Security, and WS-Addressing.
- **Total Violations**: The total number of violations since the SOA Infrastructure was restarted.
- **Security Violation**: The number of violations in each category. Category can have the following values: Authentication, Authorization, Confidentiality, and Integrity.

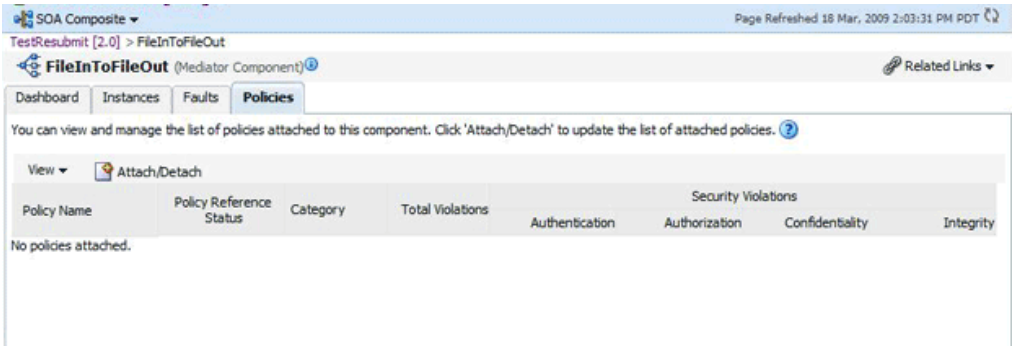
> **Note:** Before attaching policies, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for definitions of available policies and details about which ones to use in your environment.

To manage Mediator policies:

1. Access the SOA Composite home page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, select a specific SOA composite application.

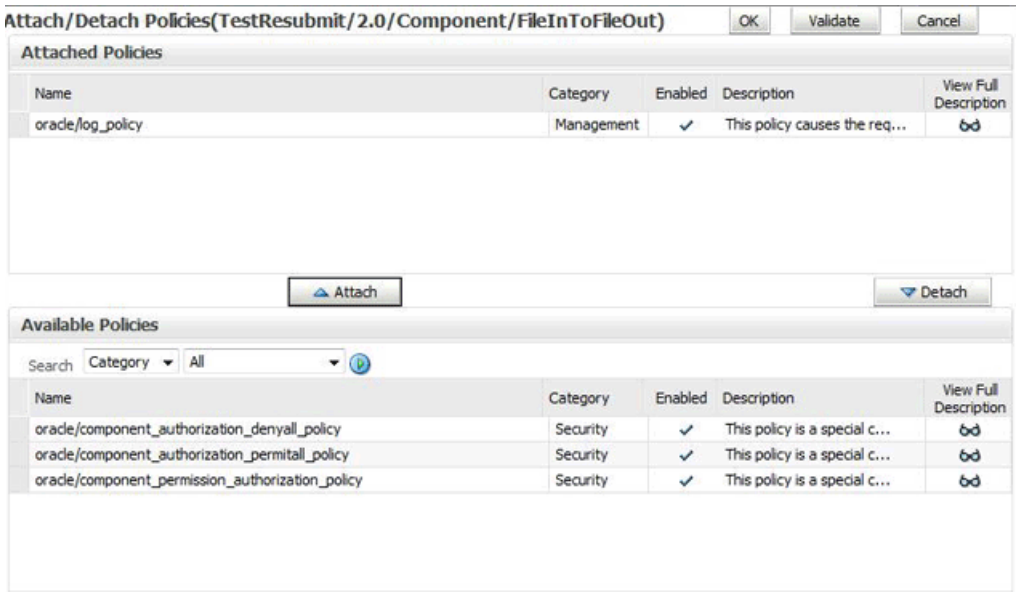
2. Click the **Policies** tab.
3. View the policies attached to a Mediator component.



4. Click **Attach/Detach** to attach or detach a policy.

5. If multiple components are available, select the service or component to which you want to attach the policy.

   The Attach/Detach Policies page appears with all attached policies displayed in the upper pane and all available policies displayed in the lower pane.



6. To attach a policy, select the policy in the lower pane and then click **Attach**.
7. To detach a policy, select the policy in the upper pane and then click **Detach**.
8. When you finish attaching or detaching policies, click **Validate**.
9. If any validation errors occur, make the necessary corrections and run the validation until no more errors occur.
10. Click **OK**.

    The dialog closes and the attached policies appear in the policies table.

For more information, see the following documentation:

- Section 1.4.3.2, "Introduction to Policies"
- Section 7.6, "Managing SOA Composite Application Policies" for the dialogs that are displayed during policy attachment
- *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for definitions of available policies and details about which ones to use for your environment

17.3 Recovering From Oracle Mediator Service Engine Faults

You can monitor and perform individual and bulk fault recoveries for Mediator service engines that are identified as recoverable. All Mediator service component faults, regardless of the SOA composite application instance of which they are a part, can be viewed in the Mediator service engine.

To manage Mediator service engine faults:

1. Access the SOA Composite home page through one of the following options:

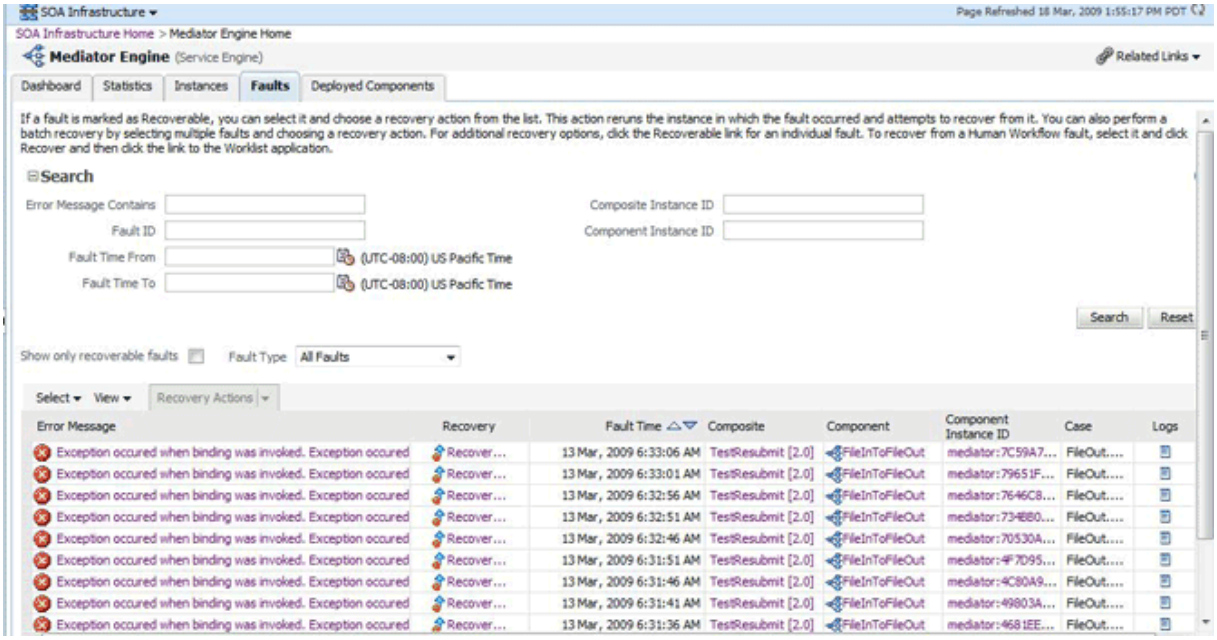
From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **Mediator**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **Mediator**.

The Mediator Engine home page appears.

2. Click **Faults**.

   The Faults page displays the following:

   - A utility so you can search for a specific fault by specifying criteria and clicking **Search**. Click the **Help** icon for more information.
   - Faults that occurred in the service component, including the error message, whether you can recover from the fault, the time at which the fault occurred, service component instance ID, activity in which the fault occurred, and a link to a log file describing the fault.



   Mediator service engine faults that are marked as recoverable can be recovered.

3. Select a fault or faults.

   As with fault recovery at the SOA Infrastructure level, SOA composite application level, and Mediator service component level, you can perform single fault recovery, bulk fault recovery, and recovery of all faults. See Step 4 of Section 17.1, "Recovering From Oracle Mediator Service Component Faults" for instructions on selecting faults to perform these types of recovery.

4. If you did not select a recovery action in the previous step, select an action from the **Recovery Action** list.

Action	Description
Retry	Retries the instance with an option to provide a retry success action. An example of a scenario in which to use this recovery action is when the fault occurred because the service provider was not reachable due to a network error. The network error is now resolved.

Action	Description
Abort	Terminates the entire instance.
Replay	Replays the entire scope activity again in which the fault occurred.
Rethrow	Rethrows the current fault.
Continue	Ignores the fault and continues processing (marks the faulted activity as a success).

5. Perform the following additional monitoring tasks from within the faults table:
   a. Click the **Show only recoverable faults** check box to only display faults from which you can recover.
   b. From the **Fault Type** list, select to display all faults, system faults, business faults, or OWSM faults in the faults table. Click **Help** for a description of these fault types.
   c. From the **View** list, select **Columns** > **Fault ID** to display the fault IDs for each error message. The fault ID is automatically generated and uniquely identifies a fault. The fault ID is also displayed when you click an error message.
   d. In the **Composite** column, click a specific SOA composite application to access its home page.
   e. In the **Component** column, click a specific service component to access its home page.
   f. In the **Component Instance ID** column, click a specific service component ID to access task details about the instance (for example, the current state of a task). Note that rejected messages do not have a component instance ID.

For more information, see the following documentation:

- Section 1.4.3.1, "Introduction to Fault Recovery" for conceptual details about faults
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

18

Managing Cross-References

This chapter describes how to delete obsolete information from cross reference tables. Cross references are used to associate identifiers for equivalent entities created in different applications.

This chapter includes the following topic:

- Section 18.1, "Deleting Cross-Reference Values"

For more information about cross-references, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

18.1 Deleting Cross-Reference Values

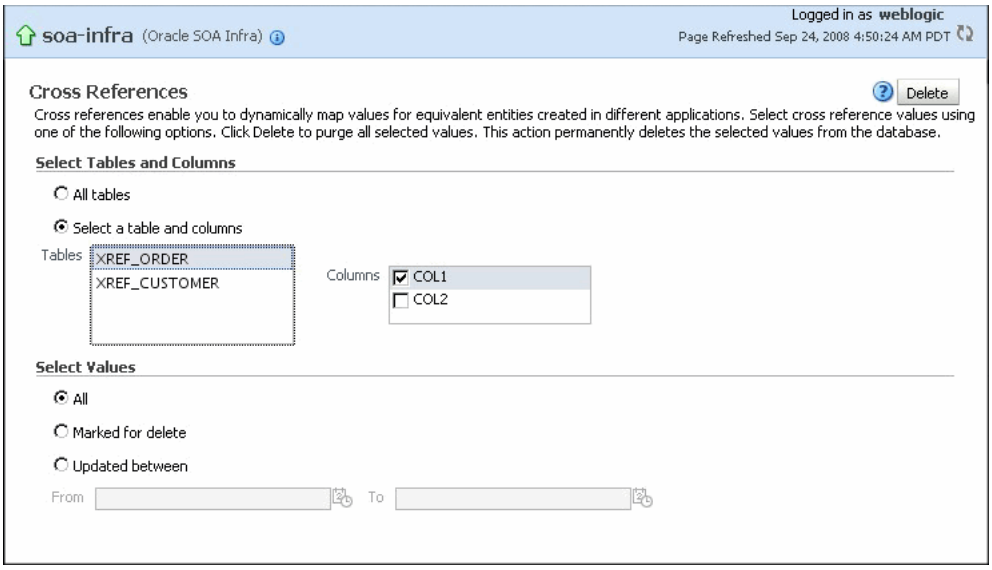
You can use the Cross References page to select tables and columns from which you want to delete the values.

To delete values from cross-reference tables:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **SOA Administration** > **Cross References**.	1. Right-click **soa-infra**. 2. Select **SOA Administration** > **Cross References**.

The Cross References page appears.



2. In the **Select Tables and Columns** section, select one of the following options:
   - **All tables**: To delete values from all tables.
   - **Select a table and columns**: To select either a specific table from which to delete values, or to select one or more columns from a specific table from which to delete values.
3. In the **Select Values** section, select one of the following options:
   - **All**: To delete all values from the selected table and columns.
   - **Marked for delete**: To delete only the values that have been marked for deletion from the selected table and columns.
   - **Updated between**: To delete values updated between a specific time periods.
4. If you select the **Updated between** option in Step 3, enter a starting date in the **From** field and an ending date in the **To** field.
5. Click **Delete**.

Part VII

Administering Decision Service Components and Business Rules Service Engines

This part describes how to administer Decision Service components and Business Rules service engines.

This part includes the following chapter:

- Chapter 19, "Monitoring Decision Service Components and Engines"

19

Monitoring Decision Service Components and Engines

This chapter describes how to monitor decision service components, including service engine instances, faults, statistics, and deployed composites; deployed decision service component instances; business rule tracing; and log files. Decision service components are also called business rules service components in the Oracle Fusion Middleware documentation.

This chapter includes the following topics:

- Section 19.1, "Monitoring Business Rules Service Engine Recent Instances and Faults"
- Section 19.2, "Monitoring Business Rules Service Engine Statistics"
- Section 19.3, "Monitoring Business Rules Service Engine Instances"
- Section 19.4, "Monitoring Business Rules Service Engine Faults"
- Section 19.5, "Monitoring Business Rules Service Engine Deployed Components"
- Section 19.6, "Monitoring Decision Service Component Instances of a Composite Application"
- Section 19.7, "Monitoring Business Rule Tracing"
- Section 19.8, "Monitoring Decision Service Component Logs"

Note: The business rules service engine does not support any user level configuration.

19.1 Monitoring Business Rules Service Engine Recent Instances and Faults

Using the business rules service engine home page Dashboard page, you can monitor recent instances and faults of decision service components running in the SOA Infrastructure. These decision service components can be part of separate SOA composite applications. Decision service components are also called business rules components in the Oracle Fusion Middleware documentation.

To monitor business rules service engine instances and faults:

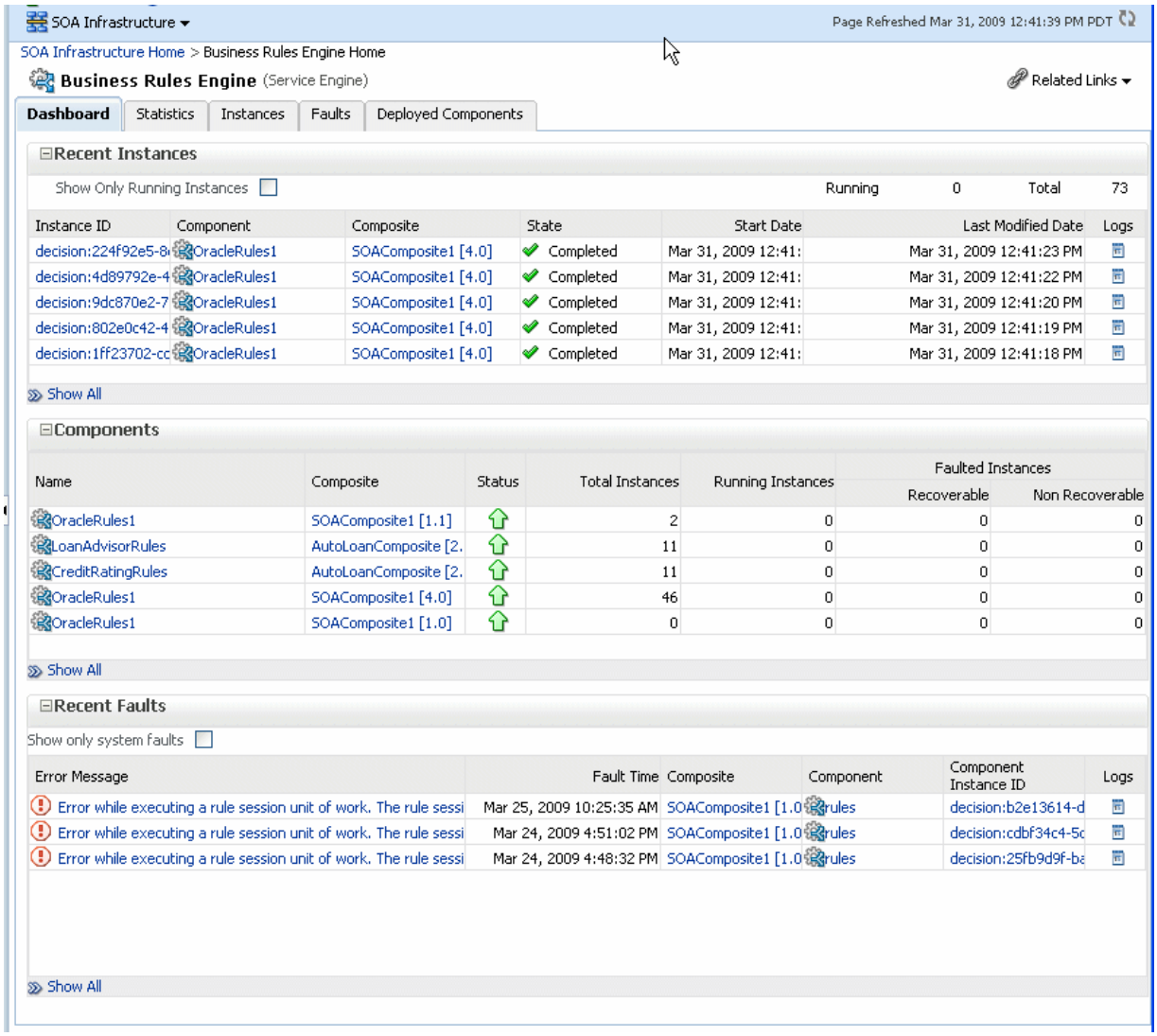
1. Access the business rules service engine home page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines > Business Rules**.	1. Select **soa-infra**. 2. Right-click and select **Service Engines** > **Business Rules**.

2. Click **Dashboard**.

   The **Recent Instances** section of the Dashboard page displays recent instances of all decision service components, including the instance ID of the decision service component, the decision service component name, the SOA composite application of which the decision service component is a part, the state of the instance (for example, completed successfully or faulted, the instance start time, the last modification time, and a **Logs** icon (clicking the **Logs** icon shows the Log Messages page with filtered messages specific to that instance)).

   Note: To see the state with the correct information, you must set the **Capture Composite Instance State** property. You can change this setting on the SOA Administration Common Properties page. Turning this feature on allows for separate tracking for running instances. However, this may impact performance. For information on setting this property, see Section 3.1, "Configuring SOA Infrastructure Properties."



3. In the **Instance ID** column, click an instance ID for a decision service component to view its audit trail.

> **Note:** The contents of the audit trail page depends on the **Audit Level** settings. When the **Audit Level** property is set to **Production**, the audit trail shows only the activity names. When the **Audit Level** is set to **Development** mode, the audit trail shows the decision service instance payload details. In other modes, for example **Off**, the audit trail does not show decision service details. You can change the **Audit Level** on the SOA Infrastructure Common Properties page. Additionally, this option can be set for a specific composite from the home page for the composite.

4. In the **Component** column, click a specific decision service component to access its home page.

5. In the **Composite** column, click a specific SOA composite application to access its home page.

6. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.
7. Click **Show All** to access the Instances page of the service engine.

The lower section of the **Dashboard** page displays the following:

- The **Components** section shows the decision service components deployed on the business rules service engine across SOA composites. It also shows the status of the SOA composites and the instance count information in the respective instance state columns.
- The **Recent Faults** section lists the recent faults in the service engine, including the error message, the time at which the fault occurred, the SOA composite application in which the fault occurred, the decision service component, the instance ID of the decision service component, and a **Logs** icon (clicking the **Logs** icon shows the Log Messages page with filtered messages specific to that instance).

For more information, see Section 1.2.4, "Introduction to Service Components and Service Component Instances."

19.2 Monitoring Business Rules Service Engine Statistics

Using the business rules service engine Statistics page, you can monitor business rules service engine performance and metrics. This page shows service engine-level, not component-level, details. Business rules service components are also called decision service components in the Oracle Fusion Middleware documentation.

To monitor business rules service engine statistics:

1. Access the business rules service engine statistics page through one of the following options:

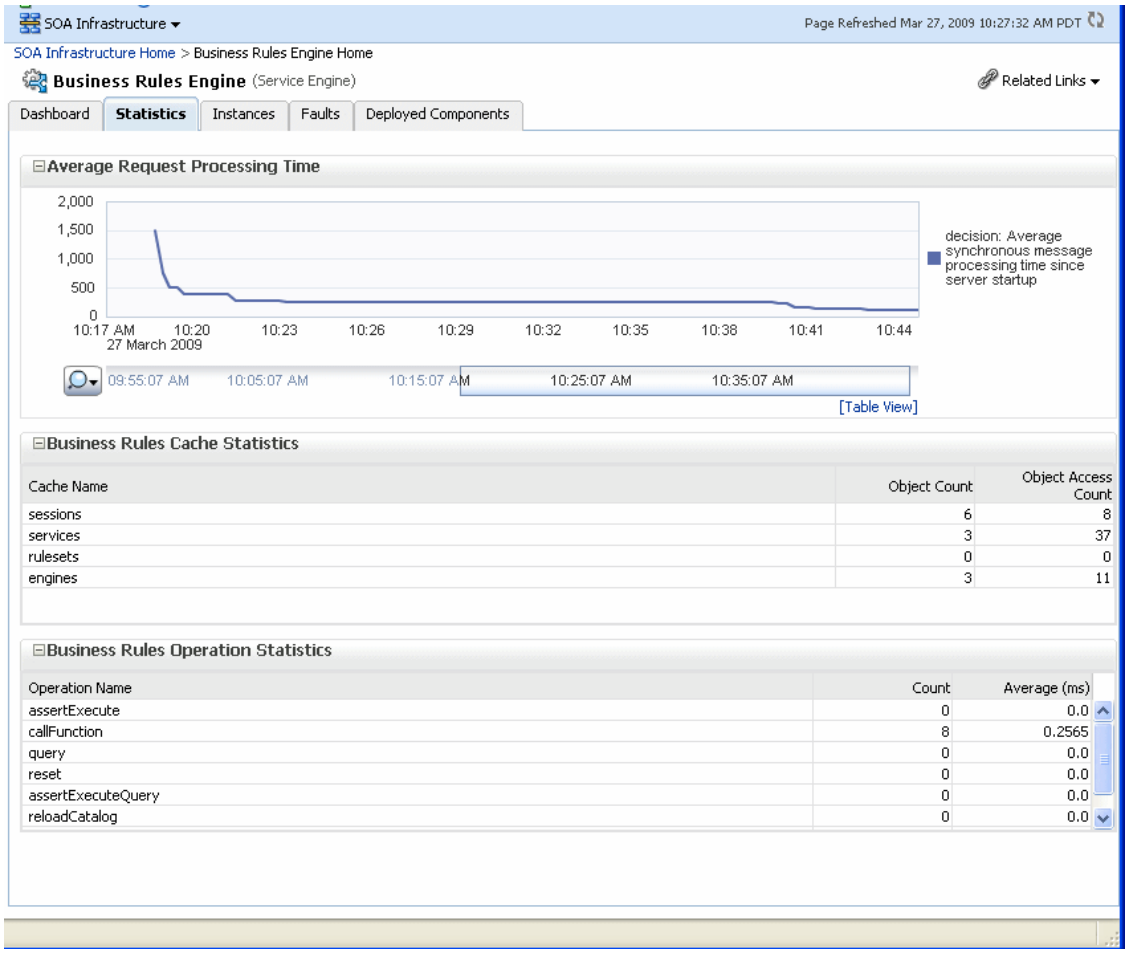
From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
Select **Service Engines > Business Rules**.	1. Select **soa-infra**. 2. Right-click and select **Service Engines** > **Business Rules**.

2. Click **Statistics**.

   The **Statistics** page displays the following:

   - **Average Request Processing Time**: This chart displays the average request processing time of the business rules service engine since server startup. That is, how many requests were processed by the service engine per unit of time.
   - **Business Rules Cache Statistics**: This section provides details about the service engine cache. This section lists the types of caches used by the service engine and the object count in each of the caches. All these metrics are based on the object count since server startup.
   - **Business Rules Operation Statistics**: This section shows the operation statistics. Using the operation statistics, you can determine the number of calls to Oracle Business Rules decision functions since server startup, and determine the total time spent in Decision Functions since server startup.

Note: When you view business rules operation statistics for composite applications created with Oracle Fusion Middleware 11*g* Release 1 (11.1.1), the only operation shown is the **callFunction** operation. In this release, the decision service only calls Oracle Business Rules using decision functions, and this operation is indicated with values for the operation named **callFunction** (with **Count** and **Average(ms)** fields). With composite applications that were migrated from older releases, the decision service performs **callFunction** operations and the other operations listed in the **Business Rules Operation Statistics** section. For these migrated projects, you can debug the flow of the request through various important operations within the service engine. Also, you can find any long-running operations and take the necessary actions. These metrics also are since server startup.



19.3 Monitoring Business Rules Service Engine Instances

Using the business rules service engine **Instances** page, you can monitor all decision service component instances. These decision service components can be part of separate SOA composite applications. Decision service components are also called business rules service components in the Oracle Fusion Middleware documentation.

To monitor business rule service engine instances:

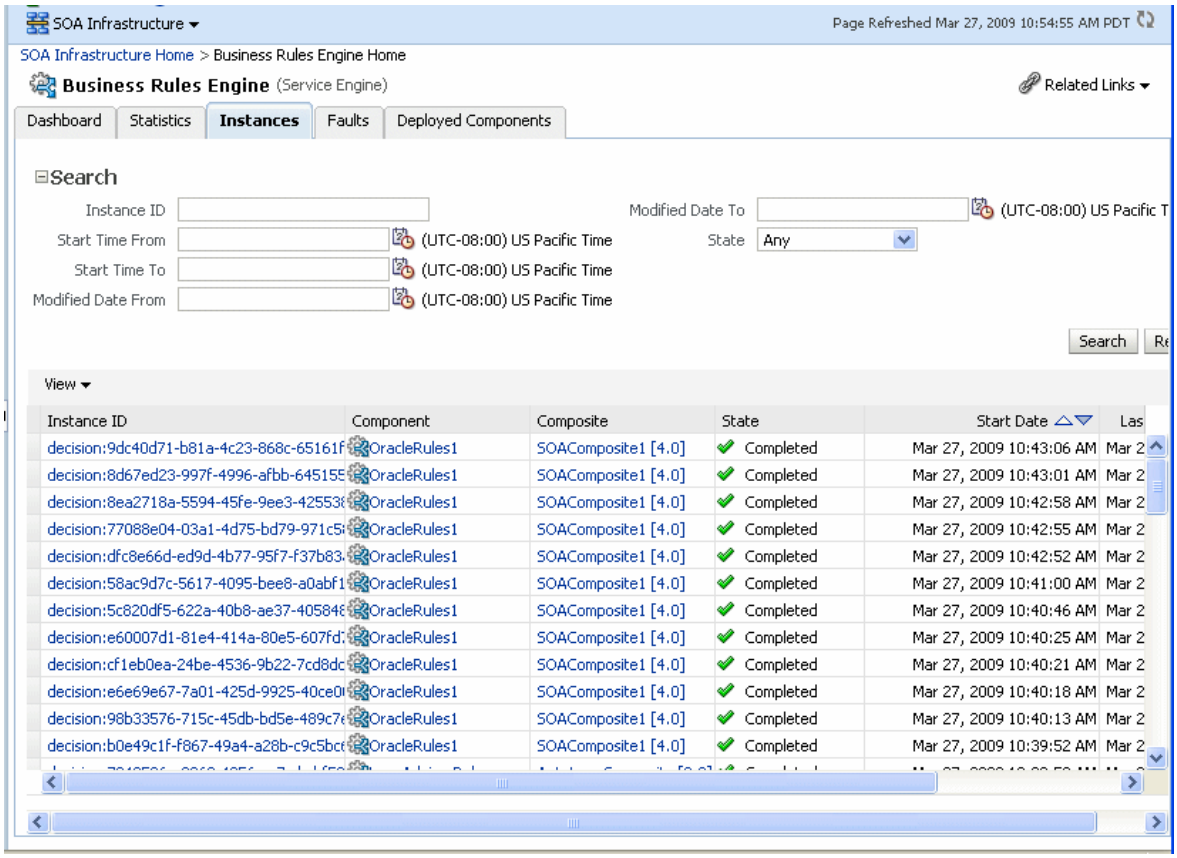
1. Access the business rules service engine Instances page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
Select **Service Engines** > **Business Rules**.	1. Select **soa-infra**. 2. Right-click and select **Service Engines** > **Business Rules**.

2. Click **Instances**.

   The Instances page displays the following:

   - A utility for searching for a specific instance by specifying criteria and clicking **Search**.
   - A list of instances, including the instance ID of the decision service component, the decision service component name, the SOA composite application name, the state of the instance (for example, completed successfully, running, or faulted), the instance start time, the last modification time, and a **Logs** icon (clicking the **Logs** icon shows the instance log messages).



3. In the **Instance ID** column, click an instance ID for a decision service component to view its audit trail details.

> **Note:** The contents of the audit trail page depends on the **Audit Level** settings. When the **Audit Level** property is set to **Production**, the audit trail shows only the activity names. When the **Audit Level** is set to **Development** mode, the audit trail shows the decision service instance payload details. You can change the **Audit Level** on the SOA Infrastructure Common Properties page. Additionally, this option can be set for a specific composite from the home page for the composite.

4. In the **Component** column, click a specific decision service component to access its home page.
5. In the **Composite** column, click a specific SOA composite application to access its home page.
6. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

For more information, see Section 1.2.4, "Introduction to Service Components and Service Component Instances."

19.4 Monitoring Business Rules Service Engine Faults

Using the business rules service engine Faults page, you can monitor all decision service component faults. The Faults page shows this information for decision service components that can be part of separate SOA composite applications. Decision service components are also called business rules components in the Oracle Fusion Middleware documentation.

To monitor business rules service engine faults:

> **Note:** Decision service component faults are always nonrecoverable.

1. Access the business rules service engine Faults page through one of the following options:

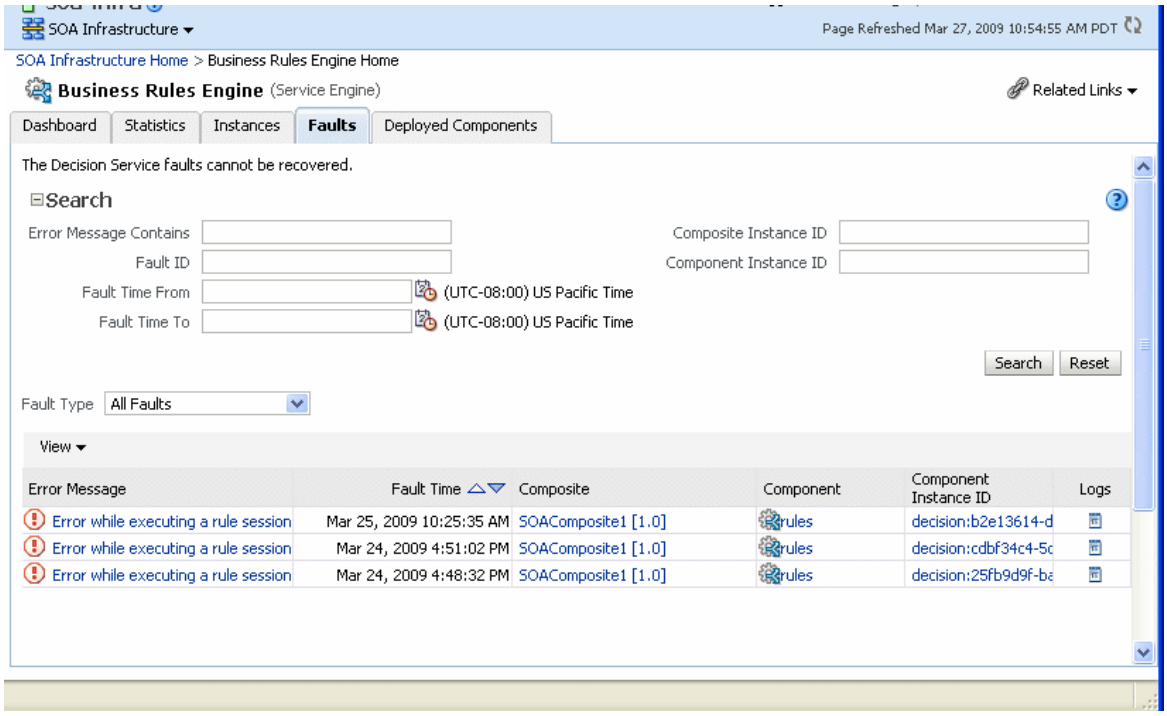
From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
Select **Service Engines** > **Business Rules**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **Business Rules**.

2. Click **Faults**.

   The Faults page displays the following:

   - A utility for searching for a specific fault by specifying criteria and clicking **Search**. Click the **Help** icon for details.
   - A list of faults that occurred in the decision service component, including the error message, the time at which the fault occurred, the SOA composite application and decision service component in which the fault occurred, the decision service component instance ID, and a **Logs** icon (clicking the **Logs** icon shows the instance log messages).

   Decision service component instance faults cannot be recovered.



3. You can perform the following monitoring tasks from within the Faults page:

   a. From the **Fault Type** list, select to display all Faults, system faults, business faults, or Oracle Web Services Manager faults in the Faults page.

   b. From the **View** list, select **Columns** > **Fault ID** to display the fault IDs for each fault. The fault ID is automatically generated and uniquely identifies a fault. The fault ID is also displayed when you click an error message.

   c. In the **Component** column, click a specific decision service component to access its home page.

   d. In the **Component Instance ID** column, click a specific decision service component instance ID to view the audit trail.

   Note: The contents of the audit trail page depend on the **Audit Level** settings. When the **Audit Level** property is set to **Production**, the audit trail shows only the activity names. When the **Audit Level** is set to **Development** mode, the audit trail shows the decision service instance payload details. You can change the **Audit Level** on the SOA Infrastructure Common Properties page. Additionally, this option can be set for a specific composite from the home page for the composite.

   e. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to the instance. Clicking the **Log** link shows the faults and error messages related to that faulted instance.

4. In the **Error Message** column, click to view the fault details.

For more information, see Section 1.2.4, "Introduction to Service Components and Service Component Instances."

19.5 Monitoring Business Rules Service Engine Deployed Components

Using the business rules service engine home page Deployed Components page, you can monitor all decision service components deployed across SOA composite applications. Decision service components are also called business rules components in the Oracle Fusion Middleware documentation.

To monitor business rule service engine deployed components:

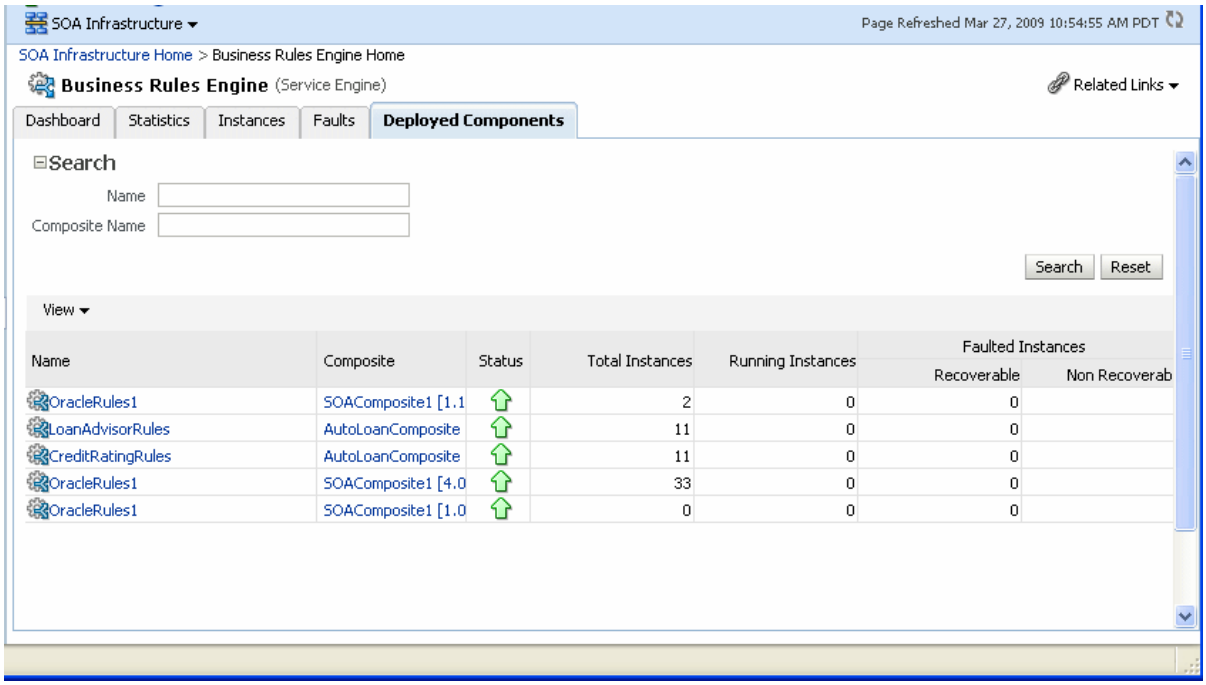
1. Access the business rules service engine Deployed Components page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
Select **Service Engines** > **Business Rules**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **Business Rules**.

2. Click **Deployed Components**.

   The Deployed Components page displays the following:

   - A utility for searching for a specific component by specifying criteria and clicking **Search**.
   - A list of components, including the name, the SOA composite application name, the status (up or down), and the instances count (total, running, and faulted).



3. In the **Name** column, click a name to navigate to the Component home page and view component details.
4. In the **Composite** column, click a specific SOA composite application to access its home page.

For more information, see Section 1.2.4, "Introduction to Service Components and Service Component Instances."

19.6 Monitoring Decision Service Component Instances of a Composite Application

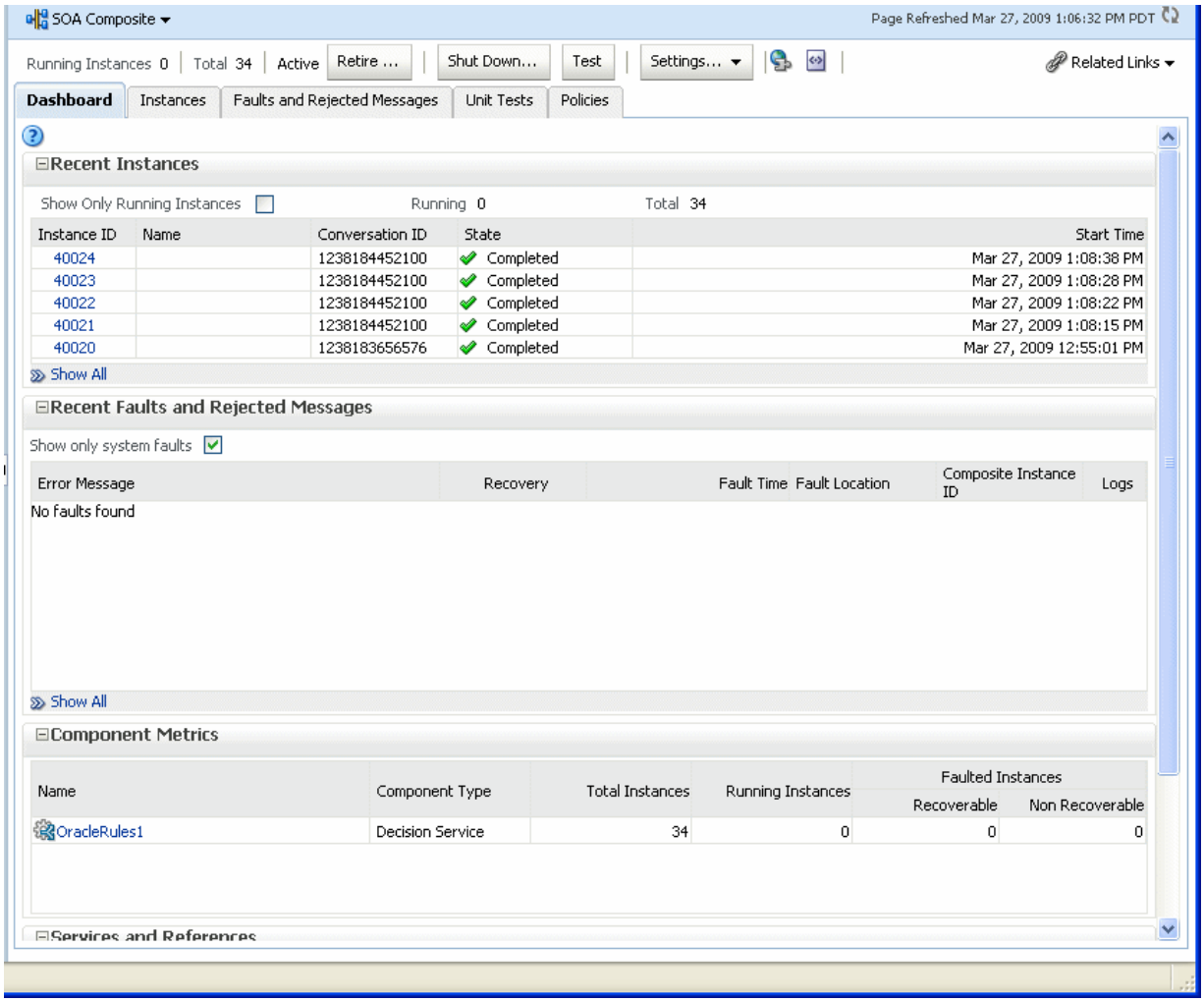
You can monitor decision service component instances of a composite application. Each decision service component instance has its own unique instance ID. This ID is in addition to the instance ID of the overall SOA composite application of which this decision service component is a part. Decision service components are also called business rules components in the Oracle Fusion Middleware documentation.

> **Note:** To see the state with the correct information, you must set the **Capture Composite Instance State** option. You can change this setting on the SOA Infrastructure Common Properties page. Turning this feature on allows for separate tracking for running instances. However, this may impact performance. For information on setting the option, see Section 3.1, "Configuring SOA Infrastructure Properties."

To monitor decision service component instances from a composite application:

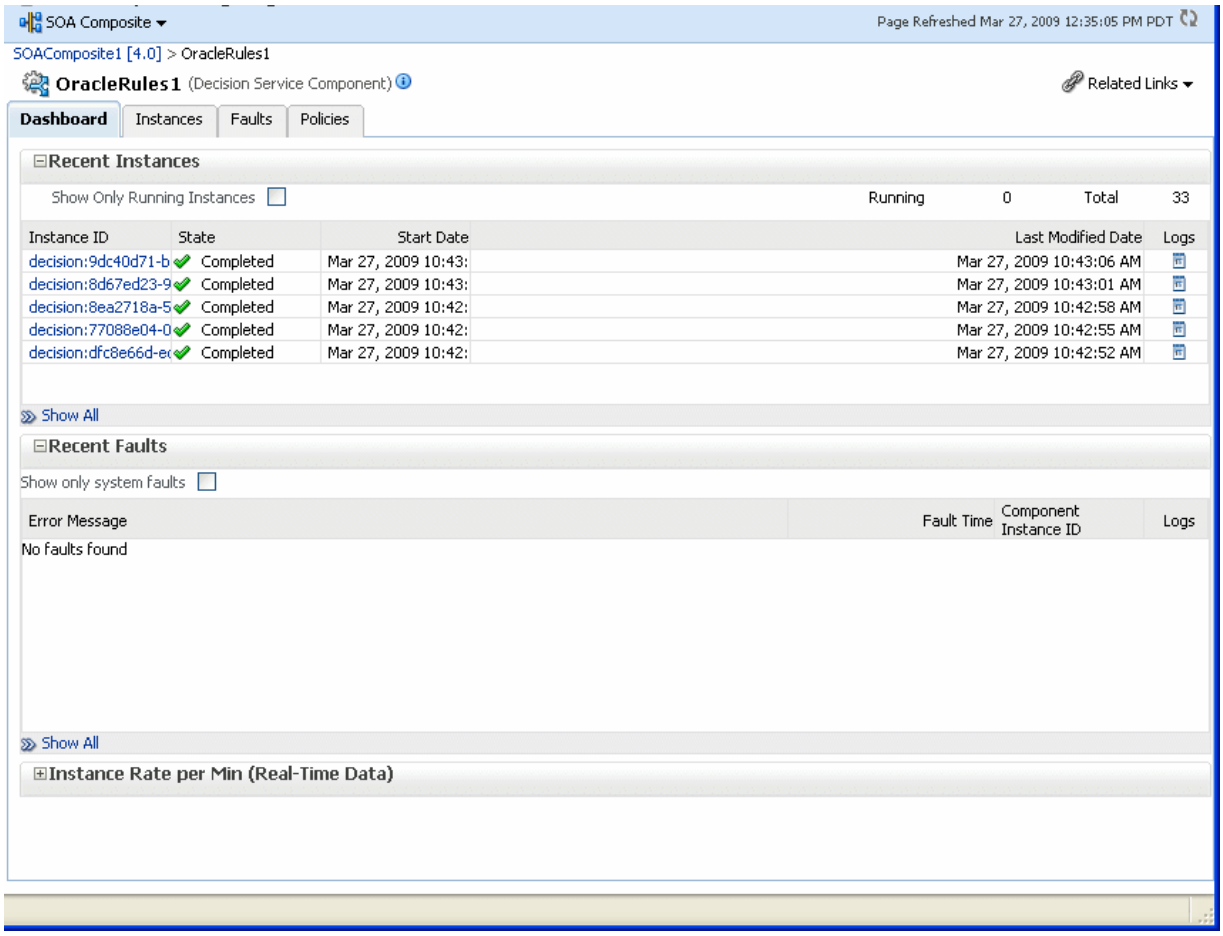
1. Access a decision service component from a composite application through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. In the navigator, select **soa-infra**. 2. From the **SOA Infrastructure** menu, Select **Home**. 3. Click **Deployed Composites** tab. 4. In the **Composite** table, select a specific SOA composite application that includes a decision service component.	1. Expand **soa-infra**. 2. Select a specific SOA composite application that includes a decision service component.



2. The **Component Metrics** section on the composite dashboard provides a high-level overview of each decision service component. This table includes columns showing the **Component Type**, the **Total Instances**, the **Running Instances**, and the **Faulted Instances** (recoverable and nonrecoverable).

3. Select a decision service component in the **Component Metrics** section to display the corresponding decision service component page.



For more information, see Section 1.2.3, "Introduction to SOA Composite Application Instances."

19.7 Monitoring Business Rule Tracing

You can use Oracle Enterprise Manager Fusion Middleware Control to perform rule execution tracing. For more information about accessing and using Fusion Middleware Control, see Chapter 2, "Getting Started with Administering Oracle SOA Suite and Oracle BPM Suite."

A rule execution trace is a mechanism of tracing business rules service engine events that occur during the evaluation of rules. The types of events traced are:

- Fact operations (assert, retract, and modify)
- Rules execution
- Rule activation
- Ruleset stack changes
- Rule compilation
- Reset (required for maintaining state during analysis)

Each trace contains information about the event that it traces. For example, a rule trace entry for an executed rule consists of:

- Rule name (RL name)

- Execution sequence number
- List of fact IDs for the facts that matched this rule
- Timestamp in milliseconds

Rule execution trace audit levels are the same as the audit levels supported in the SOA Infrastructure:

- **Off:** Rule execution tracing is disabled. The decision component instance is not created at all.
- **Development:** Full rule execution tracing that contains all the details about facts (listing, operations such as modify and assert), rule activation, pop or push rulesets, and so on. It also provides a list of fact IDs on which the executed rules are matched. See Section 19.7.1, "Tracing Rule Execution at the Development Audit Level" for an example.
- **Production:** The executed rules are traced. All the details about facts, rule activation, pop or push ruleset are not available. The trace do not contain a list of the matching facts IDs. See Section 19.7.2, "Tracing Rule Execution at the Production Audit Level" for an example.

You can set audit levels either at the SOA Infrastructure level or at the composite level. See Section 3.1, "Configuring SOA Infrastructure Properties" for SOA Infrastructure audit level configuration information. See Section 1.4.1.1, "Introduction to the Order of Precedence for Audit Level Settings" for a discussion about audit level precedence when set at the SOA Infrastructure level and the composite level. The following sections discuss setting audit levels at the composite level for the purposes of rule execution tracing.

19.7.1 Tracing Rule Execution at the Development Audit Level

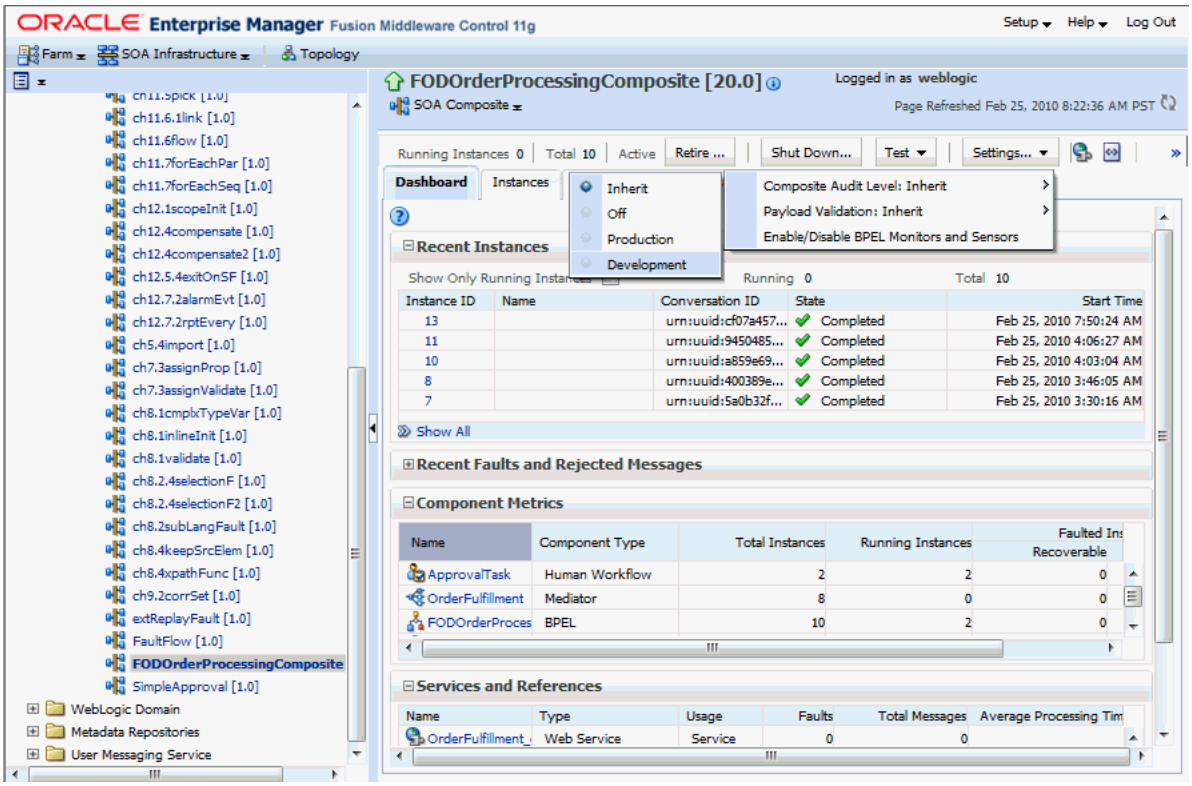
Setting the audit level to Development enables you to view all the details pertaining to a rule that has been executed.

To perform a development-level rule execution trace:

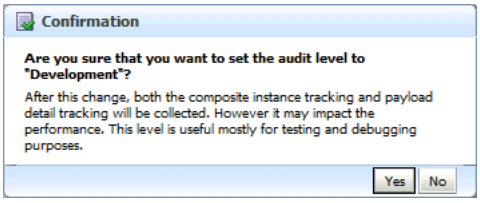
1. Open the composite application in Oracle Enterprise Manager Fusion Middleware Control.

   A list of the recent composite instances is shown on the composite Dashboard page.

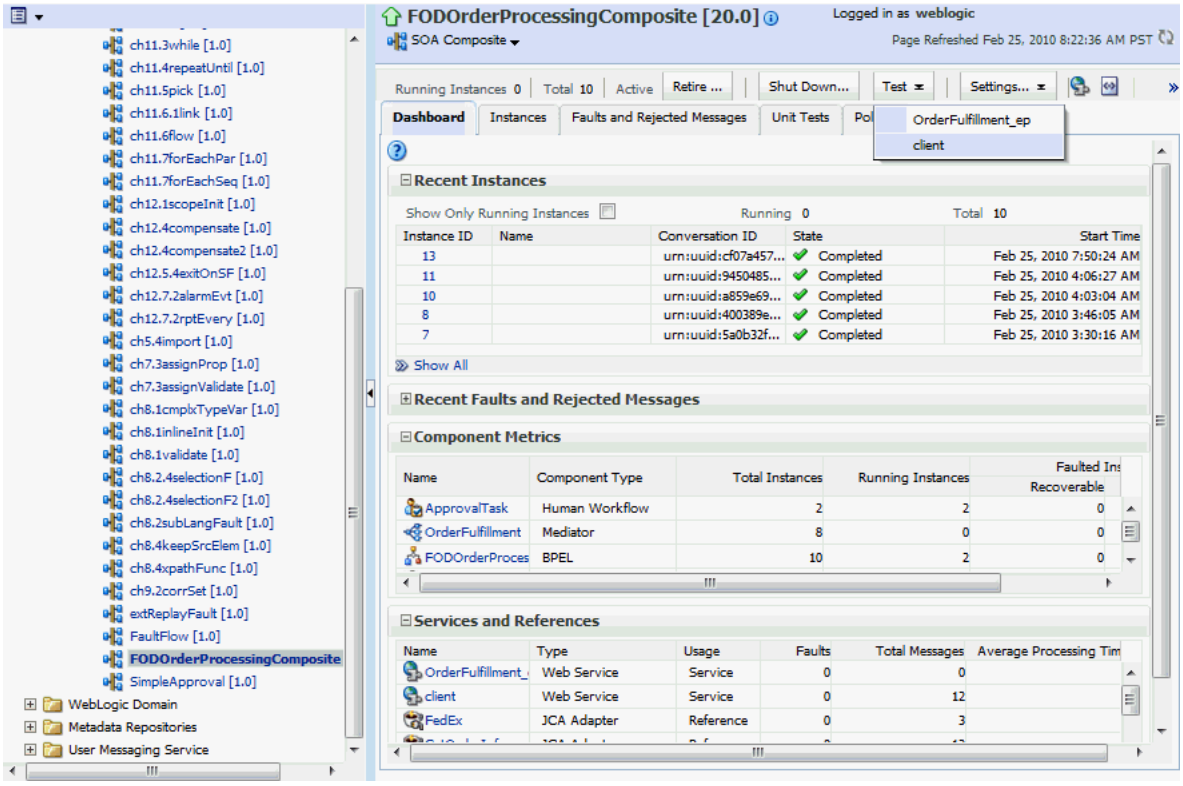
2. Click the **Settings** list, select **Composite Audit Level**, > **Development** to set the trace level as Development at the composite level.



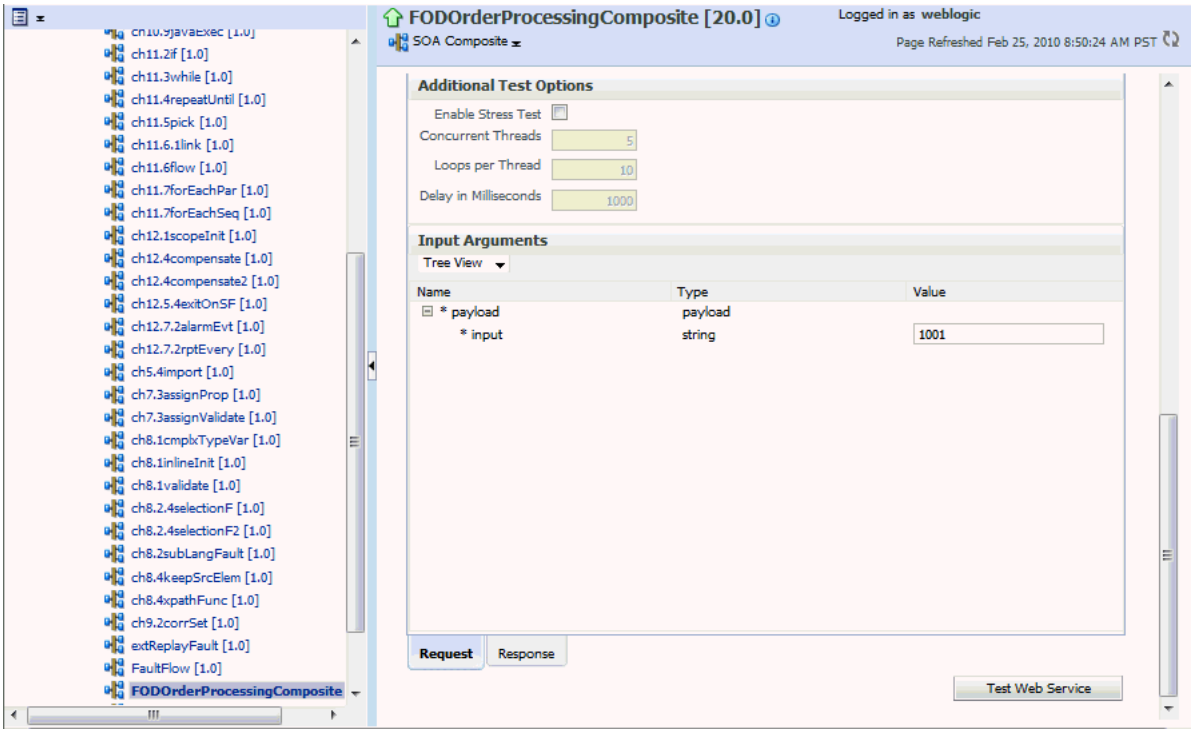
3. Click **Yes** in the Confirmation dialog.



4. Click **Test** and then **client** to invoke a test instance of the composite to view the decision traces corresponding to different input parameters, such as the Order ID.



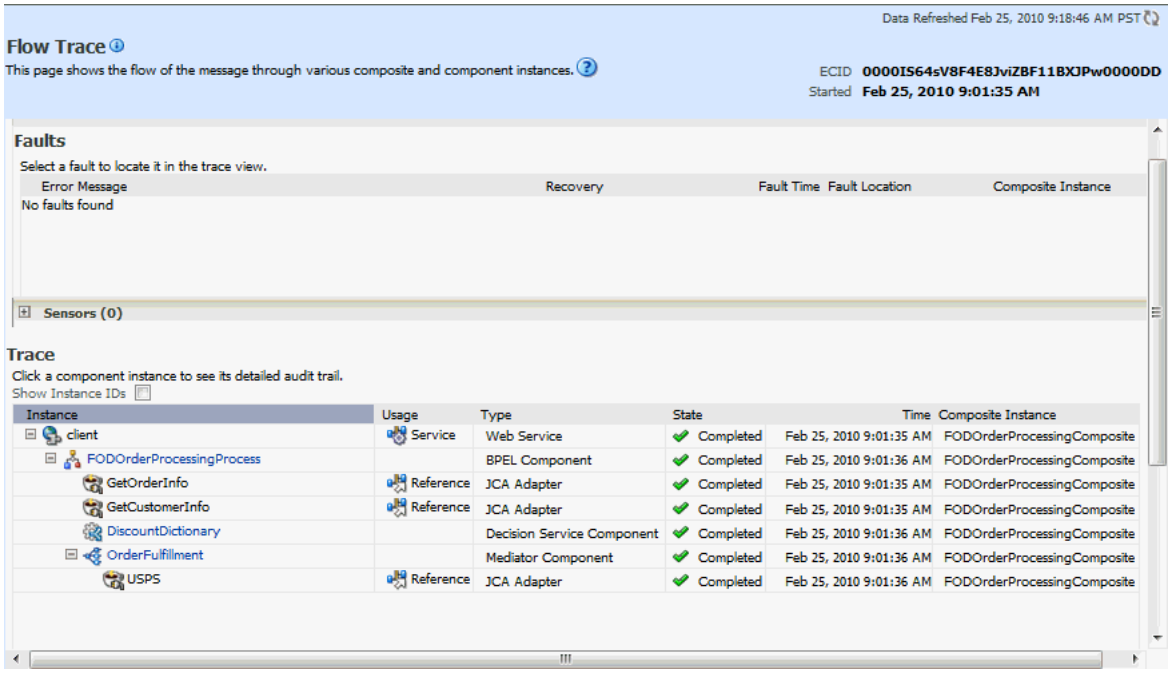
5. Enter an Order ID, for example `1001`, in the **Value** field in the **Input Arguments** section on the Test Web Service page and click the **Test Web Service** button.



Based on the input Order ID, the service invokes a BPEL process instance containing the details of the Order ID, and the rule that is relevant to the order details is executed.

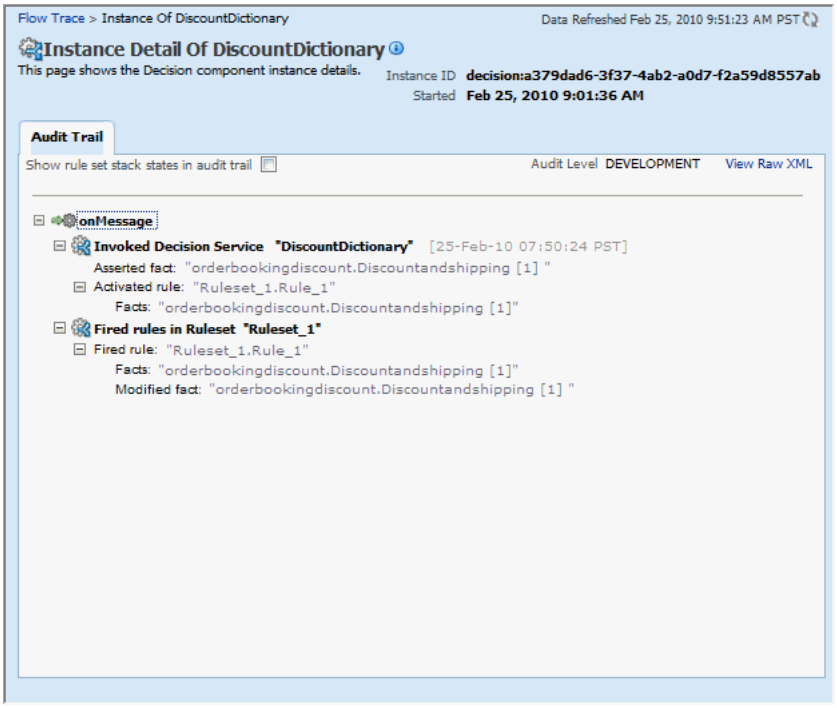
6. Click **Launch Message Flow Trace** under the **Response** tab to open the Flow Trace page.

7. Click the decision service component instance called **DiscountDictionary** in the **Trace** section to view the actual rule execution trace.

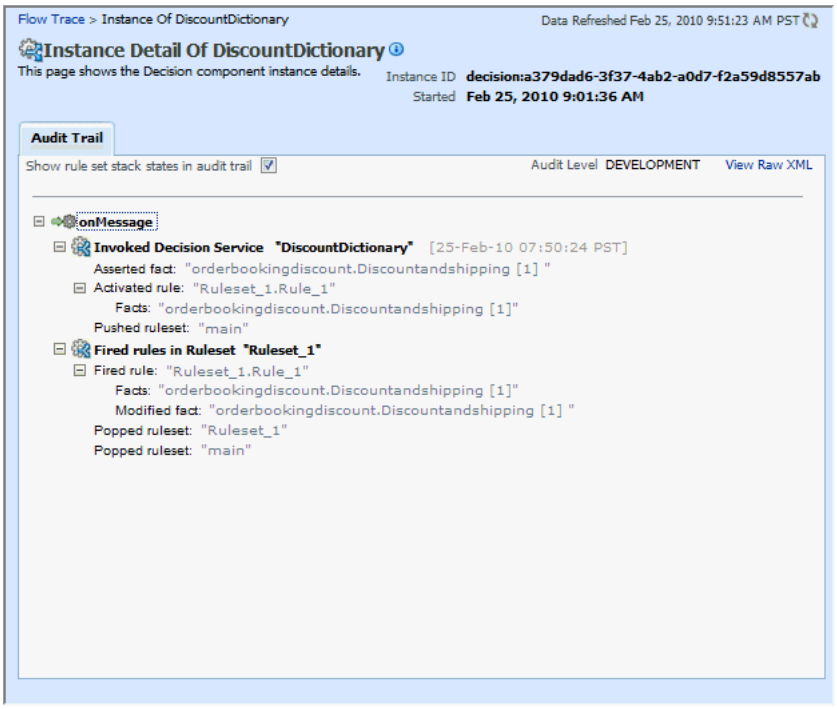


Note: You can also view the values of composite variables before and after invocation of business rule component. You must click the BPEL process component instance in the Flow Trace page, and then click the relevant payload. In this case, the BPEL component name is FODOrderProcessingProcess.

The following graphic shows the execution trace for the decision service component called **DiscountDictionary**.



8. Click the **Show rule set stack states in audit trail** checkbox to view further details of the rule execution.



The development-level trace report displays the fact name, activated rule, and the pushed and popped ruleset names.

The following table lists the entries of the trace report:

Element	Description
Invoked Decision Service Name	Displays the name of the decision service component that is invoked.
Asserted Fact	Displays the name of the fact that is passed by the BPEL service to the decision service component. This is based on the input parameter provided by the user, which according to the example is the Order ID `1001`.
Activated Rule	Displays the name of the rule that is activated, along with the name of the fact that activated the rule.
Pushed ruleset	Displays the name of the ruleset that has been invoked when the fact is asserted.
Fired rules in Ruleset	Displays the name of the ruleset whose rule is executed.
Fired rule	Displays the name of the rule that is executed.
Modified fact	Displays the name of the fact that has been modified because of the rule execution. This modified fact is then passed by the decision service component to the BPEL service.
Popped ruleset	Displays the name of the ruleset for which execution is complete and so, the ruleset is taken out of the execution queue.

Notes: The entry `Invoked Decision Service Name` appears differently in different scenarios:

- For AS10.1.3.*x* to AS11 upgraded rules dictionaries with `AssertExecuteWatch` patterns, the entry appears in the trace report as `Invoked Decision Service`.
- For AS10.1.3.*x* to AS11 upgraded rules dictionaries with `CallFunction` patterns, the entry appears in the trace report as `Invoked Decision Function`.
- For AS11 created dictionaries, the entry is displayed as `Invoked Decision Function`.

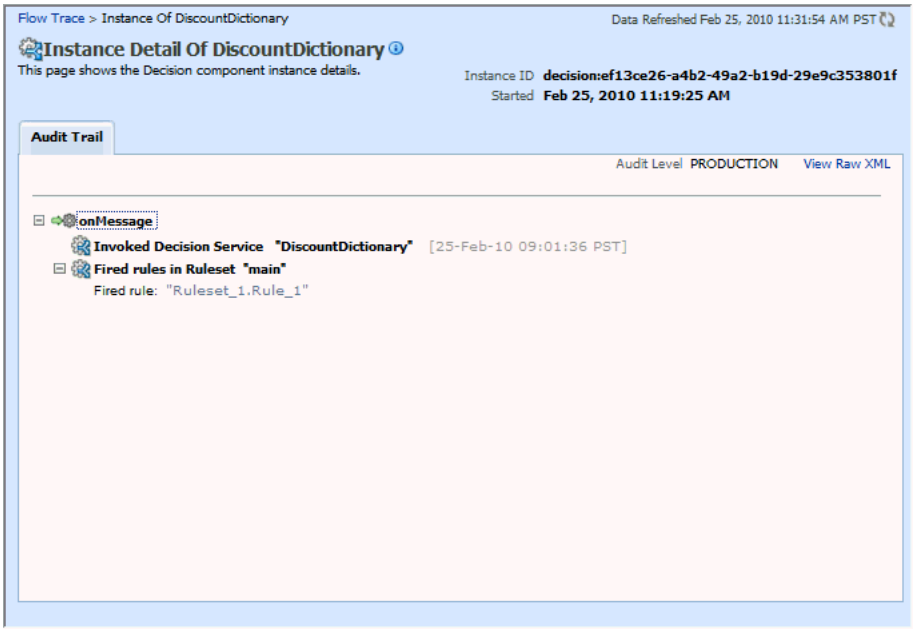
19.7.2 Tracing Rule Execution at the Production Audit Level

Setting the audit level to **Production** provides a truncated report on the rule execution trace. It only displays the ruleset and the rules that have been fired and does not display details about facts, rule activation, and so on.

The process of production-level tracing is similar to the development-level tracing. However, for Production-level tracing, you must do the following:

In Fusion Middleware Control, after opening the composite, select **Production** from the **Composite Audit Level** of the **Settings** menu.

The following graphic shows the Flow Trace page that displays the trace report.



The Production-level trace report contains only the name of the ruleset and the rules that were fired. In addition, the **Show rule set stack states in audit trail** checkbox that provides a drill-down detailed trace report is unavailable in the Production-level trace report.

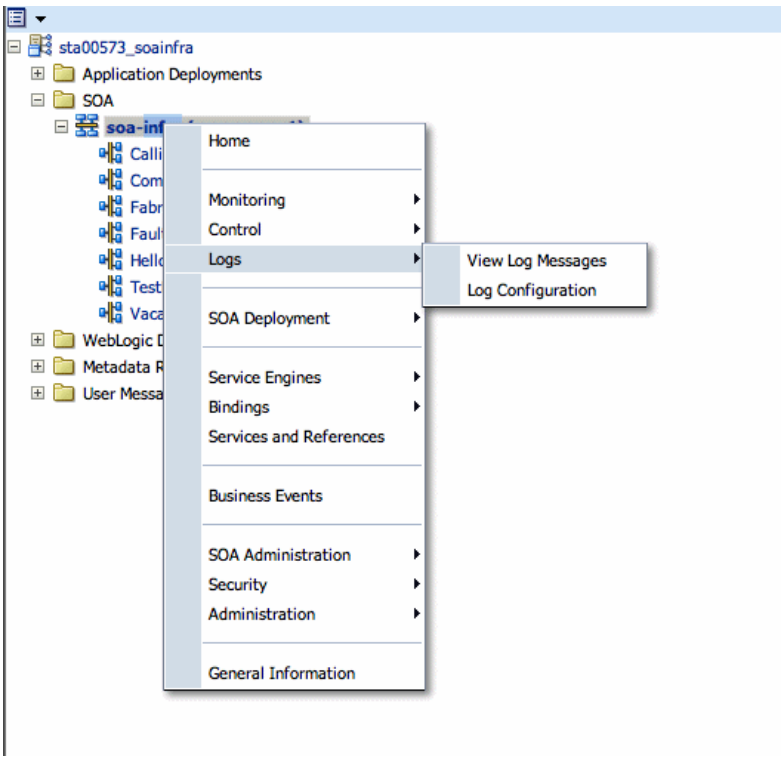
19.8 Monitoring Decision Service Component Logs

You can monitor decision service component logs. Decision service components are also called business rules service components in the Oracle Fusion Middleware documentation.

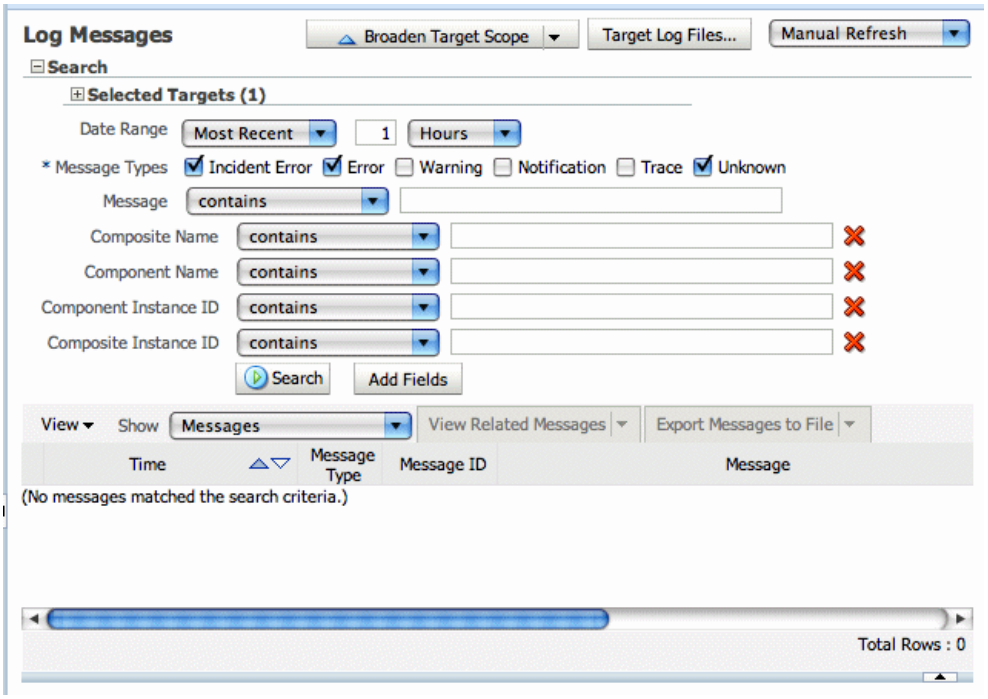
19.8.1 Viewing Decision Service Component Logs

To view decision service component logs:

1. In the navigation tree, select and right-click **soa-infra**.
2. Select **Logs** > **View Log Messages**. This displays the Log Messages page.

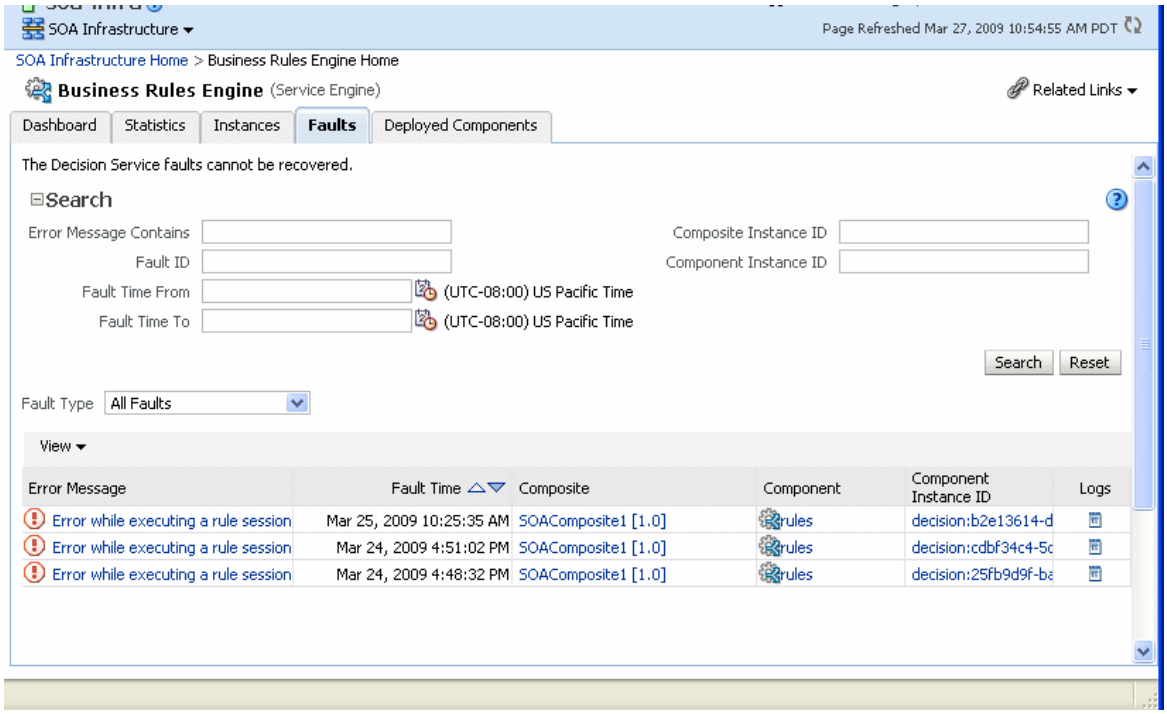


The Log Messages page opens. Use this page to select target log files.



3. To access a prefiltered list of log files for each instance or fault, click in the **Logs** column from any specific page (for example, in the decision service engine or component's faults or instances tables).

   For example, from the Faults table, click the **Logs** column.

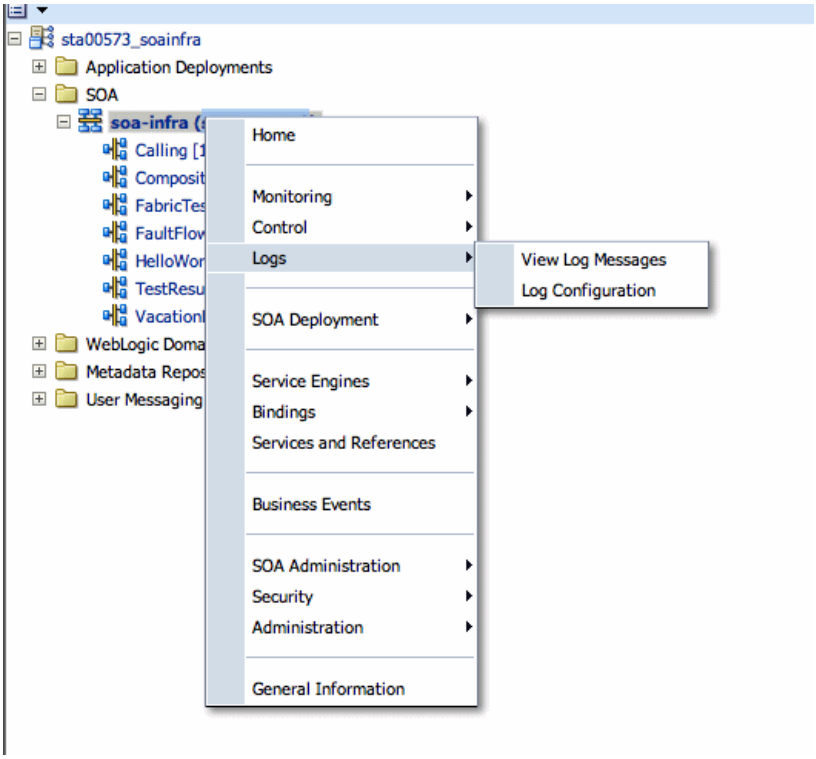


19.8.2 Setting the Diagnostic Logging Level with a Log Configuration

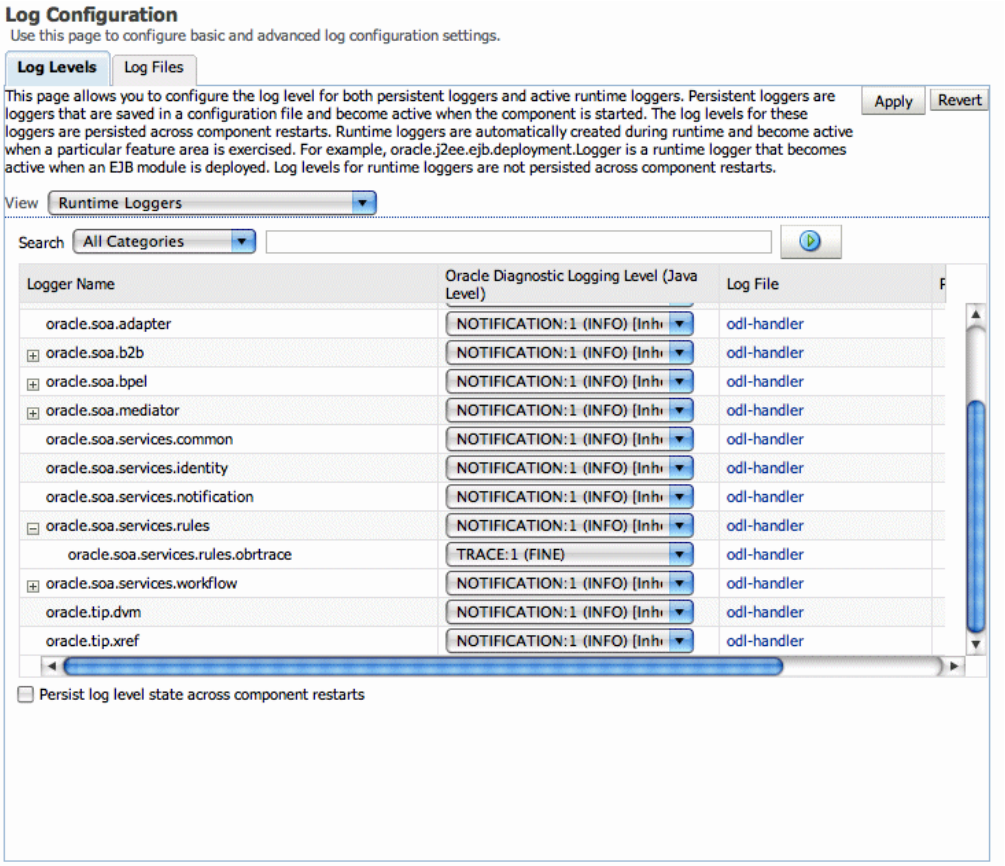
Use the Log Configuration page to configure the logging level.

To set the diagnostic logging level with a log configuration:

1. Right-click **soa-infra**, and select **Logs** > **Log Configuration**.



2. To configure the decision service component logging level, expand the **oracle.soa.service.rules** and the **oracle.soa.services.rules.obrtrace** loggers and set the notification level.


Log Configuration
Use this page to configure basic and advanced log configuration settings.
Log Levels
Log Files
This page allows you to configure the log level for both persistent loggers and active runtime loggers. Persistent loggers are loggers that are saved in a configuration file and become active when the component is started. The log levels for these loggers are persisted across component restarts. Runtime loggers are automatically created during runtime and become active when a particular feature area is exercised. For example, oracle.j2ee.ejb.deployment.Logger is a runtime logger that becomes active when an EJB module is deployed. Log levels for runtime loggers are not persisted across component restarts.
Apply
Revert
View
Runtime Loggers
Search
All Categories
Logger Name
Oracle Diagnostic Logging Level (Java Level)
Log File
oracle.soa.adapter
oracle.soa.b2b
oracle.soa.bpel
oracle.soa.mediator
oracle.soa.services.common
oracle.soa.services.identity
oracle.soa.services.notification
oracle.soa.services.rules
oracle.soa.services.rules.obrtrace
oracle.soa.services.workflow
oracle.tip.dvm
oracle.tip.xref
NOTIFICATION:1 (INFO) [Inh
TRACE:1 (FINE)
odl-handler
Persist log level state across component restarts

Part VIII

Administering Human Task Service Components and Human Workflow Service Engines

This part describes how to administer human task service components and human workflow service engines.

This part includes the following chapters:

- Chapter 20, "Configuring Human Workflow Service Components and Engines"
- Chapter 21, "Monitoring Human Workflow Service Components and Engines"
- Chapter 22, "Managing Human Workflow Service Components and Engines"

20

Configuring Human Workflow Service Components and Engines

This chapter describes how to configure human task service components and the human workflow service engine, including how to configure the notification mode for messages and actionable addresses. It also describes how to configure task service properties such as the actionable email account name and an alternative authentication provider such as Oracle Internet Directory, Microsoft Active Directory, or Sun iPlanet.

This chapter includes the following topics:

- Section 20.1, "Configuring Human Workflow Notification Properties"
- Section 20.2, "Configuring Human Workflow Task Service Properties"
- Section 20.3, "Configuring Oracle HTTP Server for Task Form Attachments"
- Section 20.4, "Configuring the Pluggable Notification Service"
- Section 20.5, "Globally Disabling the Automatic Release Timers for Oracle BPM Worklist Tasks"
- Section 20.6, "Configuring the Number of Email Notification Messages"
- Section 20.7, "Configuring Multiple Send Addresses"
- Section 20.8, "Configuring Notification Retries"
- Section 20.9, "Configuring the Identity Service"
- Section 20.10, "Seeding Users, Groups, and Application Roles using LDAP Tools"
- Section 20.11, "Enabling Case Agnostic Group Names in Human Tasks"
- Section 20.12, "Configuring Security Policies for Human Workflow Web Services"

20.1 Configuring Human Workflow Notification Properties

You can configure human workflow notification properties, such as setting the notification mode for messages and setting actionable addresses. These properties are used to notify users of changes to the state of a task. Workflow notifications can use three types of addresses:

- From address: For sending notifications.
- Actionable address: For receiving actionable responses.
- Reply to address: For receiving reply notifications.

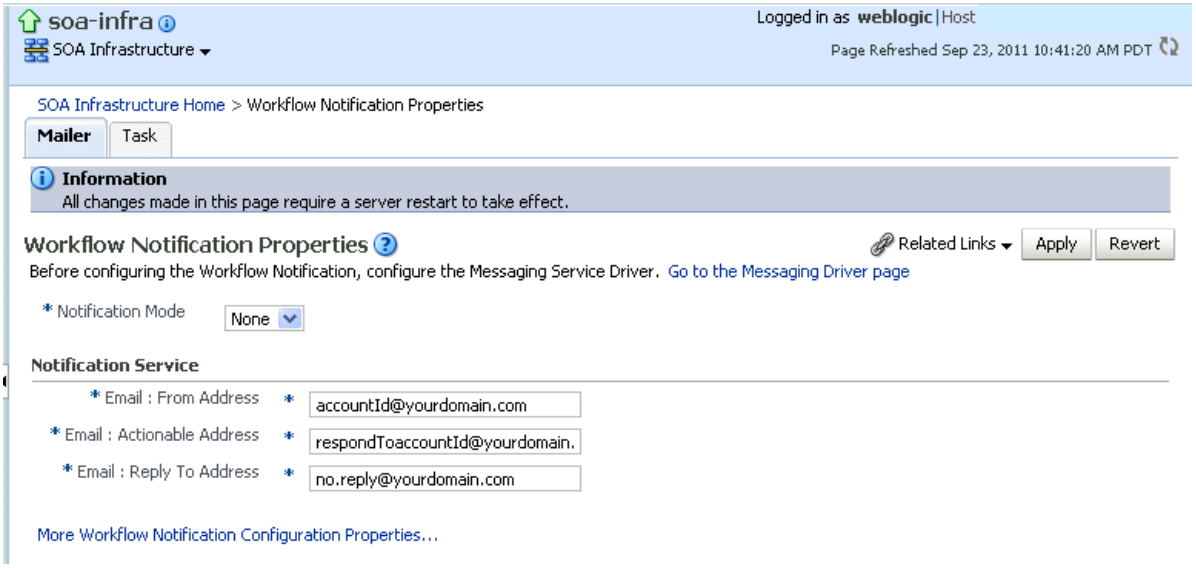
> **Note:** In the following procedures, you must configure your channel drivers before configuring your workflow notification properties. Ensure that you know all necessary driver addresses before beginning (for example, the incoming IMAP and outgoing SMTP email servers).

To configure human workflow notification properties:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **SOA Administration** > **Workflow Config** > **Mailer** tab	1. Right-click **soa-infra**. 2. Select **SOA Administration** > **Workflow Config** > **Mailer** tab

The Workflow Notification Properties page appears.

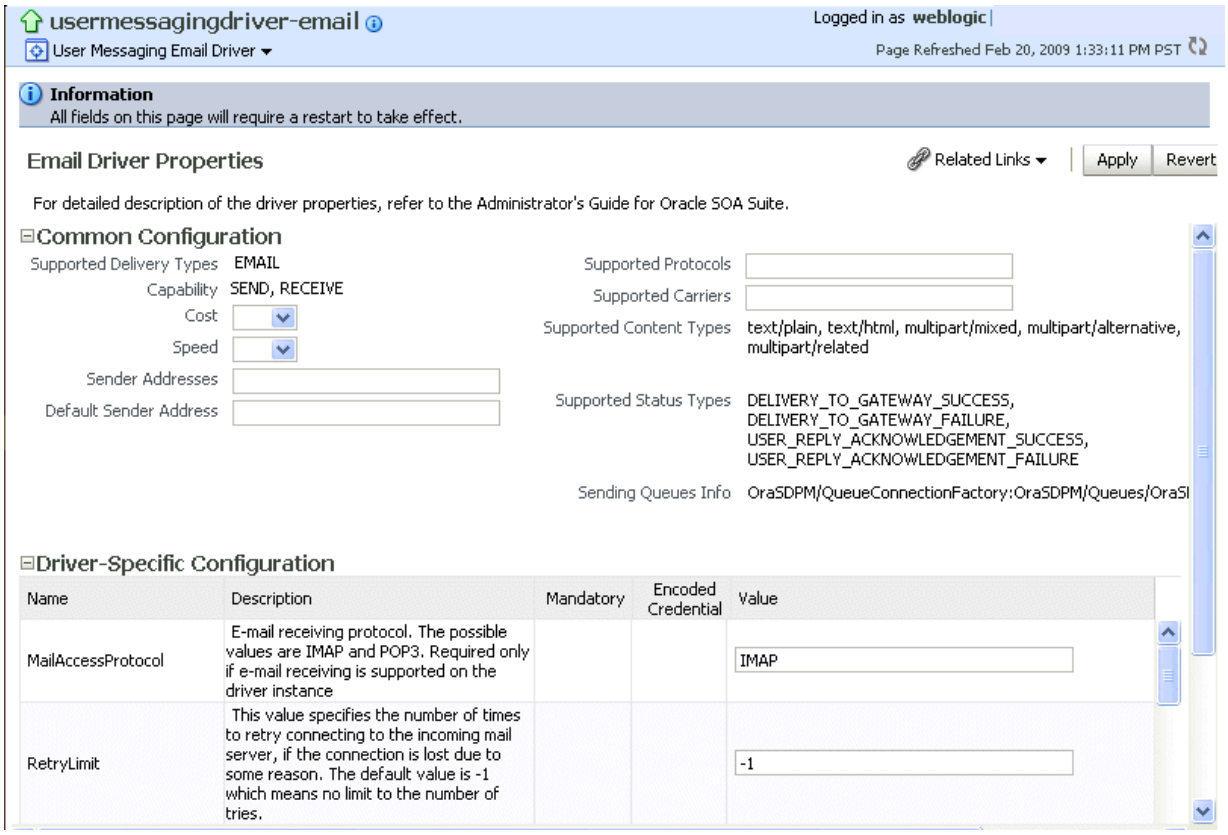


You now configure Oracle User Messaging Service to send and receive notifications. During configuration, you provide the addresses that are used by human workflow.

2. Click **Go to the Messaging Driver Page**.
3. Click **Configure Driver** in the upper right-hand section of the page. This takes you to a page to configure the messaging service driver, including properties such as the following:
   - Incoming IMAP and outgoing SMTP email servers.
   - Outgoing server user names and passwords.
   - List of sender addresses and the default sender address. (The addresses you specify for these properties must match the addresses you specify in the **Email: From Address** and **Email: Actionable Address** fields of the Workflow Notification Properties page.)

   For handling incorrect email responses, the email driver should be configured to handle incoming mail. This action enables human workflow participants to receive

and forward notifications. Messaging drivers support the various messaging transports. For instructions, see Section 26.4.1, "Configuring a Driver."



Notes:

- The hostname and IP address of the email server with which you configure must also be added to the `/etc/hosts` file of the server on which Oracle SOA Suite is running. For example, if the hostname is `xyz.oracle.com` and the IP address is `aa.bb.cc.dd`, then add this information to the `/etc/hosts` file.
- After you configure the inbound (IMAP) email server, the outbound (SMTP) email server, or both, you must restart the managed Oracle WebLogic Server on which the SOA Infrastructure is configured for these setting to take effect.

4. Return to the Workflow Notification Properties page.
5. Specify the mode of the notification service. The possible values are:
   - **ALL**: The email, short message service (SMS), instant message (IM), and voice channels are configured and notification is sent through any channel that you use.
   - **EMAIL**: Only the email channel is configured for sending notification messages.
   - **NONE**: No channel is configured for sending notification messages. This is the default setting.
6. Specify notification channel values:

Field	Description	Example
Email: From Address	Enter the outgoing email address from which end users receive notifications. The address you specify must match the sender addresses and the default sender address that you specify on the Email Driver Properties page of the Oracle User Messaging Service. **Note:** You can only receive error messages when the outgoing email address is also configured to receive incoming messages. This ensures that error messages from incorrect or nonexistent email addresses are captured by the server. Even if you configure a separate incoming account in the **Email: Reply To Address** field, error messages do not appear in the server logs.	`workflow.notifications@mycompany.com`
Email: Actionable Address	Enter the incoming email address for performing task actions. The actionable email account is the account in which task action-related emails are received and processed by human workflow. The address you specify must match the sender addresses and the default sender address that you specify on the Email Driver Properties page of the Oracle User Messaging Service.	`workflow.actions@mycompany.com`
Email: Reply To Address	Enter the address to display in emails sent out from Oracle SOA Suite. It can be a dummy address such as `no.reply@myoracle.com` or a valid address. If a valid address is provided, and configured in the Messaging Driver page, then if a user replies to actionable emails, human workflow sends an automated email indicating the correct usage. This is another incoming email account.	`workflow.no.reply@mycompany.com`

7. Click **Apply**.
8. To configure advanced notification properties in the System MBean Browser, click **More Workflow Notification Configuration Properties**. Properties that are displayed include, but are not limited to, the following. Descriptions are provided for each property.
   - **ASNSDriverIMAddress**: The address at which to receive incoming instant messages (IMs).
   - **CustomNSDriverPropertyNames**: Returns custom notification services property names.
   - **FaxCoverPageCount**: The return number of configured fax cover pages.
   - **RetryNotificationMessageThrottle**: The number of email notification messages that can be processed during notification retry cycles. For more information, see Section 20.6, "Configuring the Number of Email Notification Messages."

9. Make changes appropriate to your environment.

Note: If your IM message contains content that appears to be actionable, then acting upon the task from within the message does not cause any action to be taken. For example, acting upon the task in the following IM message does not cause any action to occur.

```
Help desk request for wfaulk  Task Help desk request for  wfaulk
requires your attention.  NOTE: You can act on the task by
copy-pasting one of following  lines as your response.

RESOLVED : [[NID]] :
Pt12uRUu9H+Xem4NYS2o7dKDtqNLs42d4YIs8ySO8Gn0ZVYFsb1SQVenRukRE+
IcE7c4XDb+tPazvP v9T2iA0qylDg0bTaVxX13HhsrCYAg= : [[NID]]
UNRESOLVED : [[NID]] :
xT9l06rbaGRAey+BtgQyJIXk62mkFtCe7ocKxwNLIsPzyE5/7AnGwXlBodEgQxr6
jorvsw2F54k/C1 r5mvyAJpAp4I4IekOHi4qhQ3eSbBHdzET1IL4F3qV/KZ/BAUsq :
[[NID]]
```

For more information about notifications and the User Messaging Service, see the following documentation:

- Part X, "Administering Oracle User Messaging Service"
- Part "Using Oracle User Messaging Service" of *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite* "

20.2 Configuring Human Workflow Task Service Properties

You can assign the actionable email account name, specify workflow session time out and custom class path URL properties values, configure dynamic assignment and task escalation functions of the assignment service, and set additional human workflow properties.

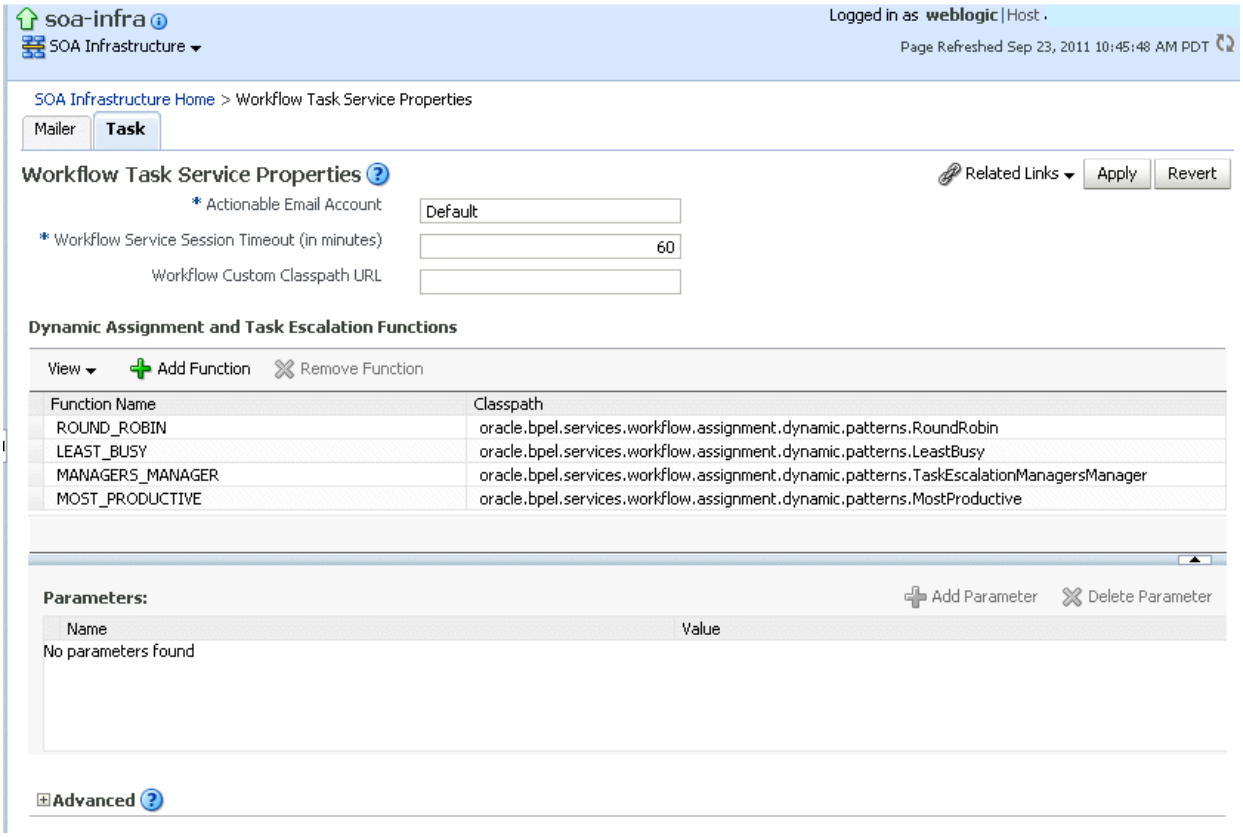
Dynamic assignment functions select a particular user or group from either a group, or from a list of users or groups. The selection is made according to criteria specific to the particular dynamic assignment function.

To configure human workflow task service properties:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **SOA Administration** > **Workflow Config** > **Task** tab.	1. Right-click **soa-infra**. 2. Select **SOA Administration** > **Workflow Config** > **Task** tab.

The upper part of the Workflow Task Service Properties page displays the field for the actionable email account and the automatically defined dynamic assignment functions.



2. Enter the following details.

Function	Description
Actionable Email Account	Enter the incoming, actionable email account to use. The default account name is **Default**, which is the account configured in Section 20.1, "Configuring Human Workflow Notification Properties." If a different account name is specified in this field, then create and configure the account as described in Section 20.7, "Configuring Multiple Send Addresses."
Workflow Service Session Timeout (in minutes)	Enter the length of time that users logged in to the Oracle BPM Worklist can remain inactive before their session expires, and they are required to log in again. This also applies to authenticated sessions created through one of the `TaskQueryService` authentication methods.
Workflow Custom Classpath URL	Enter the URL class path. This is the class path used by workflow services to look up classes implementing custom dynamic assignment and task escalation functions, custom callbacks, and customized instances of the system resource bundle, `WorkflowLabels.properties`. This can be any valid URL (either a local file path or remote URL). The class path can specify either a directory or a JAR file. If the URL specifies a directory, it must include a trailing slash ('/') character.

3. Go to the **Dynamic Assignment and Task Escalation Functions** section.

   The dynamic assignment functions are defined in the following table. You can also create your own functions and register them with the workflow service.

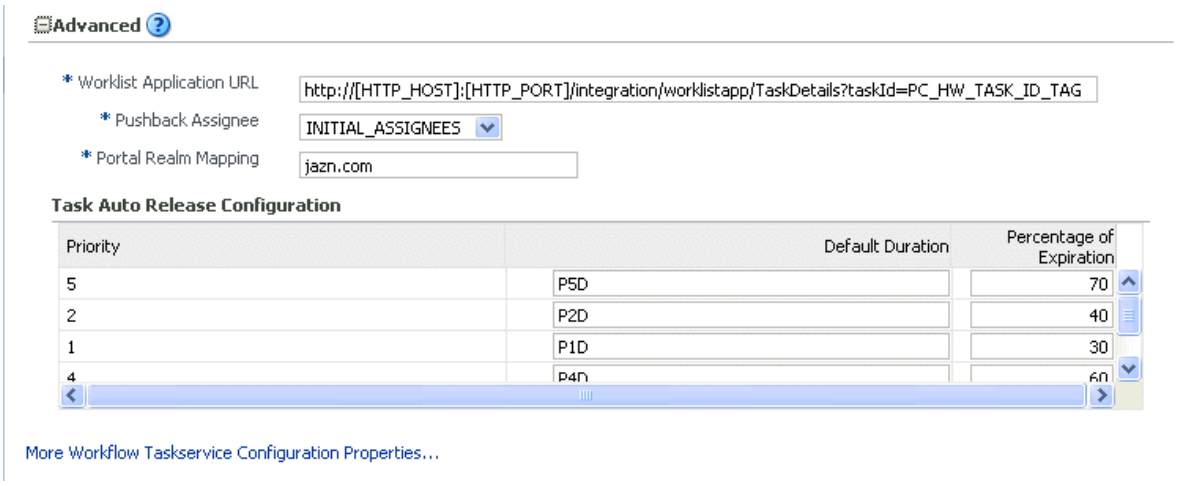
Function	Type	Description
ROUND_ROBIN	Dynamic assignment	This function picks each user or group in turn. This function uses the initialization parameter `MAX_MAP_SIZE`. This parameter specifies the maximum number of sets of users or groups for which the function can maintain `ROUND_ROBIN` counts. The dynamic assignment function holds a list of users and groups in memory for each group (or list of users and groups) on which it is asked to execute the `ROUND_ROBIN` function.
LEAST_BUSY	Dynamic assignment	This function picks the user or group with the least number of tasks currently assigned to it.
MANAGERS_MANAGER	Task escalation	This function picks the manager's manager for the task.
MOST_PRODUCTIVE	Dynamic assignment	This function picks the user or group that has completed the most tasks over a certain time period (by default, the last seven days). This function uses the initialization parameter `DEFAULT_TIME_PERIOD`. This parameter specifies the length of time (in days) over which to calculate the user's productivity. This value can be overridden when calling the `MOST_PRODUCTIVE` dynamic assignment function. Use an XPath function by specifying an alternative value as the third parameter in the XPath function call.

4. Click a function to display its parameters and values in the **Parameters** section.
5. Click **Add** to add a function. You are prompted to specify the following:
   - Function name
   - Class path
   - Function parameter name
   - Function parameter value

   You cannot add multiple properties to a function on this page. To do that, use the System MBean Browser, which is available by selecting **Administration** > **System MBean Browser** from the **SOA Infrastructure** menu.
6. Click **OK**.
7. To update the value of a parameter in a function, select the function in the **Dynamic Assignment and Task Escalation Functions** table.

   The parameter value is displayed for editing.
8. Update the value.
9. Expand the **Advanced** section.

   The **Advanced** section displays the following properties:



These properties are defined in the following table.

Properties	Description
Worklist Application URL	In the emails that are sent for tasks, the link to Oracle BPM Worklist is read from this property. This element identifies the URL. Configuring this is useful if the custom Oracle BPM Worklist is built. The tag `PC_HW_TASK_ID_TAG` in this URL is replaced with the task ID when constructing the URL for the email.
Pushback Assignee	A task can be pushed back to the previous approver or previous initial assignees. The original assignees do not need to be the approver because they may have reassigned the task, escalated the task, and so on. The possible values for this element are **INITIAL_ASSIGNEES** and **APPROVER**.
Portal Realm Mapping	This property is used when authenticating a user from an HTTP servlet request through the task query service method `createContext` (for example, when Oracle BPM Worklist runs in a single sign-on (SSO) environment). The HTTP servlet request does not carry information about the identity service realm to which the remote user belongs; this parameter is used to configure which realm to use to authenticate the user in an HTTP servlet request remote user.

Properties	Description
Task Auto Release Configuration	When a task is assigned to a group, application role, or multiple users, a user must first acquire the task before working on it. Once the task is acquired, other users cannot work on the task. If a user acquires a task, but does not act on it, the task is eventually automatically released, allowing other users to acquire the task. This prevents a user from acquiring tasks, then forgetting to work on them. This prevents others from working on them. Task automatic release enables you to configure the time period that elapses after a user acquires a task and before the system automatically releases the task and makes it available again to other users. The automatic release durations can be configured as a default duration and as a percentage of the expiration duration of a given task. The automatic release durations can also be configured differently for tasks of different priority. For example, assume the task automatic release duration for priority 2 tasks is set to 50%, with a default duration of 12 hours. If a priority 2 task is set to expire in two days, the task is automatically released after one day (which is 50% of the expiration duration). If no expiration date is set for the task, then the task is automatically released after 12 hours (which is the default automatic release duration).

10. Make changes appropriate to your environment.
11. Click **Apply**.
12. To configure advanced task service properties in the System MBean Browser, click **More Workflow Taskservice Configuration Properties**. See Step 8 of Section 20.1, "Configuring Human Workflow Notification Properties" for a list of some advanced properties that are displayed.
13. Make changes appropriate to your environment.

For more information about the task service and assignment service, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

20.3 Configuring Oracle HTTP Server for Task Form Attachments

When adding an attachment to the task form through Oracle HTTP Server (OHS), the location, `/ADFAttachmenthelper`, must be included in the OHS configuration. For example, add the following to the `mod_wl_ohs.config` file of OHS, under *`instance_home`*`/config/OHS/`*`ohs_instance`*:

```
<Location /ADFAttachmentHelper>
       SetHandler weblogic-handler
       PathTrim /weblogic
        ErrorPage  http:/WEBLOGIC_HOME:WEBLOGIC_PORT/
</Location>
```

20.4 Configuring the Pluggable Notification Service

Custom notification service implementations can be plugged in and used instead of the default notification service providers. You can plug in a custom notification service for all channels or selectively for specific channels. For example, the notification service provides the ability to plug in an existing SMS implementation instead of the default SMS notification service.

20.4.1 Pluggable Notification Service Implementation

To plug in a notification service, perform one of the following tasks:

- Implement interface `oracle.bpel.services.notification.ICustomNotificationService`
- Extend the abstract class `oracle.bpel.services.notification.AbstractCustomNotificationServiceImpl`.

This interface has methods for the following channels:

- Email
- Voice
- SMS
- Instant messaging (IM)

The plugged-in notification service can override the default providers for one or more channels. When the custom notification service is overriding the default implementation for a subset of channels, the methods corresponding to the other channels (channels that are not overridden) are not called by the notification service. Those methods can just return a null value. Alternatively, the implementation can extend the following abstract class:

```
oracle.bpel.services.notification.AbstractCustomNotificationServiceImpl
```

This class provides empty implementations for each of the channels. In that case, the implementation can just extend the methods for the appropriate channels.

The implementation and its dependent classes must be available in the class path of Oracle WebLogic Server.

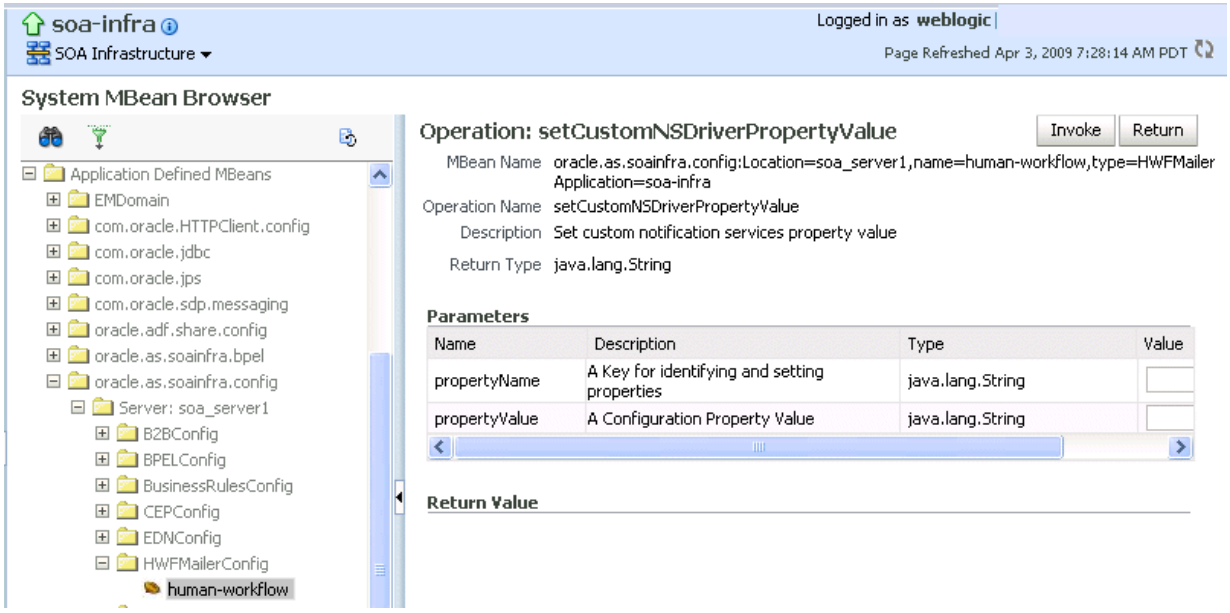
20.4.2 Pluggable Notification Service Registration

Once the implementation is available, you register it in the System MBean Browser.

To register the pluggable notification service:

1. Log in to Oracle Enterprise Manager Fusion Middleware Control.
2. In the navigator, expand the **SOA** folder.
3. Right-click **soa-infra**, and select **Administration** > **System Mbean Browser.**

   The System MBean Browser is displayed on the right side of the page.
4. Expand **Application Defined MBeans** > **oracle.as.soainfra.config** > **Server:** *server_name* > **HWFMailerConfig** > **human-workflow**.
5. Click the **CustomNSDriverPropertyNames** property on the right side of the page.
6. Record the values displayed by **CustomNSDriverPropertyNames** for the **All**, **Voice**, **Email**, **Fax**, **Pager**, **SMS**, and **IM** properties.
7. Click **Return**.
8. Click the **Operations** tab.
9. Click **setCustomNSDriverPropertyValue**.



10. In the **Value** field for **propertyName**, enter one of the values you noted down for the **All**, **Voice**, **Email**, **Fax**, **Pager**, **SMS**, and **IM** properties on the CustomNSDriverPropertyNames page. Note the following details:
    - If you are overriding the default implementation for only the email channel, use the **Email** value in the **Value** field for **propertyName** and the complete class name of your implementation in the **Value** field for **propertyValue**.
    - The override for other channels is configured the same way as the email channel.
    - Using the value of the **All** property in the **Value** field for **propertyName** refers to an implementation for all specified channels.
11. In the **Value** field for **propertyValue**, provide the complete class name of your implementation.
12. Click **Invoke**.
13. Restart Oracle WebLogic Server.

20.5 Globally Disabling the Automatic Release Timers for Oracle BPM Worklist Tasks

If automatic release timers are enabled for all Oracle BPM Worklist tasks and this is creating overhead for the database and JVM, you can globally disable the timers.

To globally disable the automatic release timers for Oracle BPM Worklist tasks:

1. Right-click **soa-infra**, and select **Administration** > **System Mbean Browser.**

   The System MBean Browser is displayed on the right side of the page.
2. Expand **Application Defined MBeans** > **oracle.as.soainfra.config** > **Server:** *server_name* > **WorkflowConfig** > **human-workflow** > **WorkflowConfig.TaskAutoReleaseConfiguration**.
3. Select the task priority to modify. Each task instance has a priority.
   - **Priority[1]**

- **Priority[2]**
- **Priority[3]**
- **Priority[4]**
- **Priority[5]**

4. In the **Attributes** tab, click **DefaultDuration**.
5. In the **Value** field, enter `P0D` to indicate zero days. The default value is `P1D` (one day).
6. Click **Apply**.
7. Perform Steps 3 through 6 for any remaining priorities for which you want to disable automatic release.

   When complete, the automatic release timers for newly created task instances with the priority you modified are disabled.

20.6 Configuring the Number of Email Notification Messages

You can control the number of email notification messages that can be processed during notification retry cycles with the System MBean Browser **RetryNotificationMessageThrottle** property. This property prevents the overloading of messages in the queue and reduces the memory size of the notification message payload.

To configure the number of email notification messages:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **SOA Administration** > **Workflow Config**	1. Right-click **soa-infra**. 2. Select **SOA Administration** > **Workflow Config**

2. Click **More Workflow Notification Configuration Properties**.
3. Click **RetryNotificationMessageThrottle**.
4. In the **Value** field, enter a value. The default is `200000` messages.
5. Click **Apply**.

20.7 Configuring Multiple Send Addresses

It may be necessary in some processes to distinguish email notification based on the from address of the email. For example, a human resources BPEL process sends emails with the from address set as `HR@yourcompany.com`, while a finance BPEL process sends emails with the from address set as `finance@yourcompany.com`.

To configure multiple send addresses:

1. Log in to Oracle Enterprise Manager Fusion Middleware Control.
2. In the navigator, expand the **SOA** folder.
3. Right-click **soa-infra**, and select **Administration** > **System Mbean Browser.**

   The System MBean Browser is displayed on the right side of the page.

4. Expand **Application Defined MBeans** > **oracle.as.soainfra.config** > **Server:** *server_name* > **HWFMailerConfig** > **human-workflow**.
5. Under the **Attributes** tab, record the value of the **ASNSDrivers** attribute. By default, only the **Default** value is available.
6. Click **Return**.
7. Click the **Operations** tab.
8. Click **setASNSDriver**.
9. For **propertyName**, enter a value (for this example, `EmailFromAddress`).
10. For **propertyValue**, enter a value (for this example, `HR@yourcompany.com`).
11. For **driverName**, enter a value (for this example, `HR`).
12. Click **Invoke**.
13. Add as many accounts as the number of from addresses needed:
    - For **propertyName**, enter a value (for this example, `EmailFromAddress`).
    - For **propertyValue**, enter a value (for this example, `finance@yourdomain.com`).
    - For **driverName**, enter a value (for this example, `Finance`).
14. Click **Invoke**.

    The **ASNSDriver** attribute now shows all the accounts created in the previous steps and the **getASNSDriverAddresses** operation now shows the addresses being used for each of the drivers.
15. Using Oracle WebLogic Server Administration Console, install multiple Oracle User Messaging Service email drivers, one for each from address. For more information, see Chapter 26, "Configuring Oracle User Messaging Service."
16. Configure the email drivers to use the required from address for sending outgoing emails.
17. In Oracle JDeveloper during design time, use `HR` as the account name to configure an email activity for an HR BPEL process and `Finance` as the account name to configure an email activity for the finance BPEL process.

20.8 Configuring Notification Retries

Oracle SOA Suite provides support for reliable notifications. The outbound notification creates a notification message with a unique notification ID and stores the message and unique ID in the dehydration store. It then enqueues this unique ID in the JMS queue and commits the transaction. A message-driven bean (MDB) listening on this queue dequeues the message and sends a notification to the user. If there is any notification failure, the notification retries three times. If the retries all fail, it marks this notification as being in error.

20.9 Configuring the Identity Service

By default, the identity service uses the embedded LDAP server in Oracle WebLogic Server as the default authentication provider. You can, however, configure Oracle WebLogic to use an alternative authentication provider, such as Oracle Internet Directory, Microsoft Active Directory, or Sun iPlanet, along with the default authenticator.

> **Note:** Although you can configure an *alternative* LDAP authentication provider for Oracle BPM Worklist, Oracle Platform Security Services does not support *multiple* LDAP authentication providers. The provider you want to use for Human Workflow authentication must be the first one listed in the order of authentication providers for Oracle SOA Suite.

This section describes how to add an authentication provider and create users and groups in the authentication provider using either Oracle WebLogic Administration Console or Oracle Directory Services Manager.

This section describes the following topics:

- Section 20.9.1, "Adding an Authentication Provider"
- Section 20.9.2, "Creating Users and Groups in the Authentication Provider"
- Section 20.9.3, "Configuring the Directory Service"

> **Note:** Oracle Fusion Middleware supports providers that enable the User and Role API to interact with custom identity stores.
>
> For more information, see Chapter "Developing with the User and Role API" of the *Oracle Fusion Middleware Application Security Guide*.

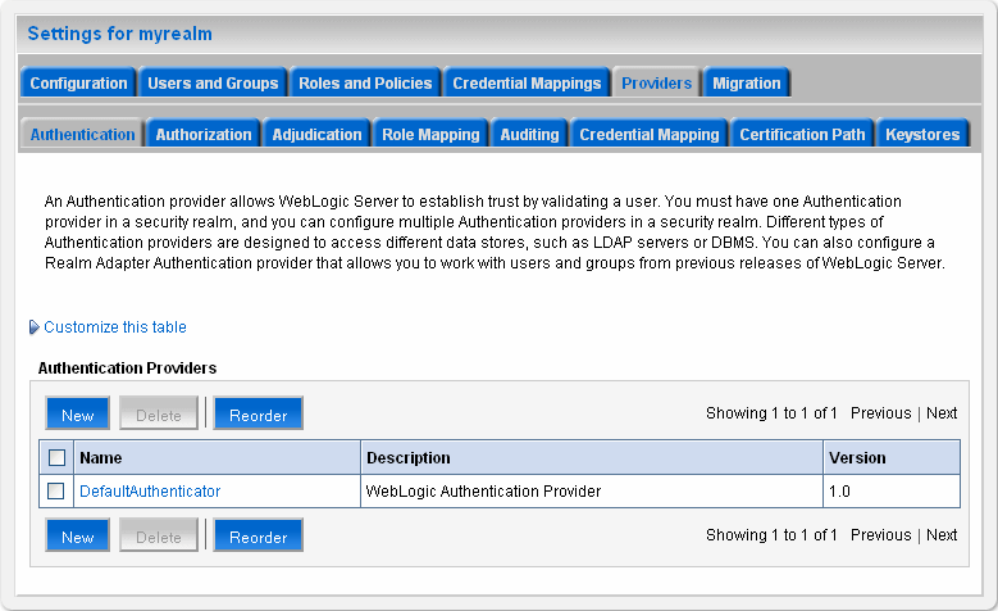
20.9.1 Adding an Authentication Provider

You can add an authentication provider to a security realm using the Oracle WebLogic Server Administration Console.

To add an authentication provider:

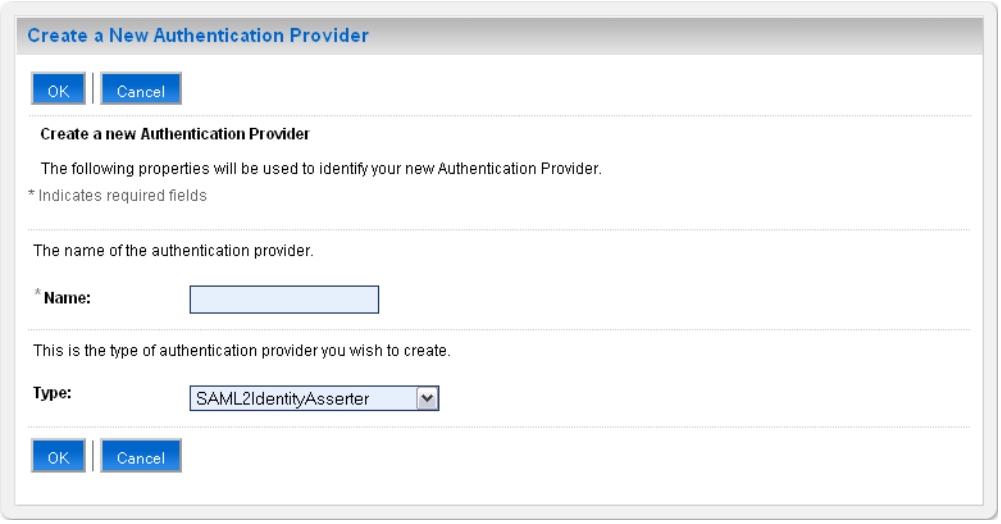
1. Log in to the Oracle WebLogic Server Administration Console.
2. Click **Security Realms** in the **Domain Structure** pane, and click the name of a realm in the list (**myrealm**, for example).
3. Click **Providers** > **Authentication**.

   The Authentication Providers page appears.



4. Click **New** to add a new authentication provider.

   The Create a New Authentication Provider page appears.



5. In the **Name** field, type a name for the provider, choose the authenticator type using the **Type** drop-down list, and click **OK**.

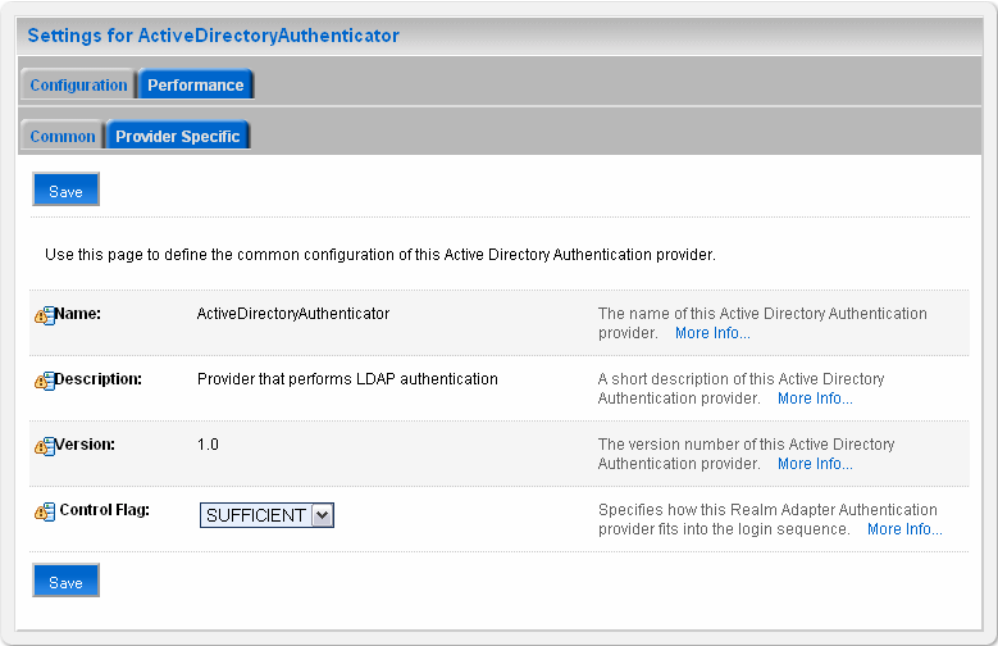
   For example, you can type `OIDAuthenticator` as the name and choose **OracleInternetDirectoryAuthenticator** as the type for a provider that authenticates users using the Oracle Internet Directory.

   Similarly, you can type a name and choose **ActiveDirectoryAuthenticator**, **iPlanetAuthenticator**, **openLDAPAuthenticator**, or **NovellAuthenticator** from the list to specify the corresponding authenticator.

> **Note:** When using Oracle Internet Directory as the authentication provider, you must set the **orclsslinteropmode** attribute to 0 (zero) using Oracle Directory Services Manager. See Section 20.9.3, "Configuring the Directory Service" for more information.

6. On the **Providers** > **Authentication** page, click the authenticator that you just created.

   The settings for the authentication provider appears.



7. From the **Control Flag** drop-down list, choose **SUFFICIENT**, and click **Save**.

   This specifies that if a user is authenticated successfully using this authenticator, WebLogic should accept the authentication and not continue to invoke any additional authenticators. If the authentication fails, Oracle WebLogic Server attempts to authenticate the user using the next authenticator in the list.

   If you set the **Control Flag** to **SUFFICIENT**, ensure that all subsequent authenticators also have the **Control Flag** set to **SUFFICIENT**. Likewise, ensure that the **Control Flag** of the default authenticator is set to **SUFFICIENT** as well.

8. Click **Provider Specific** to enter the details for the authenticator server.

9. Enter the provider-specific information about the authentication provider, check the **Use Retrieved User Name as Principal** checkbox, and click **Save**.

   You must specify the following information. Use the default setting for the rest of the fields.

Field	Description
Host	The hostname or IP address on which the authenticator server is running.
Port	The port number on which the authenticator server is running.

Field	Description
Principal	The Distinguished Name (DN) of the authenticator server user that Oracle WebLogic Server should use when connecting to the server.
Credential	The credential (usually a password) used to connect to the authenticator server.
User Base DN	The base Distinguished Name (DN) of the tree in the LDAP directory that contains users.
Group Base DN	The base Distinguished Name (DN) of the tree in the LDAP directory that contains groups.
Use Retrieved User Name as Principal	Specifies whether to use the user name retrieved from the LDAP server as the principal in the subject.
User Name Attribute	The attribute of an LDAP user object class that specifies the name of the user (for example, `UID`, `CN`, `MAIL`).

If the user attribute being used is other than the default `username` attribute in the LDAP server, add the following entries to the identity store instance in the `jps-config.xml` file:

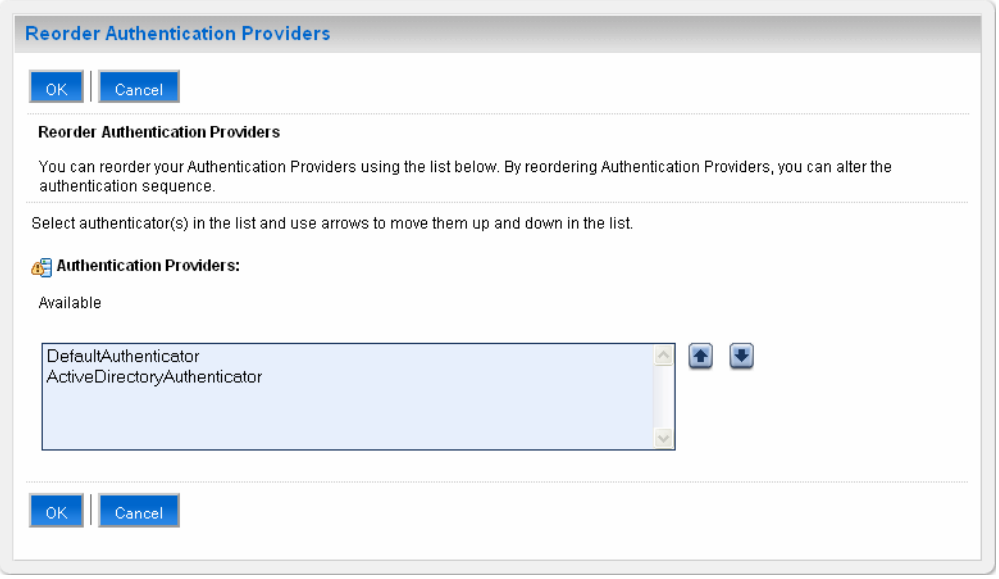
```
<property name="username.attr" value="desired_attr"/>
<property name="user.login.attr" value="desired_attr"/>
```

where *`desired_attr`* can be one of `CN`, `UID`, or `MAIL`.

10. Click **Security Realms** > **Providers** > **Authentication** to return to the list of authentication providers.

11. Click **Reorder**.

    The Reorder Authentication Providers page appears.



12. Select the new authentication provider, click the **Up** arrow to move the provider to the top of the list, and click **OK**.

    After reordering, the **DefaultAuthenticator** should appear at the bottom of the list. This action enables the system to handle logins as `weblogic` that are not typically in an LDAP directory, but still must be authenticated to start the server.

If multiple authentication providers are configured, authentication falls through the list of authenticators according to the control flags set. But the Java Portlet Specification (JPS) provides authorization against only the first entry in the list of providers.

20.9.2 Creating Users and Groups in the Authentication Provider

You can create users and groups in the authentication provider using either Oracle WebLogic Server Administration Console or Oracle Directory Services Manager.

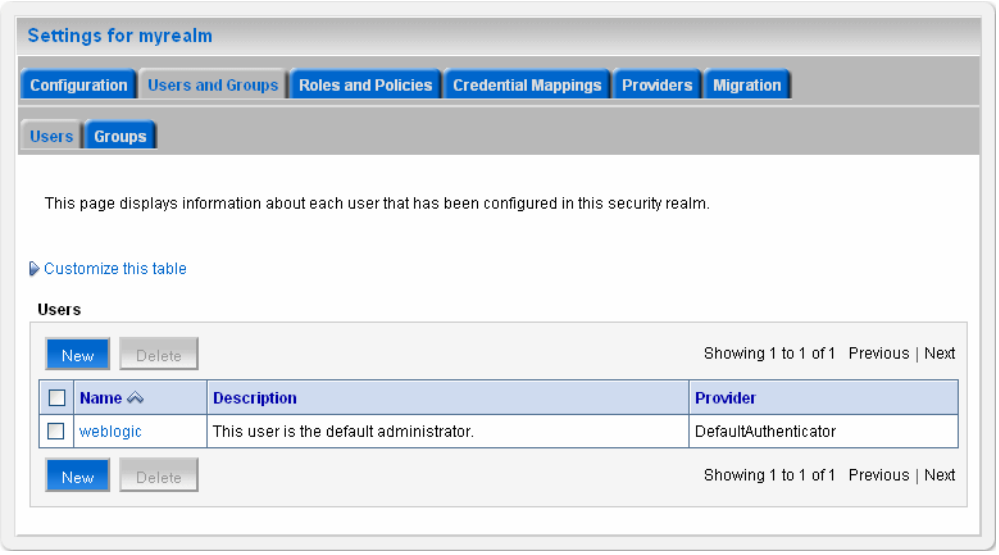
20.9.2.1 Creating Users and Groups Using WebLogic Console

You can create users and groups for a specific provider, and define user and group membership, using the Oracle WebLogic Server Administration Console.

To create a user using WebLogic Console:

1. Log in to the Oracle WebLogic Console.
2. Click **Security Realms** in the **Domain Structure** pane, and click the name of a realm in the list (**myrealm**, for example).
3. Click **Users and Groups** > **Users**.

   The Users page appears.



4. Click **New** to add a new user. The Create a New User page appears.
5. Enter the required information about the user, and click **OK**.

   You must specify the following information.

Field	Description
Name	(Required) The name of the new user.
Description	A description of the new user.
Provider	The provider for the user.
Password	The password associated with the login name for the new user.
Confirm Password	Confirmation of the password.

The system creates the new user in the specified provider and shows the Users page. You can configure group membership for the user, as required.

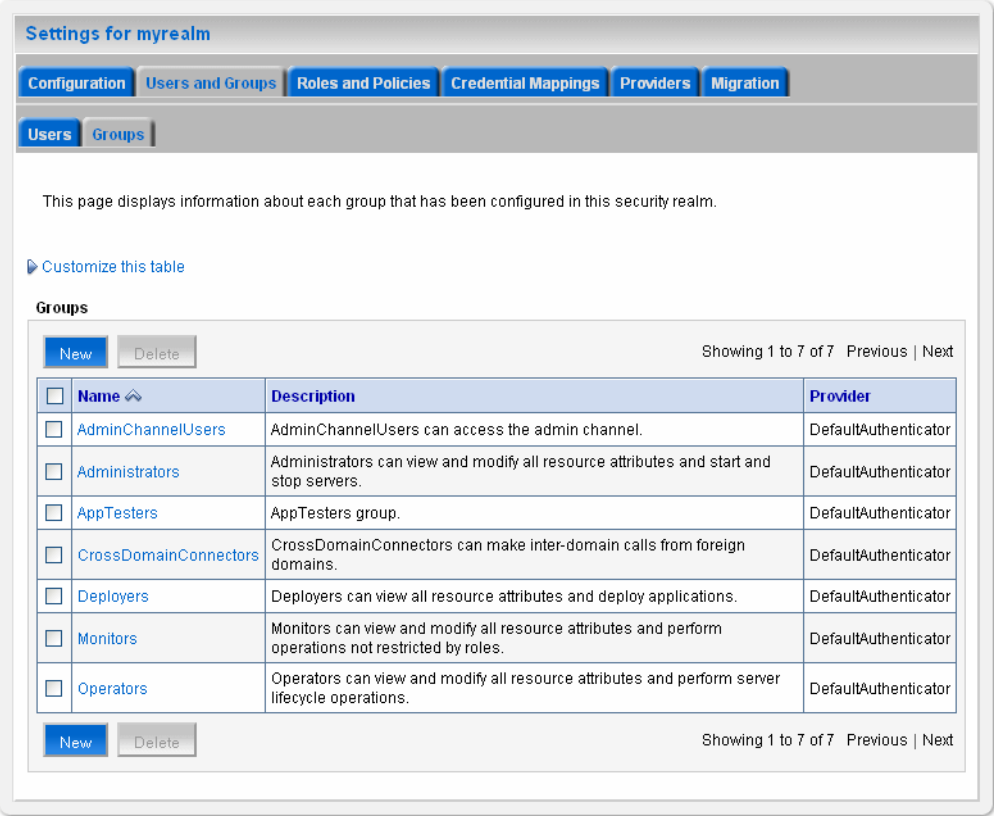
6. To specify group membership for the user, click the newly-created user in the list. The settings for the new user page appear.
7. Click **Groups** to specify group membership for the user.
8. Select a group in the **Available** list and click the right arrow to move it to the **Chosen** list.

   You can press Ctrl-Click to select multiple groups to move.
9. Click **Save**.

To create a group using WebLogic Console:

1. Click **Users and Groups** > **Groups**.

   The Groups page appears.



2. Click **New** to add a new group. The Create a New Group page appears.
3. Enter the required information about the group, and click **OK**.

   You must specify the following information.

Field	Description
Name	(Required) The name of the new group.
Description	A description of the new group.
Provider	The provider for the group.

The system creates the new group in the specified provider and shows the Groups page. You can configure group membership for the group, as required.

4. To specify group membership for the group (specify parent groups), click the newly-created group in the list. The settings for the new group page appear.
5. Click **Membership** to add the group to other groups.
6. Select a parent group in the **Available** list and click the right arrow to move it to the **Chosen** list.

   You can press Ctrl-Click to select multiple groups to move.
7. Click **Save**.

20.9.2.2 Creating Users and Groups Using Oracle Internet Directory

You can create users and groups using Oracle Internet Directory through the Oracle Directory Services Manager.

To connect to Oracle Internet Directory from the Oracle Directory Services Manager:

1. Launch the Oracle Directory Services Manager by navigating to the following URL using a web browser:

   `http://host_name:port/odsm/faces/odsm.jspx`

   where *host_name* and *port* are the hostname and the managed server port number on which Oracle Internet Directory is running.
2. Click the **Connect to a directory** link and choose **Create a New Connection** in the drop-down menu. The New Connection dialog appears.
3. Select **OID** as the directory type, enter values in the required fields, and click **Connect**.

   You can specify the following information.

Field	Description
Name	The name of the connection.
Server	(Required) The hostname or IP address of the system on which Oracle Internet Directory is running.
Port	(Required) The port number on the system on which Oracle Internet Directory is running.
SSL Enabled	Select to enable Secure Sockets Layer (SSL) communication.
User Name	(Required) The user name used to log in to Oracle Internet Directory.
Password	(Required) The password associated with the user name.
Start Page	The start page after logging into Oracle Internet Directory.

   The Oracle Directory Services Manager Home page appears.
4. Click the **Data Browser** tab. You can use this page to create and remove entries.

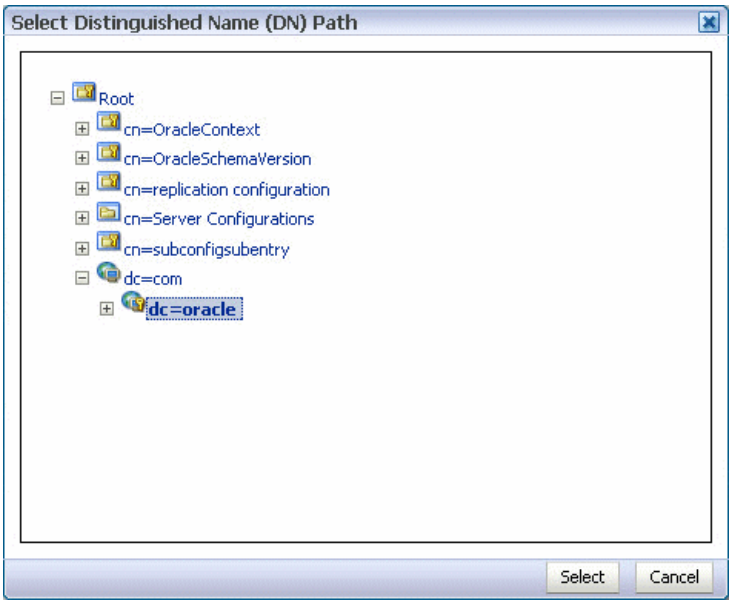
To create a domain:

1. Click the **Create a new entry** button in the **Data Tree** pane. The Entry Properties page of the Create New Entry wizard appears.

2. Click the **Add** button to add the required object class for the domain. The Add Object Class dialog appears.
3. Enter the name of the object class. When the correct object class appears in the **Name** list, select it, and click **OK**.
4. Repeat Steps 2 and 3 to add all the required object classes for the domain. Generally, **top**, **domain**, and **orclContainer** are the object classes required for a domain.

> **Note:** LDAP operations from Oracle SOA Suite can take a long time to complete if you do not index the correct LDAP attributes. The recommended searchable attribute list for indexing is **cn**, **sn**, **givenName**, **uid**, **manager**, **title**, **mail**, and **telephoneNumber**.

5. Click **Browse** to choose the parent of the domain. The Select Distinguished Name (DN) Path dialog appears.



6. Select the parent of the domain and click **Select**. You can create a hierarchy of entries by selecting the appropriate parent domains.
7. Click **Next** in the Create New Entry dialog. The Mandatory Properties page of the Create New Entry wizard appears.
8. Enter and select values for the required fields, and click **Next**.

   You can specify the following information.

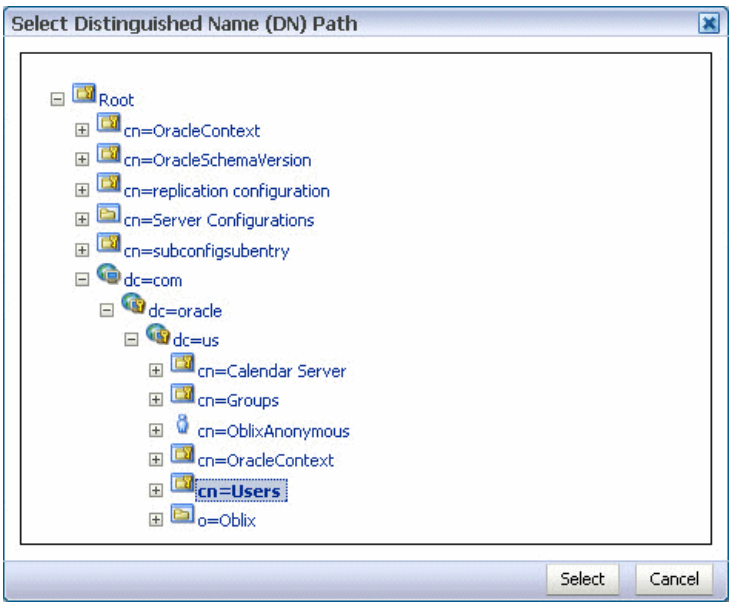
Field	Description
dc	(Required) The domain component.
Relative Distinguished Name	(Required) The relative distinguished name of the user.

   The Status page of the Create New Entry wizard appears.

9. Verify the status of the new domain, and click **Finish** to create the new domain.

To create a user:

1. Click the **Create a new entry** button in the **Data Tree** pane. The Entry Properties page of the Create New Entry wizard appears.
2. Click the **Add** button to add the required object class for the user. The Add Object Class dialog appears.
3. Enter the name of the object class. When the correct object class appears in the **Name** list, select it, and click **OK**.
4. Repeat Steps 2 and 3 to add all the required object classes for the user. Generally, **top**, **person**, **inetorgperson**, **organizationalPerson**, and **orcluser** are the object classes required for a user.
5. Click **Browse** to choose the parent of the user. The Select Distinguished Name (DN) Path dialog appears.



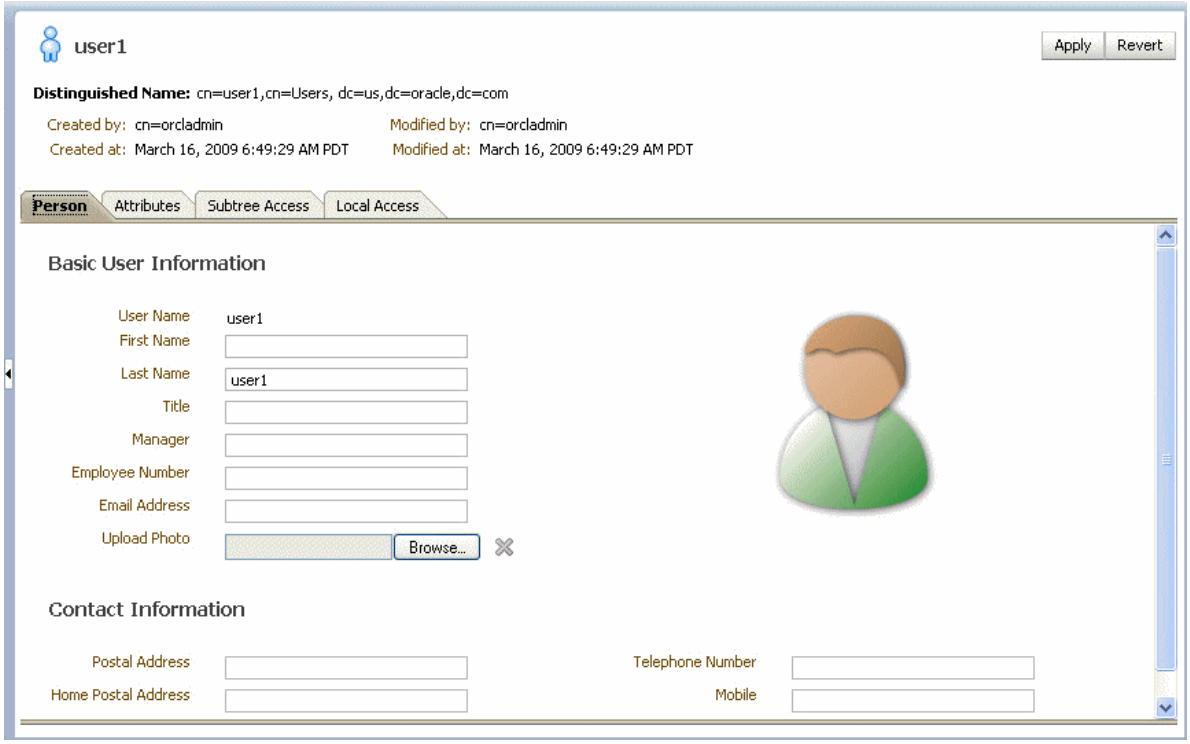
6. Select the parent of the user and click **Select**.
7. Click **Next** in the Create New Entry dialog. The Mandatory Properties page of the Create New Entry wizard appears.
8. Enter and select values for the required fields, and click **Next**.

   You can specify the following information.

Field	Description
cn	(Required) The common name.
sn	(Required) The surname (last name).
Relative Distinguished Name	(Required) The relative distinguished name of the user.

   The Status page of the Create New Entry wizard appears.

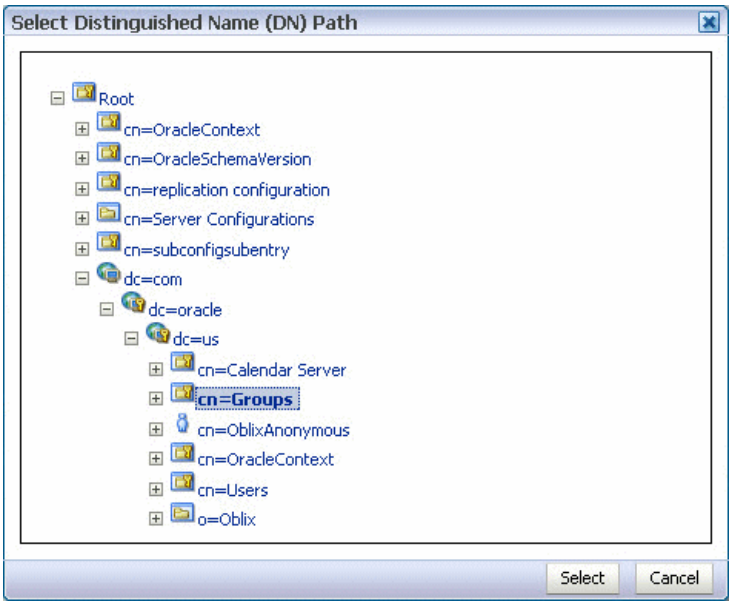
9. Verify the status of the new user, and click **Finish** to create the new user.
10. Click the entry for the newly-created user in the **Data Tree** pane. The **Person** page for the user appears.



11. Enter details about the user, and click **Apply**.

To create a group:

1. Click the **Create a new entry** button in the **Data Tree** pane. The Entry Properties page of the Create New Entry wizard appears.
2. Click the **Add** button to add the required object class for the group. The Add Object Class dialog appears.
3. Enter the name of the object class. When the correct object class appears in the **Name** list, select it, and click **OK**.
4. Repeat Steps 2 and 3 to add all the required object classes for the group. Generally, **top**, **groupOfUniqueNames**, and **orclGroup** are the object classes required for a group.
5. Click **Browse** to choose the parent of the group. The Select Distinguished Name (DN) Path dialog appears.



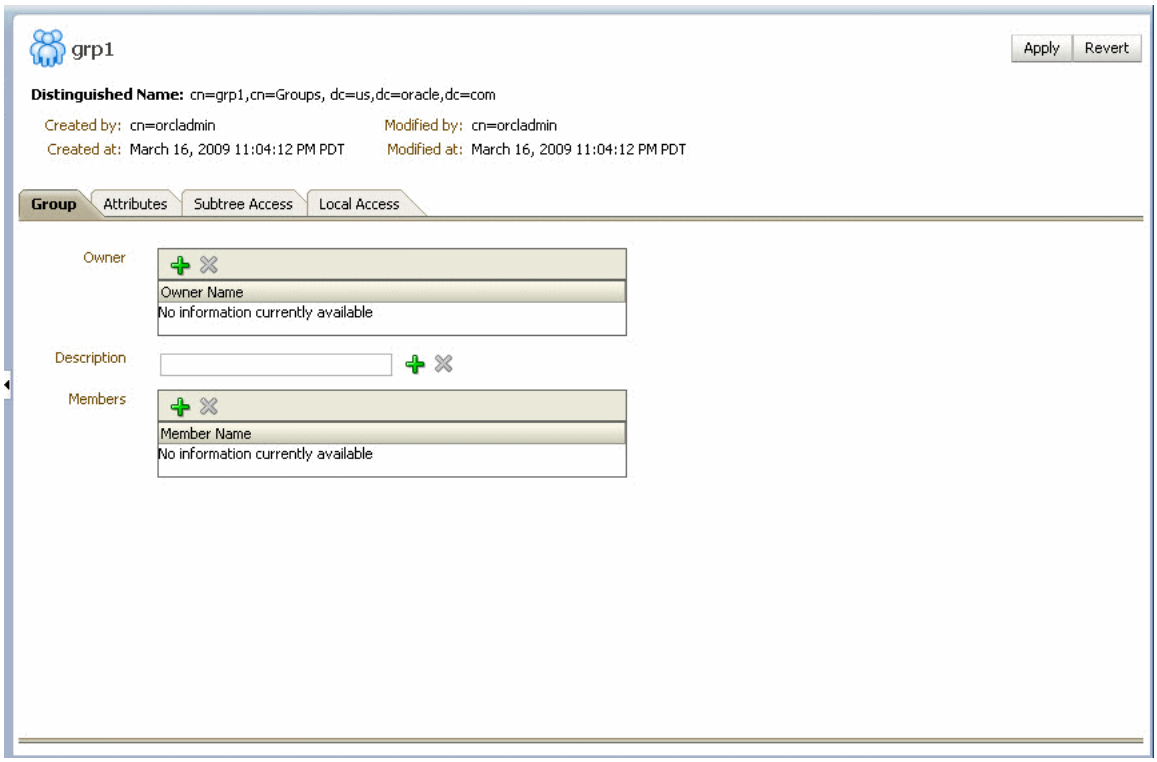
6. Select the parent of the group and click **Select**.
7. Click **Next** in the Create New Entry dialog. The Mandatory Properties page of the Create New Entry wizard appears.
8. Enter and select values for the required fields, and click **Next**.

   You can specify the following information.

Field	Description
cn	(Required) The common name.
Relative Distinguished Name	(Required) The relative distinguished name of the group.

   The Status page of the Create New Entry wizard appears.

9. Verify the status of the new group, and click **Finish** to create the new group.
10. Click the entry for the newly-created group in the **Data Tree** pane. The **Group** page for the group appears.



11. Specify details about the group, and click **Apply**.

To delete an entry:

1. Select an entry in the **Data Tree** pane.
2. Click the **Delete** this entry button in the **Data Tree** pane.

20.9.3 Configuring the Directory Service

When using Oracle Internet Directory as the authentication provider, you must set the **orclsslinteropmode** attribute to `0` (zero) using Oracle Directory Services Manager.

Note: If the `GUID` attribute in the LDAP server is set to a binary value, which cannot be properly handled in the identity service, you must map it to a *unique* attribute that exists in both the user and group object classes and cannot have a binary value. For example, if the `cn` attribute is unique, it can be used because it satisfies both of these requirements.

You map `GUID` to `cn` in the `jps-config.xml` file:

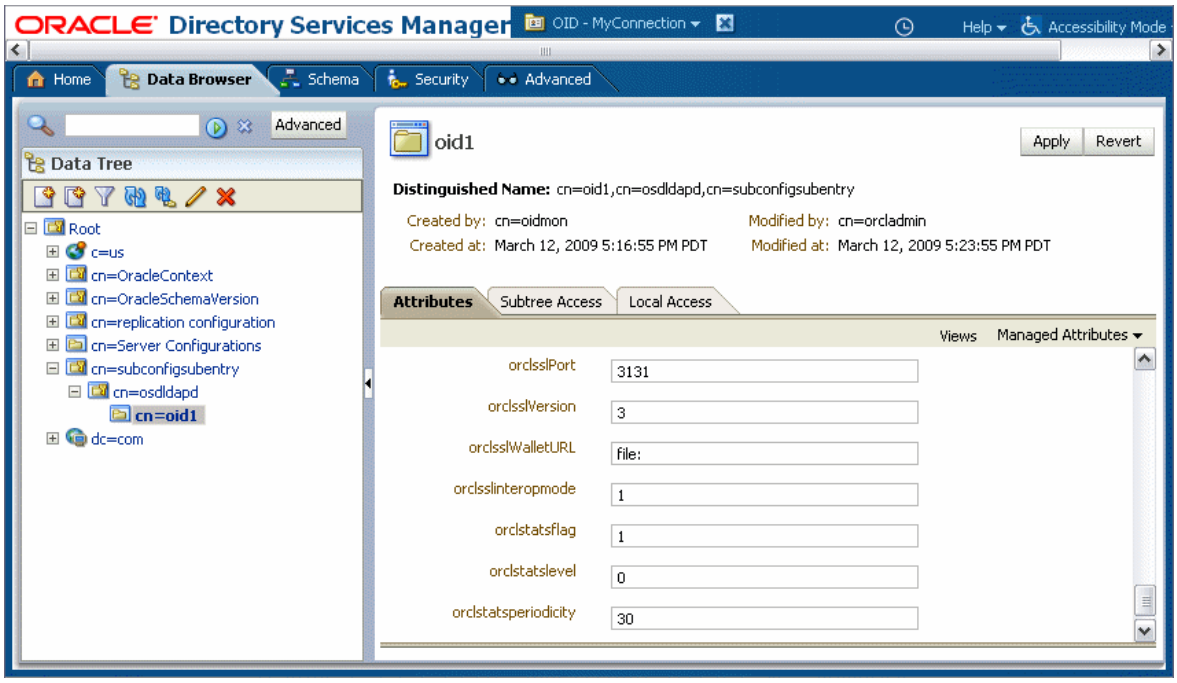
```
<property value="GUID=cn" name="PROPERTY_ATTRIBUTE_MAPPING"/>
```

For more information about identity store attribute mapping, see Chapter "Developing with the User and Role API" of the *Oracle Fusion Middleware Application Security Guide*.

To configure the directory service:

1. Launch Oracle Directory Services Manager and choose an Oracle Internet Directory connection using the drop-down list.
2. Click the **Data Browser** tab.

3. Expand the **cn=subconfigsubentry** > **cn=osdldapd** > **cn=oid1** nodes.



4. In the Attributes page, set the **orclsslinteropmode** attribute to `0`.
5. Click the **Apply** button.

20.9.4 Customizing the Identity Provider

To customize the identity provider (for example, to handle user and role information stored in home grown solutions), visit the following URL:

`http://www.oracle.com/technetwork/middleware/id-mgmt/overview/index.html`

20.10 Seeding Users, Groups, and Application Roles using LDAP Tools

This section provides an overview of the procedures required for seeding users, groups, and application roles with LDAP tools.

When you create a task, you assign humans to participate in and act upon the task. Participants can perform actions upon tasks during runtime from Oracle BPM Worklist, such as approving a vacation request, rejecting a purchase order, providing feedback on a help desk request, or some other action. There are three types of participants:

- Users
- Groups
- Application roles

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

20.10.1 Changing the Default Password in the Embedded LDAP Server

The password credential is accessible from the Oracle WebLogic Server Administration Console by selecting **Security** > **Embedded LDAP** for your domain.

For instructions on changing the default password credential, see Chapter 9, "Managing the Embedded LDAP Server" of *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

20.10.2 Seeding Users or Groups through the LDAP Browser

To seed users or groups through the LDAP browser:

1. Start an LDAP browser (for example, openLdap browser, ldapbrowser, jXplorer, and so on). See the documentation for your browser for instructions.
2. Connect to the LDAP server by providing the hostname, port number on which the server is running, and the administration user credentials with which to log in.
   - For Embedded LDAP:
     a. The default managed server port number is `7001`.
     b. The administration credential username is `cn=admin`.
     c. The administration credential password is what you set in Section 20.10.1, "Changing the Default Password in the Embedded LDAP Server."
   - For OIDm:
     a. The default port number is `3060`.
     b. The administration username is `cn=orcladmin`.
     c. The administration password is the password for the LDAP server.
3. Seed a user or group through the browser by performing the following steps:
   a. Select a parent under which to add a user or group.
   b. Select the **Edit** menu and choose an appropriate option to add a new entry.
   c. Enter all required attribute values for the entry.
4. Seed users or groups through the LDIF file by performing the following steps:
   a. Select the domain under which to seed the users or groups.
   b. Select the **LDIF** menu and choose to import an LDIF file.
   c. In the Import LDIF File dialog, browse for and select the LDIF file and click **Import**.

      Similarly, the users or groups seeded on the LDAP server can be exported to an LDIF file by selecting the **Export** option from the **LDIF** menu.
5. Add attributes to the users or groups by performing the following steps:
   a. Select an entry for which to add a new attribute.
   b. Right-click and choose the option to add a new attribute.
   c. In the Add Attribute dialog, provide the name and value of the attribute.

      You can only add attributes that are defined in the LDAP server schema.
6. Delete attributes for users or groups by performing the following steps:
   a. Select an entry for which to delete a new attribute.

b. Select an attribute from the list of attributes and delete it.

20.10.3 Seeding Application Roles using WLST Scripts

For instructions on using the WebLogic Scripting Tool (WLST) to seed application roles, see Chapter 4, "Infrastructure Security Custom WLST Commands" of *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

20.10.4 Managing Application Roles in Oracle Enterprise Manager Fusion Middleware Control

This section describes how to manage application roles in Oracle Enterprise Manager Fusion Middleware Control.

Note: Follow these steps to provide nonadministrators with access to Oracle SOA Composer. This is accomplished by assigning the **SOADesigner** role to users or groups on the Edit Application Role page. The users must exist in the Oracle WebLogic Server realm.

To manage application roles in Oracle Enterprise Manager Fusion Middleware Control:

1. In the navigator, select the appropriate Oracle WebLogic Server under **WebLogic Domain** > ***Farm_Domain_name***.
2. Right-click the domain name, and select **Security** > **Application Roles**.
3. Create an application role by performing the following steps:
   a. Select the **Create** option in the Application Roles page.

      The Create Application Role page appears.

   b. In the **Application** list, select the application name (***server_name***/**soa-infra**) under which to create a role.
   c. Enter the role name, display name, and description for the application role.
   d. Add members by selecting **Add Role** in the **Roles** section and **Add User** in the **Users** section.
   e. Click **OK** to create the application role.
4. Edit application roles by performing the following steps:
   a. In the **Select Application Name to Search** list of the **Search** section of the Application Roles page, select an appropriate application (for example, **soa_server1/soa-infra**).
   b. To the right of the **Role Name** list, click the **Search** icon.

      This action lists all the application roles created for that application.

   c. Select the application role to edit (for example, select **SOADesigner**).
   d. Click **Edit**.

      The Edit Application Role page appears.

   e. Add application roles and groups in the **Roles** section and users in the **Users** section (for example, assign **SOADesigner** to a user to which to provide access

to Oracle SOA Composer). The user must be defined in the Oracle WebLogic Server realm.

f. Click **OK**.

5. Delete application roles by performing the following steps:

   a. In the **Select Application Name to Search** list of the **Search** section of the Application Roles page, select an appropriate application.

   b. To the right of the **Role Name** list, click the **Search** icon.

      This action lists all the application roles created for that application.

   c. Select the application role to delete.

   d. Click the **Delete** button to delete the application role.

   e. Click **Yes** in the Confirmation dialog.

20.11 Enabling Case Agnostic Group Names in Human Tasks

By default, only user names in human tasks are case agnostic (case insensitive). This behavior is controlled by the value of the **caseSensitive** property in the System MBeans Browser for users, which is set to `false` by default. Group names in human tasks must be identical to what is seeded in the user directory. However, if you also want group names in human tasks to be case agnostic, you must set the **caseSensitiveGroups** property to `false`.

To enable case agnostic behavior for group names in human tasks:

1. Right-click **soa-infra**, and select **Administration** > **System Mbean Browser.**

   The System MBean Browser is displayed on the right side of the page.

2. Expand **Application Defined MBeans** > **oracle.as.soainfra.config** > **Server:** *server_name* > **WorkflowIdentityConfig** > **human-workflow** > **WorkflowIdentityConfig.PropertyType** > **caseSensitiveGroups**.

3. Click the **Operations** tab.

4. Click **setValue**.

5. In the **Value** field, enter `false`.

6. Click **Invoke**.

20.12 Configuring Security Policies for Human Workflow Web Services

A policy set, which can contain multiple policy references, enables you to attach policies globally to a range of endpoints of the same type. Attaching policies globally using policy sets enables you to ensure that all subjects are secured in situations in which multiple users, such as a developer, assembler, or deployer, did not explicitly specify the policies to attach. Policies that are attached using a policy set are considered externally attached.

For example, if the developer did not specify policies in annotations or include policy references in deployment descriptors, then the deployer must attach them or risk a potential security risk. By attaching policies globally to a set of subjects by type, the administrator can ensure that all subjects are secured by default independent of, and even before, deployment. For example, the administrator can define a policy set that attaches a security policy to all web service endpoints in a domain. In this case, any

new services added to the domain automatically inherit the security configuration defined in the policy set.

For more information about attaching policies globally using policy sets, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

21

Monitoring Human Workflow Service Components and Engines

This chapter describes how to monitor human task service components and the human workflow service engine, including monitoring workflow service component and service engine instances and faults, viewing task status, monitoring active requests and operation statistics, and monitoring deployed human workflows.

This chapter includes the following topics:

- Section 21.1, "Monitoring Recent Human Task Service Component Instances and Faults"
- Section 21.2, "Viewing the Status of Human Workflow Tasks"
- Section 21.3, "Monitoring Human Task Service Component Instances"
- Section 21.4, "Monitoring Human Workflow Service Engine Recent Instances and Faults"
- Section 21.5, "Monitoring Human Workflow Service Engine Active Requests and Operation Statistics"
- Section 21.6, "Monitoring Human Workflow Service Engine Instances"
- Section 21.7, "Monitoring Deployed Human Workflows in the Service Engine"

For more information, see the following sections:

- Section 1.2.4, "Introduction to Service Components and Service Component Instances"
- Section 1.2.6, "Introduction to Service Engines"

21.1 Monitoring Recent Human Task Service Component Instances and Faults

You can monitor recent instances and faults for human task service components. Each service component in a SOA composite application has its own instance ID. These IDs are different from the overall instance ID of the SOA composite application of which each service component is a part.

To monitor human task service component instances and faults:

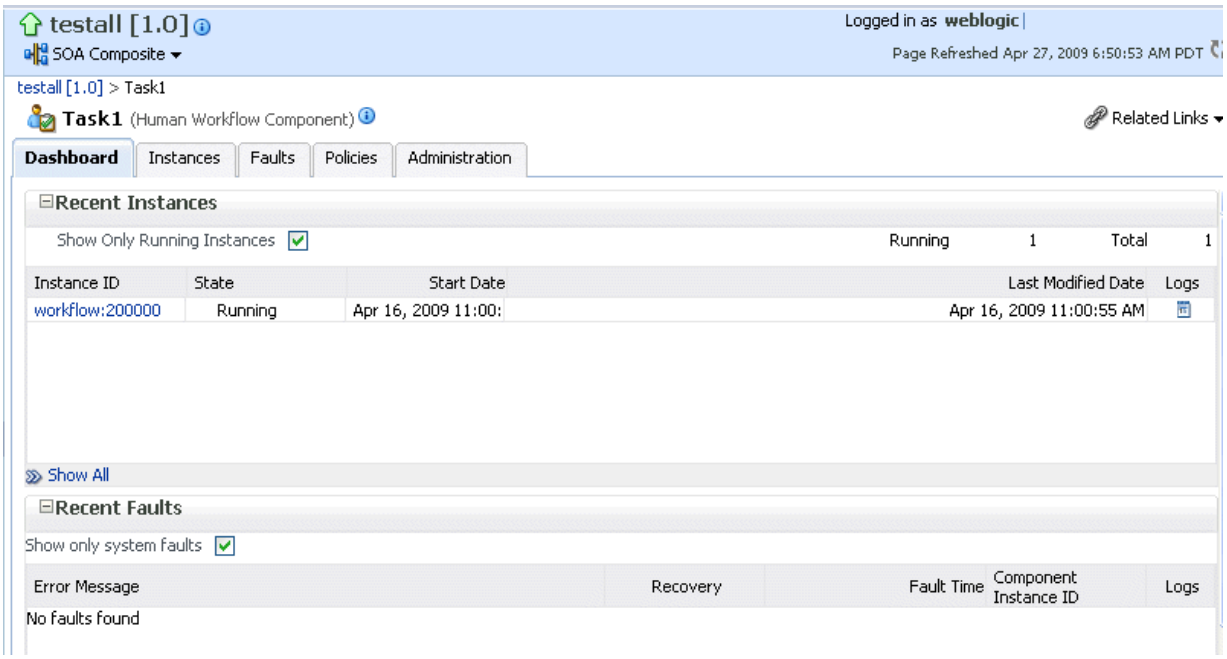
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, select a specific SOA composite application.

2. Select the human task service component in the **Component Metrics** table.

   The upper part of the Dashboard page displays the following details:

   - Recent instances of the human task service component, including the instance ID of the service component, the state of the instance (for example, completed or running), the instance start time, the last modification time, and links to log files describing the instances.
   - Recent faults in the human task service component, including the error message, whether the fault is recoverable, the time at which the fault occurred, the instance ID of the human task service component, and links to log files describing the faults.



3. In the **Recent Instances** section, perform the following tasks:

   a. In the **Instance ID** column, click an instance ID for a service component to monitor the current status of a task on which approval actions are being taken.

   b. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

   c. Click **Show All** below the section to access the Instances page of the service component.

4. In the **Recent Faults** section, perform the following tasks:

   a. In the **Error Message** column, click an error message to display complete information about the fault. If the fault is identified as recoverable, click the **Recover Now** link to perform fault recovery.

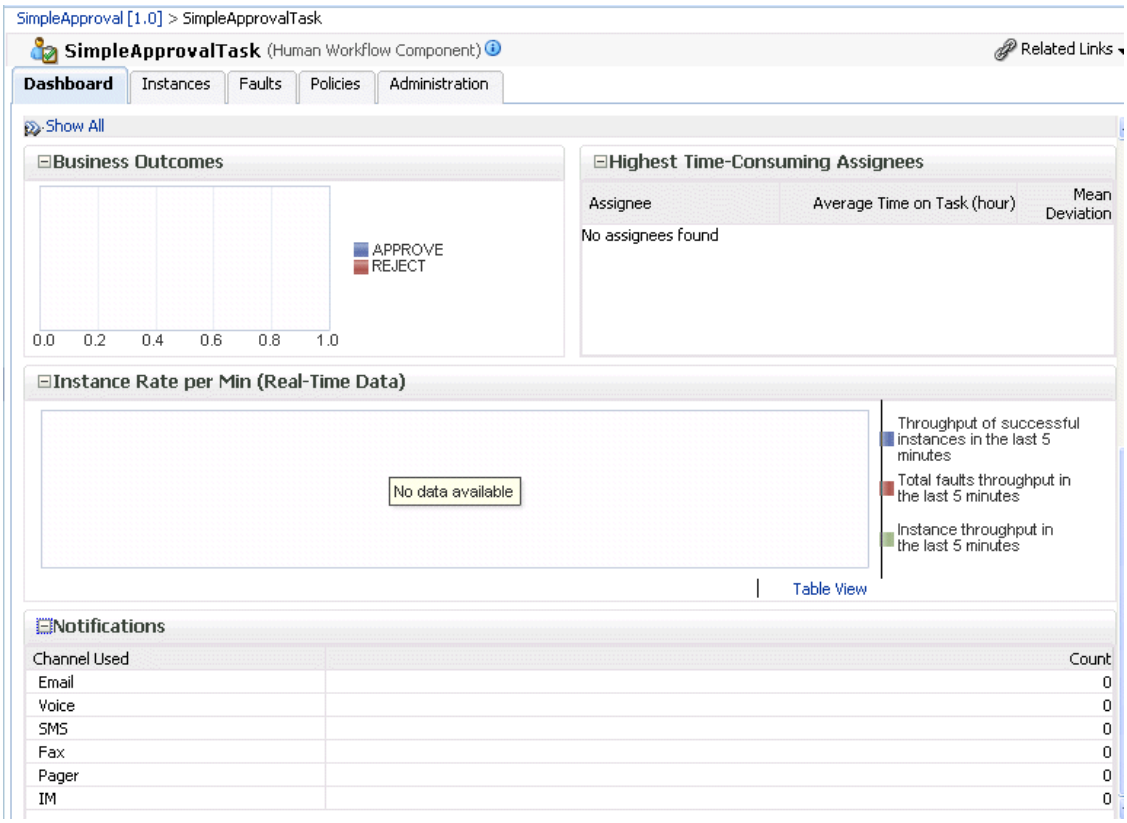
b. In the **Recovery** column, click a fault identified as **Recoverable** to perform fault recovery at the component instance level.

c. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

d. Click **Show All** below the section to access the Faults page of the service component.

The lower part of the Dashboard page displays the following details:

- A graphical representation of business outcomes.
- Assignees that take the longest time to respond to and act upon tasks.
- The number of successful, faulted, and instantiated instances processed per minute. Click **Table View** to display throughput details for the last five minutes, including the throughput for successful instances, the total faults throughput, and the instance throughput.
- The notification channels used for task approval in the human task service component.



For more information, see Section 1.2.3, "Introduction to SOA Composite Application Instances" and *Oracle Fusion Middleware Administrator's Guide* for details about viewing and searching log files.

21.2 Viewing the Status of Human Workflow Tasks

You can monitor the current status of human workflow tasks being acted upon by participants.

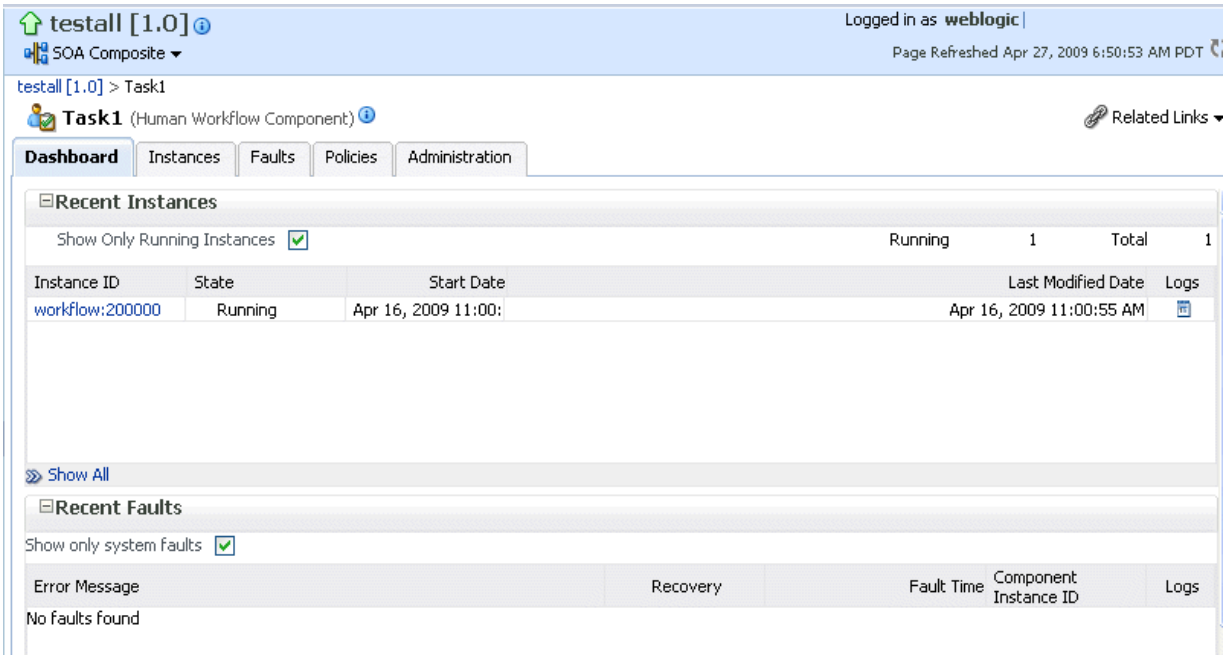
To view the status of human workflow tasks:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, select a specific SOA composite application.

2. Select the human task service component in the **Component Metrics** table.

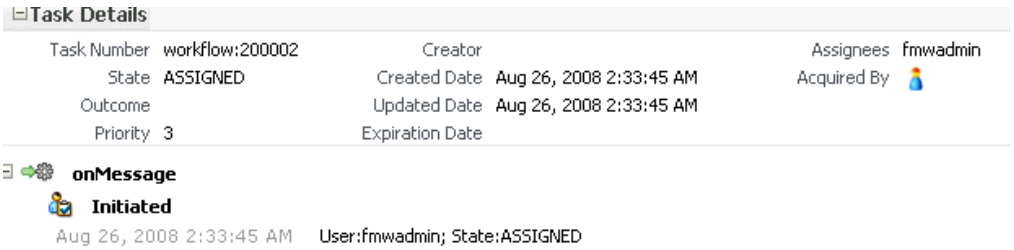
   The Dashboard page appears.



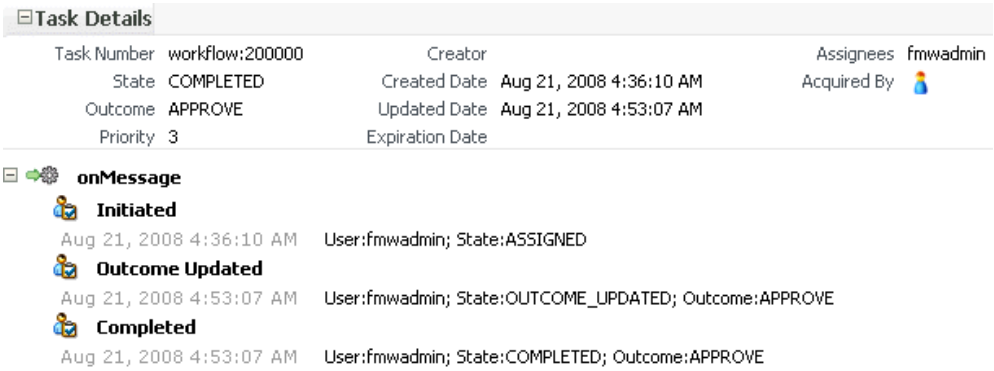
3. In the **Instance ID** column, click an instance ID of a service component to monitor the current status of a task on which approval actions are being taken.

   The Task Details page displays the following details:

   - Task details about the selected service component instance ID (task number), including the current state (for example, assigned), task outcome (if completed), task priority, creation date, updated date, expiration date, and task assignee.
   - A flow of the current status of a task (for example, assigned or approved).



   As the task is acted upon by participants, the flow is updated.



4. In the upper right-hand corner of the page, click **Worklist Application** to access the login page for Oracle BPM Worklist.

For more information, see Section 1.2.3, "Introduction to SOA Composite Application Instances."

21.3 Monitoring Human Task Service Component Instances

You can monitor human task service component instances. Each service component has its own unique instance ID. This ID is in addition to the instance ID of the overall SOA composite application of which this service component is a part.

> **Note:** Human workflow invocations from the BPEL service engine use different transactions than BPEL processes. Therefore, if a BPEL transaction is rolled back for any reason, the workflow task instances are still created.

To monitor human task service component instances:

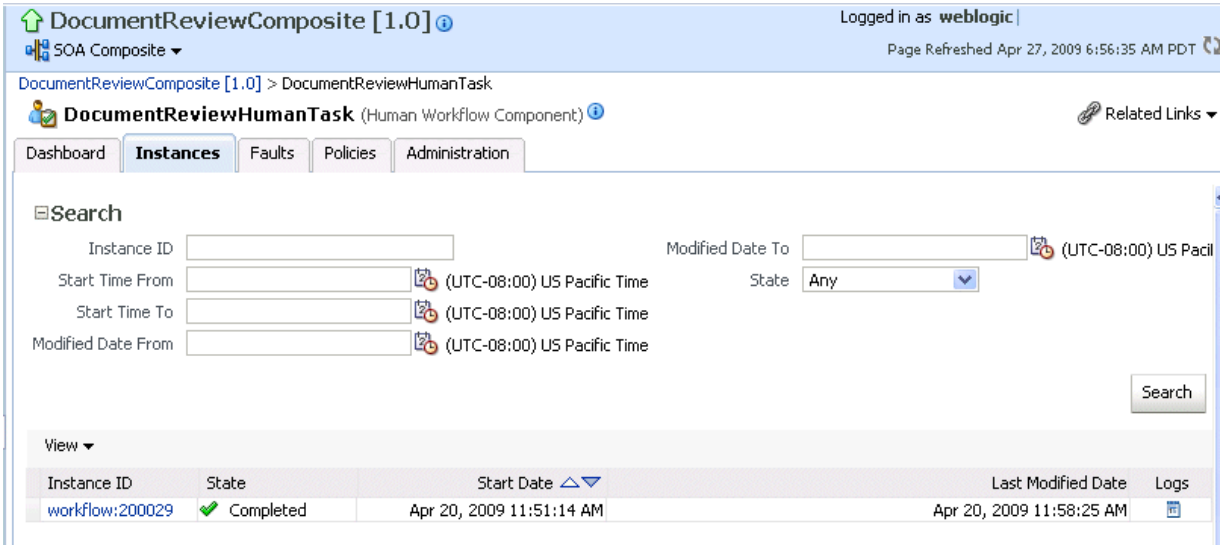
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, select a specific SOA composite application.

2. Select the human task service component in the **Component Metrics** table.
3. Click **Instances**.

   The Instances page displays the following details:

   - A utility for searching for a specific human task service component instance by specifying criteria and clicking **Search**.
   - Instances, including the instance ID of the service component, the state of the instance (for example, completed successfully or faulted), the instance start time, the last modification time, and links to log files describing the instance.



4. In the **Instance ID** column, click a specific ID for a service component to monitor the current status of a task on which approval actions are being taken.
5. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

For more information, see Section 1.2.3, "Introduction to SOA Composite Application Instances" and *Oracle Fusion Middleware Administrator's Guide* for details about viewing and searching log files.

21.4 Monitoring Human Workflow Service Engine Recent Instances and Faults

You can monitor instances and faults of all human task service component instances running in the human workflow service engine. These human task service components can be part of separate SOA composite applications.

To monitor human workflow service engine instances and faults:

1. Access this page through one of the following options:

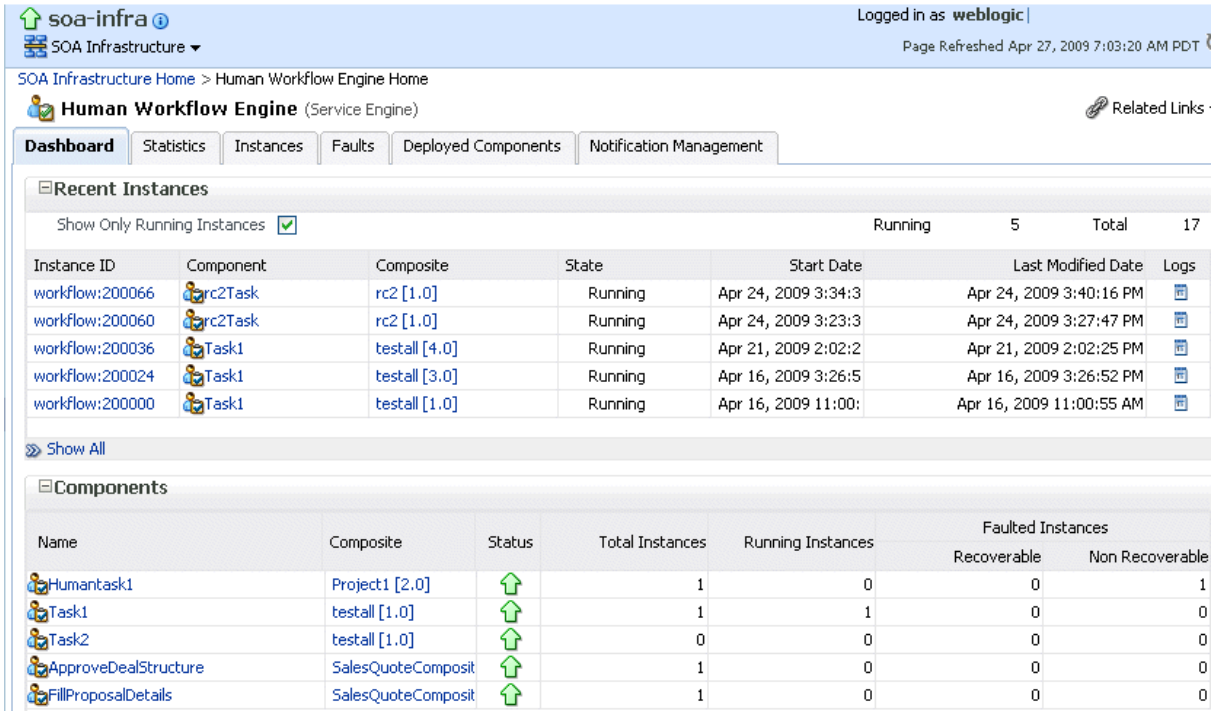
From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **Human Workflow**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **Human Workflow**.

2. Click **Dashboard**.

   The upper part of the Dashboard page displays the following details:

   - Recent instances of all human task service components running in the human workflow service engine, including the instance ID of the service component, the service component, the SOA composite application of which the service component is a part, the state of the instance (for example, completed, running, or faulted), the instance start time, and the last modification time.

- Service components, including the service component name, SOA composite application, state of the service component, and total, running, and faulted instances.

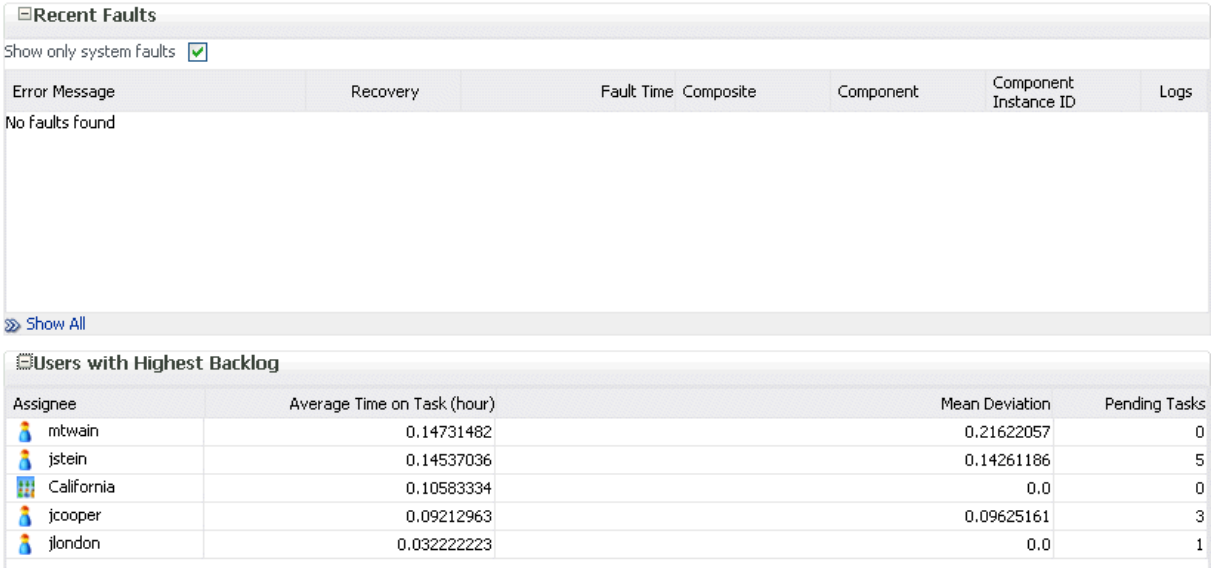


3. In the **Recent Instances** section, perform the following tasks:
   a. In the **Instance ID** column, click an instance ID for a service component to monitor the current status of a task on which approval actions are being taken.
   b. In the **Component** column, click a specific service component to access its home page.
   c. In the **Composite** column, click a specific SOA composite application to access its home page.
   d. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.
   e. Click **Show All** below the section to access the Instances page of the service engine.
4. In the **Components** section, perform the following tasks:
   a. In the **Name** column, click a specific service component to access its home page.
   b. In the **Composite** columns, click a specific SOA composite application to access its home page.
   c. Click **Show All** below the section to access the Deployed Components page of the service engine.

   The lower part of the Dashboard page displays the following details:

   - Recent faults, including the error message, the time at which the fault occurred, the SOA composite application, the service component, and the service component instance ID.

- Task assignees who take the longest time to act upon a task and who have the highest backlog of pending tasks to which to respond.



5. In the **Recent Faults** section, perform the following additional tasks:

   a. In the **Error Message** column, click an error message to display complete information about the fault. If the fault is identified as recoverable, click the **Recover Now** link to perform fault recovery. Clicking this link invokes the human workflow audit trail page for the instance. The audit trail page has a link to Oracle BPM Worklist called **Go to Worklist Application**, where you can go to recover from the fault. This link does not take you directly to the fault; you must manually locate the fault.

   b. In the **Recovery** column, click a fault identified as **Recoverable** to perform fault recovery at the component instance level.

   c. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

   d. Click **Show All** below the section to access the Faults page of the service engine.

21.5 Monitoring Human Workflow Service Engine Active Requests and Operation Statistics

You can view details about active requests in the human workflow service engine and operational statistics. such as service used, operations performed, and active and completed requests.

To monitor human workflow service engine active requests and operation statistics:

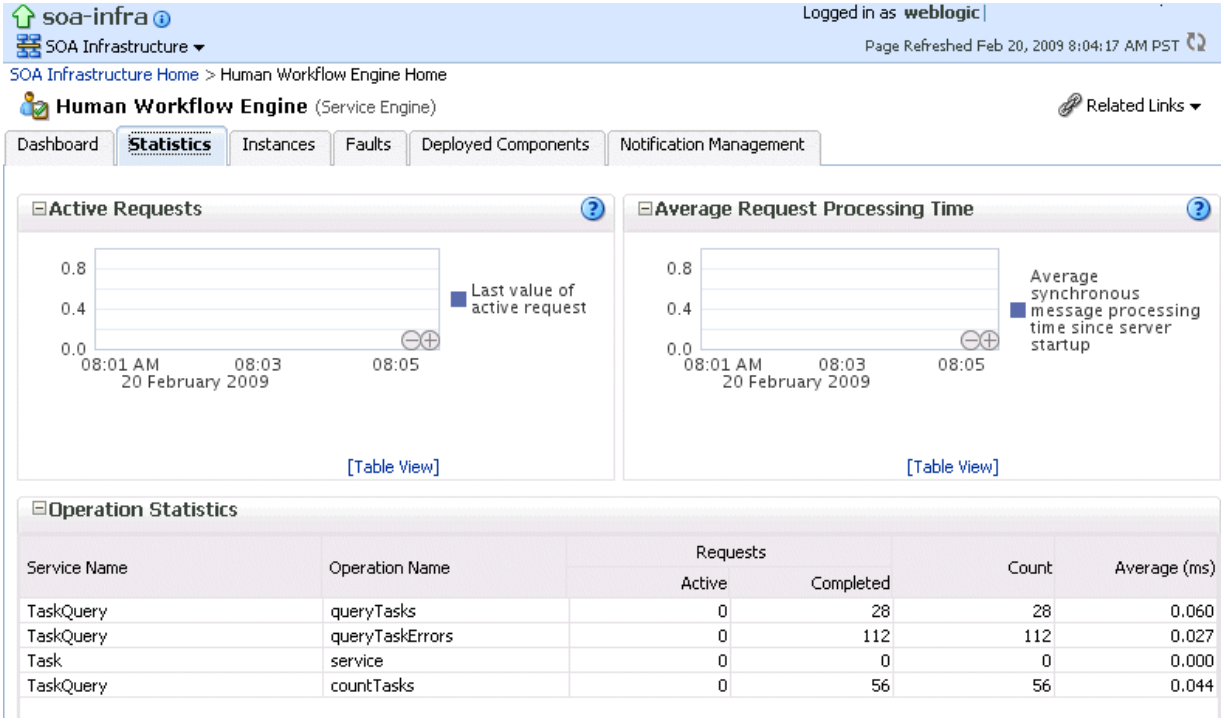
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **Human Workflow**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **Human Workflow**.

2. Click **Statistics**.

   The Statistics page displays the following details:

   - Active requests in the service engine. Use this graph to get an idea of the current service engine load. Only under extreme load conditions is there data shown in the graph. This is because most requests are processed instantaneously by the service engine. The data is collected by a Dynamic Monitoring Service (DMS) state sensor. Once the requests are processed by the service engine, the count goes to zero. This action enables you to know the current load on the service engine (for example, if it is too high).
   - Average request message processing time in the service engine since the last startup of the SOA Infrastructure. Use this graph to check service engine performance. While the processing time is calculated based on the last startup of the SOA Infrastructure, the data that is displayed in the graph is gathered only from the time at which you first accessed this page. The graph does not continue calculating and displaying data if you have not accessed this page. The DMS phase event sensor calculates the average request processing time and provides the processing time data.
   - Operation statistics about human workflow services used in the service engine, including the human workflow service used, the operation performed by the service, the number of active and completed requests, the count, and the average processing time.



For more information, see the following documentation:

- Section 1.2.4, "Introduction to Service Components and Service Component Instances"
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite* for details about human workflow services and operations

21.6 Monitoring Human Workflow Service Engine Instances

You can monitor all human task service component instances running in the service engine. These human task service components can be part of separate SOA composite applications.

To monitor human workflow service engine instances:

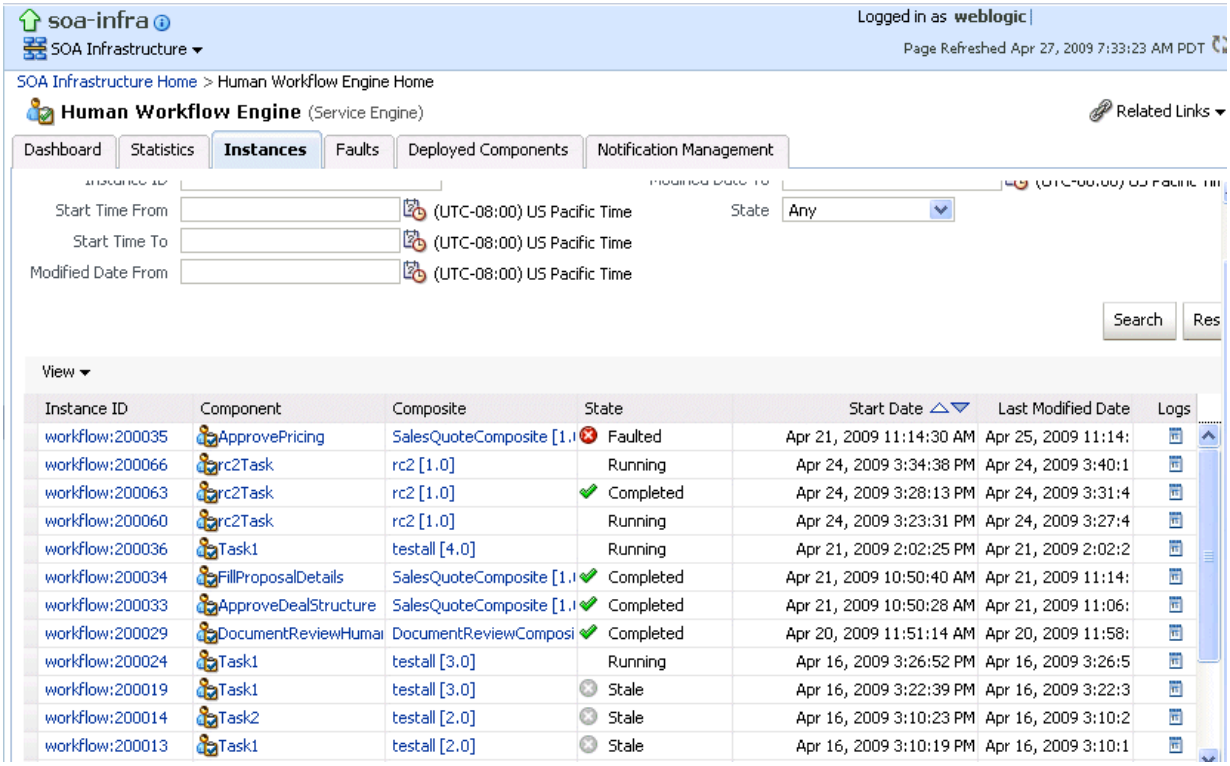
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **Human Workflow**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **Human Workflow**.

2. Click **Instances**.

   The Instances page displays the following details:

   - A utility for searching for a specific instance by specifying criteria and clicking **Search**.
   - Instances, including the instance ID of the service component, the service component, the SOA composite application of which the service component is a part, the state of the instance (for example, completed, running, or faulted), the instance start time, the last modification time, and links to log files that describe the instance.



3. In the **Instances** section, perform the following additional tasks:

   a. In the **Instance ID** column, click an instance ID for a service component to monitor the current status of a task on which approval actions are being taken.

b. In the **Component** column, click a specific service component to access its home page.

c. In the **Composite** column, click a specific SOA composite application to access its home page.

d. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

For more information, see Section 1.2.3, "Introduction to SOA Composite Application Instances" and *Oracle Fusion Middleware Administrator's Guide* for details about viewing and searching log files.

21.7 Monitoring Deployed Human Workflows in the Service Engine

You can monitor all deployed SOA composite applications with human task service components running in the service engine.

To monitor deployed human workflows in service engines:

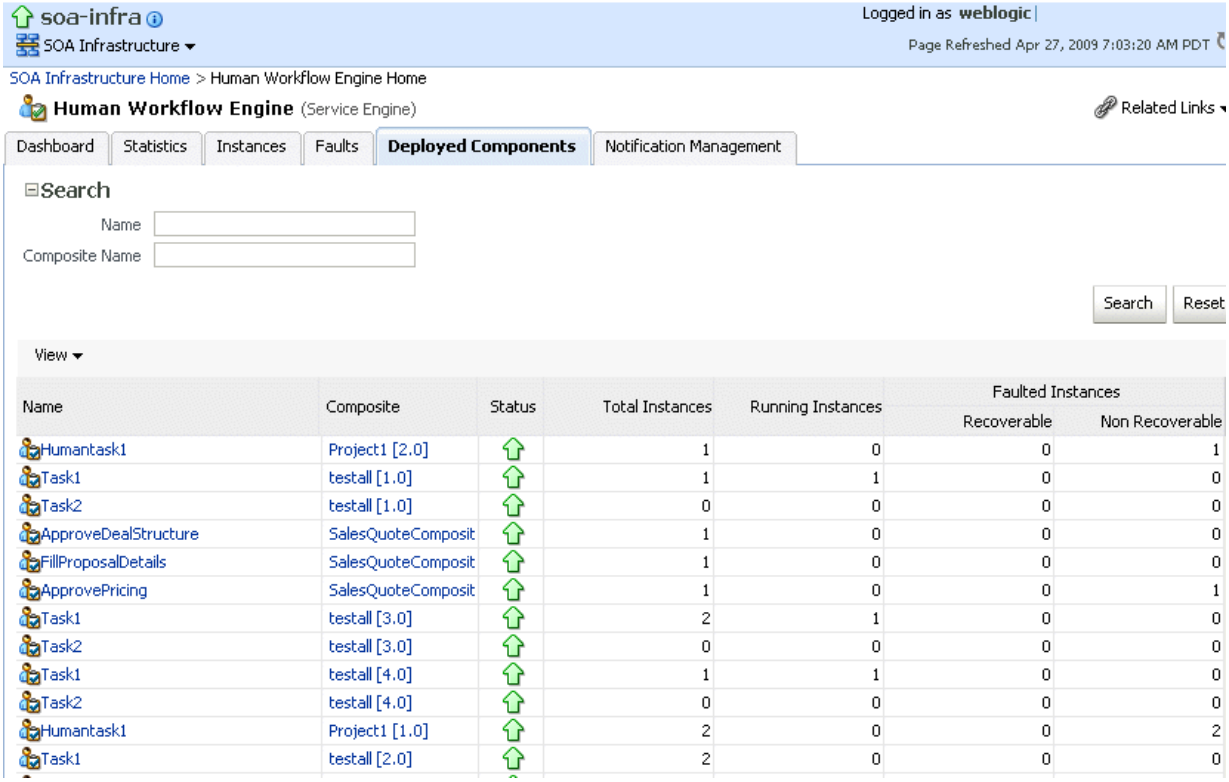
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **Human Workflow**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **Human Workflow**.

2. Click **Deployed Components**.

   The Deployed Components page displays the following details:

   - A utility for searching for a specific deployed SOA composite application by specifying the full name and clicking **Search**.
   - Details about deployed human task service components running in this service engine, including the service component name, the SOA composite application, the current status, and the number of total, faulted, and running instances.



3. In the **Name** column, click a specific service component to access its home page.
4. In the **Composite** column, click a specific SOA composite application to access its home page.

22

Managing Human Workflow Service Components and Engines

This chapter describes how to manage human task service components and the human workflow service engine, including managing policies, recovering from workflow faults, managing the task details application URI, managing outgoing and incoming email notifications, and moving workflow data from test to production environments.

This chapter includes the following topics:

- Section 22.1, "Managing Human Task Service Component Policies"
- Section 22.2, "Recovering from Human Workflow Service Engine Faults"
- Section 22.3, "Managing the URI of the Human Task Service Component Task Details Application"
- Section 22.4, "Recovering from Human Task Service Component Faults"
- Section 22.5, "Managing Outgoing Notifications and Incoming Email Notifications"
- Section 22.6, "Moving Human Workflow Data from a Test to a Production Environment"

> **Note:** Human task service components are also known as human workflow service components in Oracle Enterprise Manager Fusion Middleware Control.

For more information, see the following sections:

- Section 1.2.4, "Introduction to Service Components and Service Component Instances"
- Section 1.2.6, "Introduction to Service Engines"

22.1 Managing Human Task Service Component Policies

You can attach and detach security policies to and from human task service components of currently deployed SOA composite applications. Policies apply security to the delivery of messages. Oracle Fusion Middleware uses a policy-based model to manage web services.

Notes:

- Before attaching policies, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for definitions of available policies and details about which ones to use in your environment.
- Human tasks have a port that is protected by default using the SAML policy `oracle/wss10_saml_token_service_policy`. Oracle recommends that you *not* use this policy in a production environment.

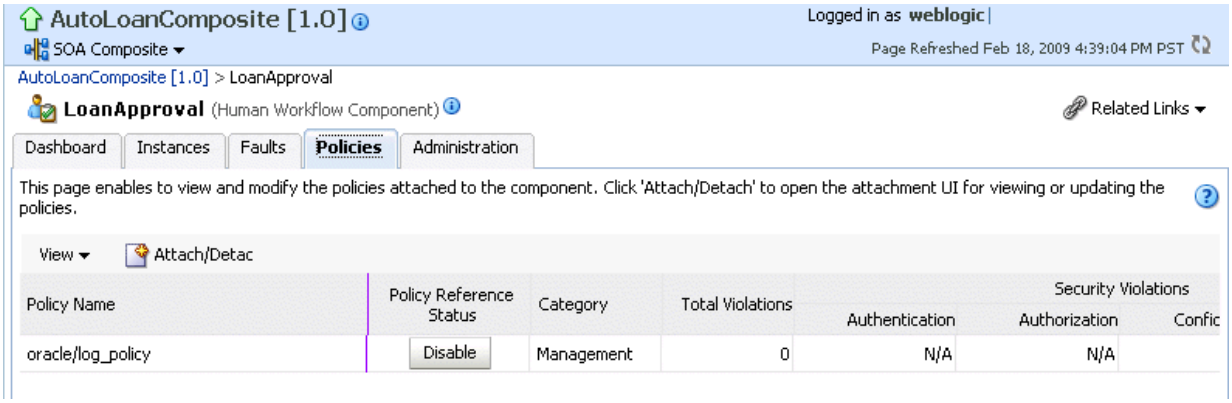
To manage human task service component policies:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, select a specific SOA composite application.

2. Select the human task service component in the **Component Metrics** table.
3. Click **Policies**.

   The Policies page enables you to attach and detach security policies to and from a human task service component. The policies table displays the attached policy name, the policy reference status (enabled or disabled) that you can toggle, the category (Management, Reliable Messaging, MTOM Attachment, Security, or WS Addressing), the total violations, and the authentication, authorization, confidentiality, and integrity failures since the SOA Infrastructure was last restarted.



4. Click **Attach/Detach**.

   If multiple components are available, you are prompted to select the service or component for which to perform the attachment or detachment.

5. Select the service or component to which to attach or detach a policy.

   This invokes a dialog for attaching or detaching policies.

   Policies currently attached appear in the **Attached Policies** section. Additional policies available for attachment appear in the **Available Policies** section.

6. Select to attach policies appropriate to your environment.
7. Click **Attach**.
8. When you are finished attaching policies, click **Validate**.
9. If an error message appears, make the necessary corrections until you no longer have any validation errors.
10. Click **OK**.

    The attached policy is displayed in the policies table.

For more information, see the following documentation:

- Section 1.4.3.2, "Introduction to Policies"
- Section 7.6, "Managing SOA Composite Application Policies" for the dialogs that are displayed during policy attachment
- *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for definitions of available policies and details about which ones to use for your environment

22.2 Recovering from Human Workflow Service Engine Faults

You can view and recover from faults in the human workflow service engine. All human task service component faults, regardless of the SOA composite application instance of which they are a part, can be viewed in the human workflow service engine.

Human workflow invocations from the BPEL service engine use different transactions than BPEL processes. Therefore, if a BPEL transaction is rolled back for any reason, the workflow task instances are still created.

To view and recover from human workflow service engine faults:

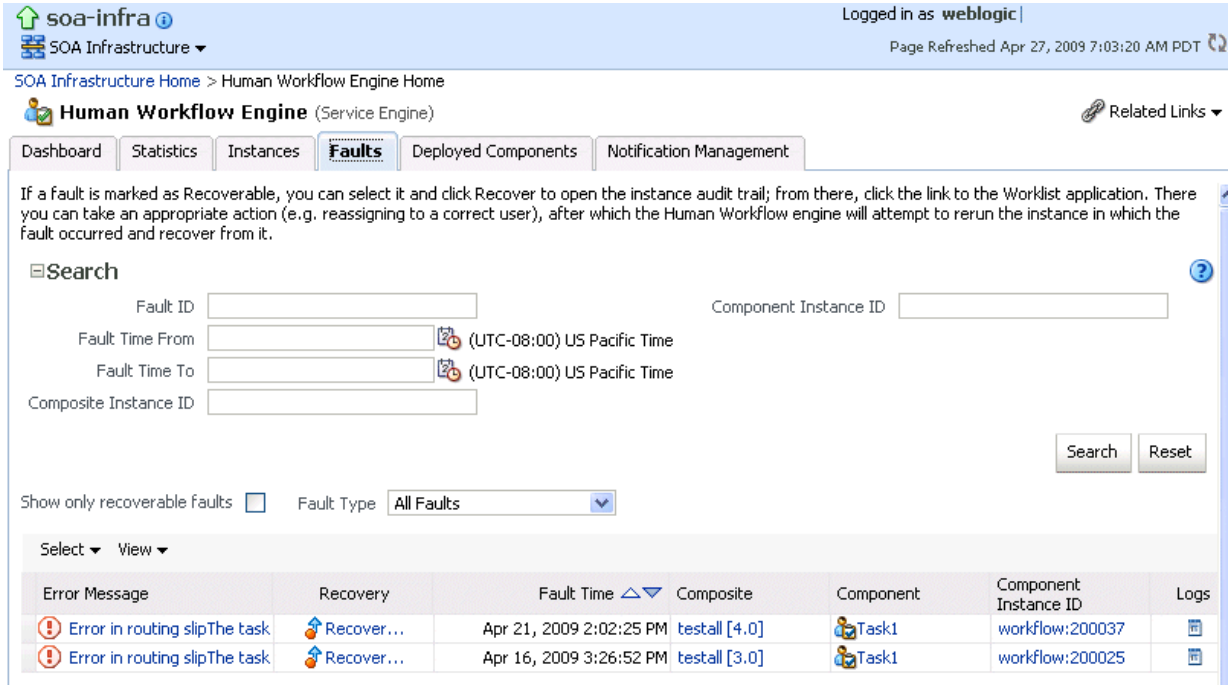
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **Human Workflow**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **Human Workflow**.

2. Click **Faults**.

   The Faults page displays the following details:

   - A utility for searching for a specific fault by specifying criteria and clicking **Search**. Click the **Help** icon for details.
   - Faults that occurred in the human workflow service engine, including the fault ID, error message, whether you can recover from the fault, the time at which the fault occurred, the SOA composite application and human task service component in which the fault occurred, the instance ID of the human task service component, and a link to a log file describing the fault.



Human task service engine faults identified as recoverable can be recovered from Oracle BPM Worklist.

3. Perform fault recovery through either of the following methods:

   a. In the **Error Message** column, click a specific message to display complete fault details, including the fault ID, fault time, fault location, fault type, and error message text. If the fault is recoverable, a **Recover Now** button is displayed that you can click to recover from the fault. Clicking this button invokes the human workflow audit trail page for the instance. The audit trail page has a link to Oracle BPM Worklist called **Go to Worklist Application**, where you can go to recover from the fault. The Oracle BPM Worklist link does not take you directly to the fault; you must manually locate the fault.

   b. In the **Recovery** column, click a fault that is marked as recoverable to invoke the human workflow audit trail page for the instance. The audit trail page provides the same link to Oracle BPM Worklist called **Go to Worklist Application**.

4. Perform the following additional monitoring tasks from within the faults table:

   a. Click the **Show only recoverable faults** checkbox to display only faults from which you can recover.

   b. From the **Fault Type** list, select to display all faults, system faults, business faults, or Oracle Web Services Manager (OWSM) faults in the faults table. Click the **Help** icon for a description of these fault types.

   c. From the **View** list, select **Columns** > **Fault ID** to display the fault IDs for each error message. The fault ID is automatically generated and uniquely identifies a fault. The fault ID is also displayed when you click an error message.

   d. In the **Composite** column, click a specific SOA composite application to access its home page.

   e. In the **Component** column, click a specific service component to access its home page.

f. In the **Component Instance ID** column, click a specific service component ID to access task details about the instance (for example, the current state of a task). Rejected messages do not have a component instance ID.

g. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

22.3 Managing the URI of the Human Task Service Component Task Details Application

You can add or remove the URI of the task details application used in human workflow.

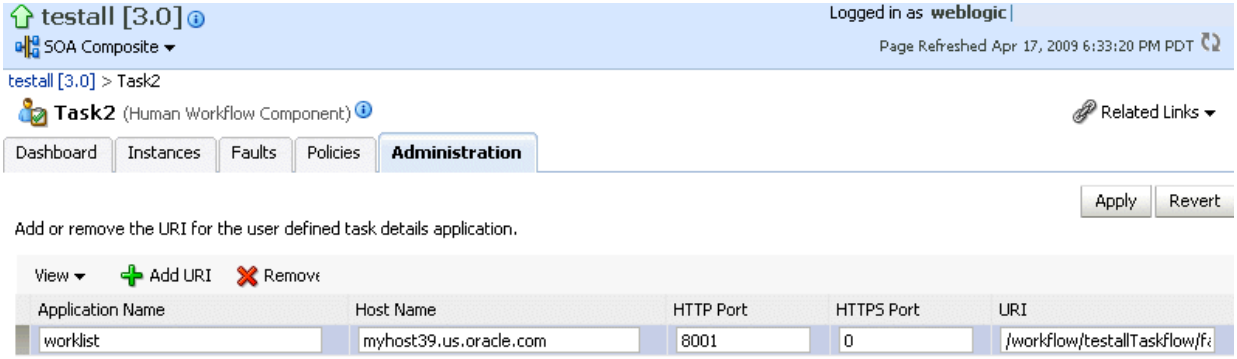
To manage the URI of the human task service component task details application:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, select a specific SOA composite application.

2. Select the human task service component in the **Component Metrics** table.
3. Click **Administration**.

   The Administration page shows the URI for the task details application.



> **Note:** If the SOA server is SSL enabled or disabled, then you must manually enable or disable SSL for any already deployed workflow task detail applications. Change the workflow task display URL to use the correct protocol and port number. To enable the use of the SSL (HTTPS) URL, ensure that the HTTP port setting is left blank.

4. Click the **Add** icon to specify the following details for the URI:
   - Application name
   - Hostname
   - HTTP port
   - HTTPS port (optional)

- URI

5. Click **Apply**.

22.4 Recovering from Human Task Service Component Faults

You can view and recover from human task service component faults. The human task service component is also known as the human workflow service component.

To view and recover from human task service component faults:

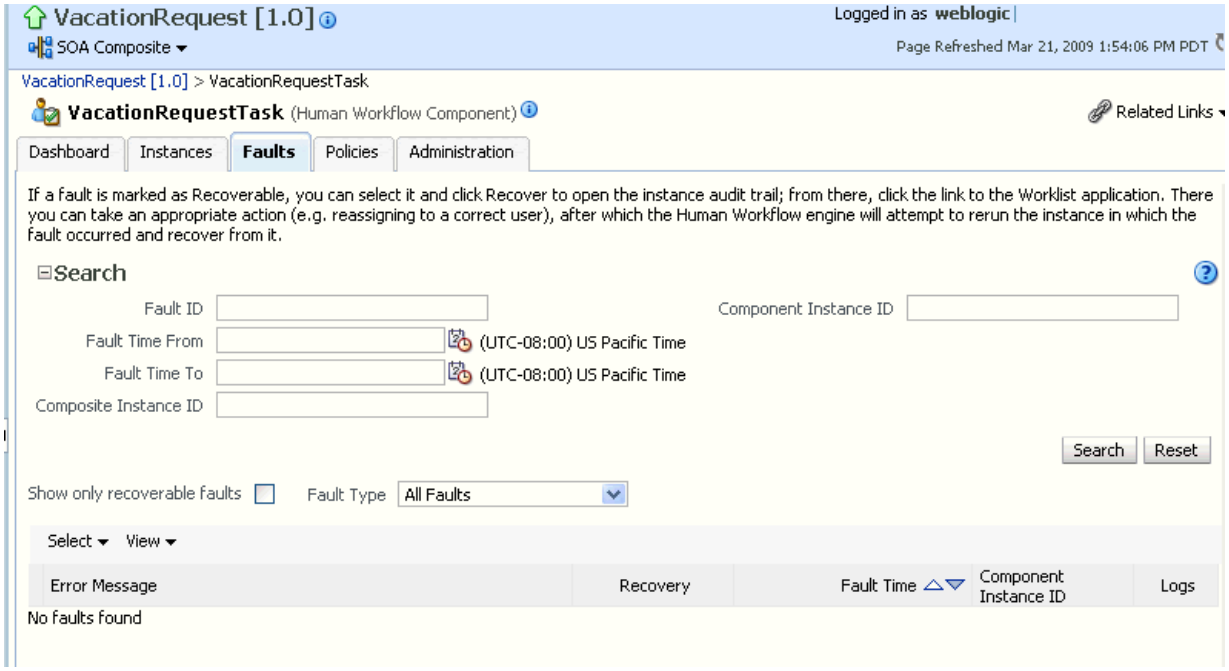
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, select a specific SOA composite application.

2. Select the human task service component in the **Component Metrics** table.
3. Click **Faults**.

   The Faults page displays the following details:

   - A utility for searching for a specific human task service component fault by specifying criteria and clicking **Search**. Click the **Help** icon for details.
   - Faults that occurred in the human task service component, including the fault ID, error message, whether you can recover from the fault, the time at which the fault occurred, the instance ID of the human task service component, and a link to a log file describing the fault.



Human workflow service engine faults identified as recoverable can be recovered from Oracle BPM Worklist.

4. Perform fault recovery through either of the following methods:
   a. In the **Error Message** column, click a specific message to display complete fault details, including the fault ID, fault time, fault location, fault type, and error message text. If the fault is recoverable, a **Recover Now** button is displayed that you can click to recover from the fault. Clicking this button invokes the human workflow audit trail page for the instance. The audit trail page has a link to Oracle BPM Worklist called **Go to Worklist Application**, where you can go to recover from the fault. The Oracle BPM Worklist link does not take you directly to the fault; you must manually locate the fault.
   b. In the **Recovery** column, click a fault that is marked as recoverable to invoke the human workflow audit trail page for the instance. The audit trail page provides the same link to Oracle BPM Worklist called **Go to Worklist Application**.
5. Perform the following additional monitoring tasks from within the faults table:
   a. Click the **Show only recoverable faults** checkbox to display only faults from which you can recover.
   b. From the **Fault Type** list, select to display all faults, system faults, business faults, or OWSM faults in the faults table. Click the **Help** icon for a description of these fault types.
   c. From the **View** list, select **Columns** > **Fault ID** to display the fault IDs for each error message. The fault ID is automatically generated and uniquely identifies a fault. The fault ID is also displayed when you click an error message.
   d. In the **Component Instance ID** column, click a specific service component ID to access task details about the instance (for example, the current state of a task). Rejected messages do not have a component instance ID.
   e. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

22.5 Managing Outgoing Notifications and Incoming Email Notifications

You can manage incoming and outgoing notifications through email in human workflow, including testing messages, resending messages, and identifying messages as spam.

Incoming and outgoing notifications are sent to and from human workflow. Incoming notifications are responses to actionable notifications. For example, an outgoing notification is sent to the manager of an employee requesting vacation leave. The manager approves the request by clicking the **Approve** link in the actionable notification email. This action sends an incoming notification to human workflow for possible additional processing.

To manage outgoing notifications and incoming email notifications:

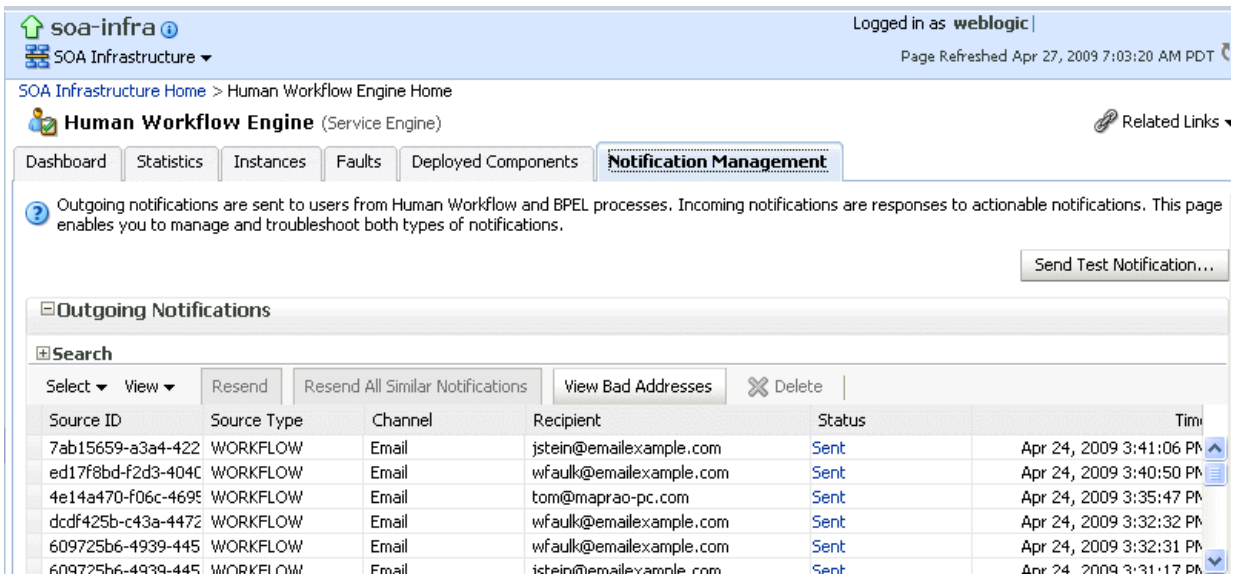
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **Human Workflow**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **Human Workflow**.

2. Click **Notification Management**.

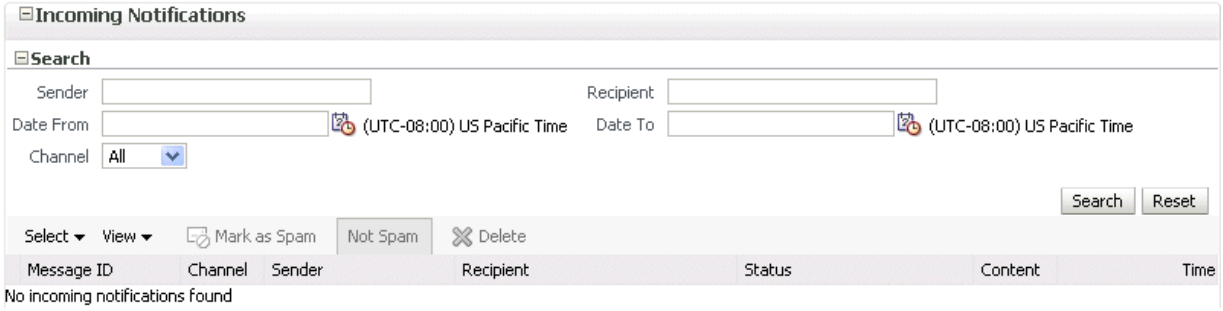
The upper part of the Notification Management page displays the following details:

- A utility for searching for a specific message by specifying criteria and clicking **Search**. You must expand the **Search** icon to display this utility.
- Outgoing notifications, including the source ID, the source type (for example, if a notification is sent by a BPEL service component, the type is BPEL), the channel used (for example, email, SMS, instant messenger, or voice), the address of the message recipient, the message status (for example, error, send, retry, sent), and the time at which the message was sent.



The lower part of the Notification Management page displays the following details:

- A utility for searching for a specific message by specifying criteria and clicking **Search**. You must expand the **Search** icon to display this utility.
- Incoming notifications, including the message ID, the channel used (same types as for outgoing notifications), the address of the message sender, the address of the message recipient, the message status (replied email notification, unsolicited email, unknown email content, response not processed, and response processed), a link to the content of the message, and the time at which the message was received.



3. Perform the following actions on outgoing notifications.

Action	Description
Send Test Notification	Test that outgoing messages are arriving at the correct destination. This ensures that the destination is reachable and messages are arriving. Selecting this option invokes a dialog for specifying the following destination details: ■ Destination address ■ Delivery channel (for example, email) ■ Message subject and content **Note**: You cannot send test notification messages with the messaging extension driver because it requires the following: ■ Specific data to be manually entered into the test page (such as the URI of the task details) ■ URI-specific headers (such as time, user, and so on)
Resend	Select specific outgoing notification messages in the table and click **Resend** to resend. Use this option if you believe that messages are not arriving at their correct destination. For example, you may have incorrectly configured a recipient address. After correcting the address, click **Resend** to test the delivery.
Resend All Similar Notifications	Resend all error notification messages having the same recipient address as the selected one.
View Bad Addresses	Click to display a list of bad or invalid addresses. The addresses are automatically removed from the bad address list after one hour. If you do not want to wait an hour, you can explicitly select and delete them.
Delete icon	Click to delete a selected message.

If outgoing notifications are sent to an incorrect address of a message recipient, they are displayed as errors in the **Recipient** column. You can correct the recipient's address and resend the notification.

4. In the **Recipient** column, click the email address and correct the address.
5. Perform the following actions on incoming notifications.

Action	Description
Mark as Spam	Mark the message sender's address of the selected notification as spam. This action prevents incoming notifications from the same sender address from being delivered again.
No Spam	Mark incoming messages as not being spam. This action enables new messages from the sender's address to be delivered again.
Delete icon	Click to delete a selected message.

For more information about notifications, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

22.6 Moving Human Workflow Data from a Test to a Production Environment

You can migrate Human Workflow user metadata, such as views, mapped attribute (previously known as flex field) mappings, and vacation rules, from a test environment to a production environment using the Human Workflow User Config

Data Migrator. The Data Migrator is available as an `ant` target that can be executed at the command line. You specify the input parameters for the migration of data in a properties file, `migration.properties`.

For example, assume you have two SOA servers installed:

- SOAServer_A

  A test server that includes human workflow user-configurable data (user views, standard views, user rules, group rules, attribute labels, and task payload mapped attribute mappings).

- SOAServer_B

  A production server to which you want to move the data on SOAServer_A.

Since you have a significant amount of data on SOAServer_A, it can be time consuming to manually migrate all of the data to SOAServer_B.

You can use the Data Migrator to move the data from the test server to the production server. You run the `ant` target at the command line of SOAServer_A to migrate data to SOAServer_B.

Migration is always performed through an XML file. The Data Migrator supports the following operations:

- Export operation: Stores all the human workflow user-configurable data from the source SOA server to the XML file.
- Import operation: Creates all the human workflow user-configurable data in the target SOA server by reading from the XML file.

The Data Migrator consists of the following files:

- `migration.properties`: Contains all required input properties in terms of `key-value` pairs for migration operations.
- `build.xml`: Contains the `ant` target `runHwfMigrator` that executes the Data Migrator.

22.6.1 Moving Human Workflow Data from Test to Production Environments

Perform the following steps to move data from a test to a production environment.

To move human workflow data from test to production environments:

1. Ensure that the `PATH` environment variable contains the `JAVA_HOME` and `ANT_HOME` environment variables and that they point to the locations within the Oracle SOA Suite installation.
2. Create a `migration.properties` file in any location to export user metadata for the worklist application (for example, group rules, views, mapped attribute mappings, and vacation rules) from the test environment. See Section 22.6.2.1, "Migration Property File Examples" for instructions on how to specify properties.

   Note the following:

   - You can export mapped attribute mappings.
   - You can export attribute labels.
   - You can only export one type of data at a time.
   - When you export data for a particular user or group, you must export each in separate operations.

– You must export attribute labels before you export mapped attribute mappings.

  To export attribute labels, use the following values in the `migration.properties` file:

```
objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
migrateAttributeLabel = true
```

  To export mapped attribute mappings, use the following values in the `migration.properties` file:

```
objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
migrateAttributeLabel = false
```

3. Export the data with the `ant` script. The following example shows how to invoke the command and specify the parameters:

```
ant -f ant-t2p-worklist.xml
  -Dbea.home=/scratch/oracle/MW_HOME
  -Dsoa.home=/scratch/oracle/MW_HOME/AS11gR1SOA
  -Dmigration.properties.file=migration.properties
  -Dsoa.hostname=hostname -Dsoa.rmi.port=7001
  -Dsoa.admin.user=weblogic
  -Drealm=jazn.com
  -Dmigration.file=/tmp/export_all_userRules.xml
  -Dmap.file=/tmp/export_all_userRules_mapper.xml
```

Note: After specifying the Admin user name, enter the password when prompted.

See Section 22.6.3, "ant Script Data Migration Syntax" for instructions on specifying `ant` properties.

4. Ensure that the application is deployed to the production system.

Note: Human workflow artifacts such as task mapped attribute mappings, rules, views, and approval groups are defined based on namespace. The Data Migrator migrates human workflow artifacts based on namespace. Therefore, it is not possible to migrate human workflow artifacts based on a partition.

5. Create the `migration.properties` file to import user metadata for the worklist application to the production environment.

   Note the following:

   – You can only import one type of data at a time.

   – When you import data for a particular user or group, you must import it in separate operations.

   – You must import attribute labels before you import mapped attribute mappings.

     To import attribute labels, use the following values in the `migration.properties` file:

```
objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
```

```
migrateAttributeLabel = true
```

To import mapped attribute mappings, use the following values in the `migration.properties` file:

```
objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
migrateAttributeLabel = false
```

6. Import the data to the production environment from the file `export_all_userRules.xml`, which you created with the `map.file` property in Step 3. The following example shows how to invoke the command and specify the properties:

```
ant -f ant-t2p-worklist.xml
  -Dbea.home=/scratch/oracle/MW_HOME
  -Dsoa.home=/scratch/oracle/MW_HOME/AS11gR1SOA
  -Dmigration.properties.file=migration.properties
  -Dsoa.hostname=hostname
  -Dsoa.rmi.port=7001
  -Dsoa.admin.user=weblogic
  -Dsoa.admin.password=password
  -Drealm=jazn.com
  -Dmigration.file=/tmp/export_all_userRules.xml
  -Dmap.file=/tmp/export_all_userRules_mapper.xml
```

If the data, such as rules and views, are attached to the user, then the user must be an available user in the production SOA server.

7. Deploy J2EE human task forms, as you would deploy any `.ear` file.
8. If necessary, update the workflow notification configuration with production mail server and inbound and outbound email accounts. See Section 20.1, "Configuring Human Workflow Notification Properties."

22.6.2 migration.properties File Syntax

The `migration.properties` file specifies the input parameters for data migration. The template for this file is located in the following directory:

The `migration.properties` file contains the following input parameters:

```
operationType = {EXPORT | IMPORT}
objectType = {VIEW | RULE | TASK_PAYLOAD_FLEX_FIELD_MAPPING}
name = name of VIEW or TASK_PAYLOAD_FLEX_FIELD_MAPPING
user = username of VIEW or RULE
group = groupname for RULE
grantPermission = {true | false}
migrateAttributeLabel = {true | false}
override = {true | false}
skip = {true | false}
migrateToActiveVersion = {true | false}
```

Argument	Definition
`operationType`	Specify to perform one of the following actions: ▪ `EXPORT`: Data is migrated from a SOA server instance into an XML file. ▪ `IMPORT`: Data is migrated from the XML file into the SOA server instance.

Argument	Definition
`objectType`	Specify the type of object to migrate: ■ `VIEW`: Migrates views. ■ `RULE`: Migrates vacation rules. ■ `TASK_PAYLOAD_FLEX_FIELD_MAPPING`: Migrates mapped attribute mappings.
`name`	Specify the object name if you specified `VIEW` or `TASK_PAYLOAD_FLEX_FIELD_MAPPING` values for the `objectType`. This property refers to the following: ■ `viewName` for VIEW ■ `taskDefinitionId` for `TASK_PAYLOAD_FLEX_FIELD_MAPPING` Specify `ALL` to identify all objects of this type.
`user`	Specify the user name only if you specified the `VIEW` or `RULE` value for the `objectType` property. If a user is not specified for `VIEW`, it implies `STANDARD_VIEW`.
`group`	Specify this property only if you specified the `RULE` value of the `objectType` property. It identifies the group name (for example, `LoanAgentGroup`).
`grantPermission`	Specify this property only if you specified the `VIEW` value of the `objectType` property. ■ `true`: Migrates view definitions and grants. ■ `false`: Migrates only view definitions.
`migrateAttributeLabel`	Specify one of the following values: ■ `true`: Migrates only attribute labels. Payload mappings are *not* migrated. ■ `false`: Does not migrate attribute labels and payload mappings.
`override`	Specify whether to override the data on the target SOA server: ■ `true`: Overrides the existing workflow user-configurable data on the target SOA server. ■ `false`: Does not override the target SOA server instance that has the workflow user-configurable data.
`skip`	Specify error handling details. ■ `true`: Errors are skipped and the migration utility continues processing. ■ `false`: Any encountered error halts the migration.
`migrateToActiveVersion`	Specify a value for mapping task definition IDs. ■ `true`: Maps task definition IDs to the active version in the target SOA server instance. ■ `false`: Does not map task definitions.

22.6.2.1 Migration Property File Examples

This section provides examples how to configure the `migration.properties` file.

22.6.2.1.1 Exporting All Attribute Labels The following example exports all attribute labels.

```
operationType = EXPORT
```

```
objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
name = ALL
user = jcooper
group =
grantPermission = true
migrateAttributeLabel = true
override = true
skip = true
migrateToActiveVersion = true
```

22.6.2.1.2 Importing All Attribute Labels The following example imports all attribute labels.

```
operationType = IMPORT
objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
name = ALL
user = jcooper
group =
grantPermission = true
migrateAttributeLabel = true
override = true
skip = true
migrateToActiveVersion = true
```

22.6.2.1.3 Exporting Specific Attribute Labels The following example exports specific attribute labels.

```
operationType = EXPORT
objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
name = cb801c91-4605-4e96-a234-aeb8441f0388
user = jcooper
group =
grantPermission = true
migrateAttributeLabel = true
override = true
skip = true
migrateToActiveVersion = true
```

22.6.2.1.4 Importing Specific Attribute Labels The following example imports specific attribute labels.

```
operationType = IMPORT
objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
name = cb801c91-4605-4e96-a234-aeb8441f0388
user = jcooper
group =
grantPermission = true
migrateAttributeLabel = true
override = true
skip = true
migrateToActiveVersion = true
```

22.6.2.1.5 Exporting Task Payload Mapped Attribute Mappings for All Task Definition IDs The following example exports task payload mapped attribute mappings for all task definition IDs.

```
operationType = EXPORT
objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
name = ALL
user = jcooper
group =
```

```
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = true
```

22.6.2.1.6 Importing Task Payload Mapped Attribute Mappings for All Task Definition IDs The following example imports task payload mapped attribute mappings for all task definition IDs. Task payload mapped attribute mappings use attribute labels. As a prerequisite, find out the attribute labels involved in the task payload mapped attribute mappings to import. These attribute labels must be available in the target SOA server before the import of task payload mapped attribute mappings into the target SOA server.

The recommended steps are as follows:

- Import the attribute labels into the target SOA server.
- Import the task payload mapped attribute mappings into the target SOA server.

```
operationType = IMPORT
objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
name = ALL
user = jcooper
group =
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = true
```

22.6.2.1.7 Exporting Task Payload Mapped Attribute Mappings for a Specific Task Definition ID The following example exports task payload mapped attribute mappings for a specific task definition ID.

```
operationType = EXPORT
objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
name =
default/HelpDeskRequestComposite!1.0*c9856b8b-bc9e-46a4-8aef-698e539ba1d7/HelpDesk
RequestHumanTask
user = jcooper
group =
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = true
```

22.6.2.1.8 Importing Task Payload Mapped Attribute Mappings for a Specific Task Definition ID The following example imports task payload mapped attribute mappings for a specific task definition ID. Task payload mapped attribute mappings make use of attribute labels. As a prerequisite, find out the attribute labels that are involved in the task payload mapped attribute mappings to import. These attribute labels must be available in the target SOA server before the import of task payload mapped attribute mappings into the target SOA server.

The recommended steps are as follows:

- Import the attribute labels into the target SOA server.
- Import the task payload mapped attribute mappings into the target SOA server.

```
operationType = IMPORT
objectType = TASK_PAYLOAD_FLEX_FIELD_MAPPING
name =
default/HelpDeskRequestComposite!1.0*c9856b8b-bc9e-46a4-8aef-698e539ba1d7/HelpDesk
RequestHumanTask
user = jcooper
group =
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = true
```

22.6.2.1.9 Exporting All Rules for a Specific User This example exports all rules for a specific user. The `group` property is left blank when you export rules for a specific user.

```
operationType = EXPORT
objectType = RULE
name = ALL
user = jcooper
group =
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

22.6.2.1.10 Importing All Rules for a Specific User This example imports all rules for a specific user. The `group` property is left blank when you import rules for a specific user.

```
operationType = IMPORT
objectType = RULE
name = ALL
user = jcooper
group =
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

22.6.2.1.11 Exporting All Rules for a Specific Group This example exports all rules for a specific group. The `user` property is left blank when you export rules for a specific group.

```
operationType = EXPORT
objectType = RULE
name = ALL
user =
group = LoanAgentGroup
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

22.6.2.1.12 Importing All Rules for a Specific Group This example imports all rules for a specific group. The `user` property is left blank when you import rules for a specific group.

```
operationType = IMPORT
objectType = RULE
name = ALL
user =
group = LoanAgentGroup
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

22.6.2.1.13 Exporting All User Views This example exports all user views.

```
operationType = EXPORT
objectType = VIEW
name = ALL
user = jcooper
group =
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

22.6.2.1.14 Importing All User Views This example imports all user views.

```
operationType = IMPORT
objectType = VIEW
name = ALL
user = jcooper
group =
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

22.6.2.1.15 Exporting a Specific User View This example exports a specific user view.

```
operationType = EXPORT
objectType = VIEW
name = jcooperUserView1
user = jcooper
group =
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

22.6.2.1.16 Importing a Specific User View This example imports a specific user view.

```
operationType = IMPORT
objectType = VIEW
name = jcooperUserView1
user = jcooper
group =
grantPermission = true
```

```
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

22.6.2.1.17 Export All Standard Views This example exports all standard views.

```
operationType = EXPORT
objectType = VIEW
name = ALL
user =
group = LoanAgentGroup
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

22.6.2.1.18 Importing All Standard Views This example imports all standard views.

```
operationType = IMPORT
objectType = VIEW
name = ALL
user =
group = LoanAgentGroup
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

22.6.2.1.19 Exporting a Specific Standard View This example exports a specific standard view.

```
operationType = EXPORT
objectType = VIEW
name = MyStandardView1
user =
group = LoanAgentGroup
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

22.6.2.1.20 Importing a Specific Standard View This example imports a specific standard view.

```
operationType = IMPORT
objectType = VIEW
name = MyStandardView1
user =
group = LoanAgentGroup
grantPermission = true
migrateAttributeLabel = false
override = true
skip = true
migrateToActiveVersion = false
```

22.6.3 ant Script Data Migration Syntax

Use the `ant` script for data migration. The script is located in the following directory:

```
ORACLE_HOME/bin/ant-t2p-worklist.xml
```

The script uses the following format to migrate human workflow configurable data from one SOA server to another:

```
ant -f ant-t2p-worklist.xml
-Dbea.home=BEA_HOME
-Dsoa.home=SOA_HOME
-Dmigration.properties.file=MIGRATION_PROPERTY_FILE_PATH
-Dsoa.hostname=SOA_HOSTNAME
-Dsoa.rmi.port=SOA_RMI_PORT
-Dsoa.admin.user=SOA_ADMIN_USER
-Dsoa.admin.password=SOA_ADMIN_PASSWORD
-Drealm=REALM -Dmigration.file=MIGRATION_FILE
-Dmigration.file=<MIGRATION_FILE>
-Dmap.file=MAP_FILE
```

Argument	Definition
`bea.home`	The absolute path of the installation directory for Oracle WebLogic Server.
`soa.home`	The absolute path of the Oracle SOA Suite home directory.
`migration.properties.file`	The absolute path to the `migration.properties` file.
`soa.hostname`	The hostname of the SOA server instance. **Note:** You must specify the complete domain name, such as `myhost.us.oracle.com`, instead of `myhost`.
`soa.rmi.port`	The remote method invocation (RMI) port of the SOA server instance.
`soa.admin.user`	The Admin user name to connect to the SOA server instance.
`soa.admin.password`	The Admin user password to connect to the SOA server instance.
`realm`	The realm of the SOA server instance.
`migration.file`	The complete path location of the migration file in which all user-configurable data from the SOA server is exported to or imported from.
`map.file`	The full path location of the map file in which all the `TaskDefinitionId` mappings in the target SOA server are provided. This file enables you to customize the mapping.

For example:

```
ant -f ant-t2p-worklist.xml
-Dbea.home=/net/myhost/jsmith/fmwhome
-Dsoa.home=/net/myhost/jsmith/fmwhome/AS11gR1SOA
-Dmigration.properties.file=migration.properties
-Dsoa.hostname=myhost.us.oracle.com -Dsoa.rmi.port=7001
-Dsoa.admin.user=weblogic
-Drealm=jazn.com
-Dmigration.file=/tmp/export_all_userRules.xml
-Dmap.file=/tmp/export_all_userRules_mapper.xml
```

Note: After specifying the Admin user name, enter the password when prompted.

Part IX

Administering Oracle Business Activity Monitoring

This part describes how to administer Oracle Business Activity Monitoring.

This part includes the following chapters:

- Chapter 23, "Configuring Oracle Business Activity Monitoring"
- Chapter 24, "Monitoring Oracle Business Activity Monitoring"
- Chapter 25, "Managing Oracle Business Activity Monitoring"

23

Configuring Oracle Business Activity Monitoring

This chapter describes how to configure basic properties of Oracle Business Activity Monitoring (Oracle BAM) Server components and Oracle BAM web applications using Oracle Enterprise Manager Fusion Middleware Control, advanced properties using System MBean Browser, and Oracle BAM Adapter properties and connection factories using Oracle WebLogic Server Administration Console. It also explains how to configure logging and log levels, security, distribution lists for alerts and reports, and Oracle User Messaging Service to send email notifications to Oracle BAM users.

This chapter includes the following topics:

- Section 23.1, "Introduction to Configuring Oracle BAM"
- Section 23.2, "Configuring Oracle BAM Web Basic Properties"
- Section 23.3, "Configuring Oracle BAM Server Basic Properties"
- Section 23.4, "Configuring the Logger"
- Section 23.5, "Configuring Oracle User Messaging Service"
- Section 23.6, "Configuring Oracle BAM Distribution Lists"
- Section 23.7, "Configuring Oracle BAM Adapter"
- Section 23.8, "Configuring Oracle BAM Batching Properties"
- Section 23.9, "Configuring Security"
- Section 23.10, "Configuring Advanced Properties"
- Section 23.11, "Oracle BAM Configuration Property Reference"

23.1 Introduction to Configuring Oracle BAM

Oracle BAM Server is the collection of the components Oracle BAM Active Data Cache (Oracle BAM ADC), Oracle BAM Report Cache, Oracle BAM Enterprise Message Sources (EMS), and Oracle BAM Event Engine.

The Oracle BAM web applications are a collection of thin clients (Active Studio, Architect, Administrator, and Active Viewer) and Oracle BAM Report Server.

For more information about each of the subcomponents of Oracle BAM Server and Oracle BAM web applications, see the *Oracle Fusion Middleware User's Guide for Oracle Business Activity Monitoring*.

At runtime, the Oracle BAM application checks the configuration property values. If they are set to `DEFAULT`, Oracle BAM dynamically determines the values of all the

parameters by querying the JRF API, that in turn queries the MBean Server. This feature enables you to set up Oracle BAM on any port number and work out of the box without any configuration changes. By default all of the parameters are set to `DEFAULT`.

If the administrator provides the configuration values during initial postinstallation configuration then Oracle BAM honors them as the values going forward. If for some reason the parameters are not present, Oracle BAM first looks for host and port details in JRF. If Oracle BAM is not able to access JRF, hard-coded default values are provided from the code.

The most commonly used Oracle BAM Server and web applications properties are configured in Oracle Enterprise Manager Fusion Middleware Control. For more information, see Section 23.3, "Configuring Oracle BAM Server Basic Properties" and Section 23.2, "Configuring Oracle BAM Web Basic Properties".

Security configuration is discussed in Section 23.9, "Configuring Security", and adding and managing Oracle BAM roles (permissions) and user accounts is discussed in Section 25.3, "Managing Oracle BAM Users."

Oracle User Messaging Service (UMS) must be configured so that alerts are delivered to Oracle BAM users. For more information, see Section 23.5, "Configuring Oracle User Messaging Service".

When using Oracle BAM in a SOA composite application, the Oracle BAM Adapter configuration must be completed. For more information, see Section 23.7, "Configuring Oracle BAM Adapter".

Advanced properties are located in configuration files that must be edited using the System MBean Browser. For more information, see Section 23.10, "Configuring Advanced Properties". Also, see Section 23.11, "Oracle BAM Configuration Property Reference" for a listing of all properties available for Oracle BAM configuration.

Oracle BAM distribution lists are used to send alerts and reports to multiple Oracle BAM users, rather than having to specify several individual recipients. Distribution lists are configured in the Oracle BAM Administrator application. For more information, see Section 23.6, "Configuring Oracle BAM Distribution Lists".

For performance tuning information for Oracle BAM, see the *Oracle Fusion Middleware Performance and Tuning Guide*.

For Oracle BAM high availability configuration information, see the *Oracle Fusion Middleware High Availability Guide*.

23.2 Configuring Oracle BAM Web Basic Properties

Some basic Oracle BAM web applications properties are configured using Oracle Enterprise Manager Fusion Middleware Control.

Oracle BAM web applications must be restarted when changes are made to any Oracle BAM properties. For information about restarting Oracle BAM, see Section 25.2, "Managing Oracle BAM Availability".

The following topics describe how to configure each property:

- Section 23.2.1, "Configuring Oracle BAM Web Applications Properties"
- Section 23.2.2, "Configuring the Application URL"
- Section 23.2.3, "Configuring the Report Loading Indicator"
- Section 23.2.4, "Configuring the Server Name"

For information about configuring advanced properties, see Section 23.10, "Configuring Advanced Properties" and Section 23.11, "Oracle BAM Configuration Property Reference."

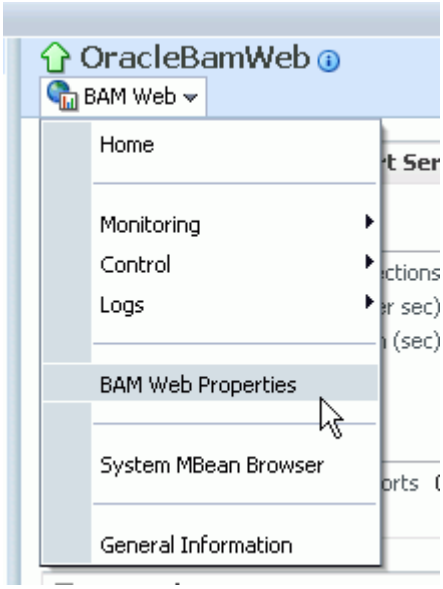
23.2.1 Configuring Oracle BAM Web Applications Properties

The Oracle BAM web applications properties are configured in the BAM Web Properties page in Fusion Middleware Control.

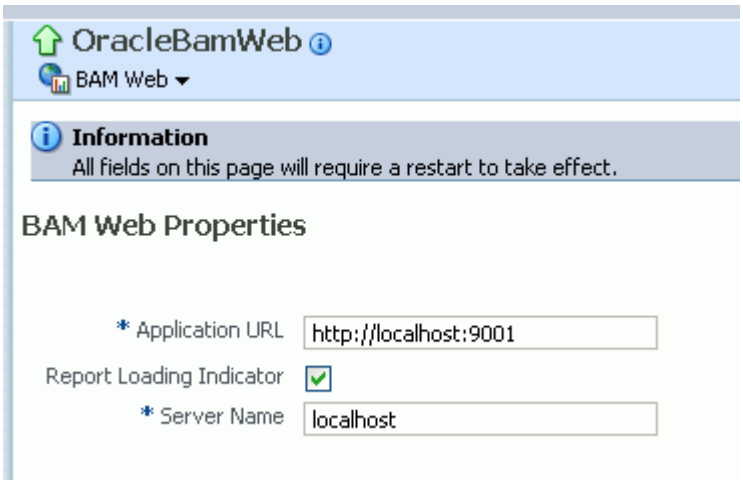
To configure Oracle BAM web applications properties:

1. Go to the BAM Web Properties page in Fusion Middleware Control by selecting the **OracleBamWeb** node in the **BAM** folder.

2. Choose **BAM Web Properties** from the **BAM Web** menu.



The BAM Web Properties page opens.



3. Enter the appropriate values in each of the fields provided.

   See the following sections for information about configuring each of the properties:

   - Section 23.2.2, "Configuring the Application URL"
   - Section 23.2.3, "Configuring the Report Loading Indicator"
   - Section 23.2.4, "Configuring the Server Name"

4. Click **Apply**.

   You must restart Oracle BAM after any property changes.

23.2.2 Configuring the Application URL

The **Application URL** property value must be updated from `localhost` to the actual hostname to generate the correct URLs for reports and alerts.

The **Application URL** property is configured in both the Oracle BAM web applications properties page and the Oracle BAM Server properties page. In Oracle BAM web applications page, the URL is used to generate the full URL for reports and alerts.

This property is common to both Oracle BAM web applications and Oracle BAM Server. If it is configured on one page, the same value appears on the other configuration page.

23.2.3 Configuring the Report Loading Indicator

The **Report Loading Indicator** property specifies whether the report loading indicator is enabled or disabled by default when viewing reports in Oracle BAM Active Studio and Oracle BAM Active Viewer.

With this property enabled, the report loading indicator is shown by default when a report is viewed; however, individual users can choose to disable the report loading indicator in their user preferences configuration. For more information, see the *Oracle Fusion Middleware User's Guide for Oracle Business Activity Monitoring*.

23.2.4 Configuring the Server Name

The **Server Name** property provides the hostname of Oracle BAM Server. Oracle BAM web applications are separate applications that require the location of Oracle BAM

Server to get data from the Active Data Cache and Report Cache for the Oracle BAM Report Server.

23.3 Configuring Oracle BAM Server Basic Properties

Some basic Oracle BAM Server properties are configured using Oracle Enterprise Manager Fusion Middleware Control.

Oracle BAM Server must be restarted after any changes to Oracle BAM properties. For information about restarting Oracle BAM, see Section 25.2, "Managing Oracle BAM Availability".

The following topics describe how to configure each property:

- Section 23.3.1, "Configuring Oracle BAM Server Properties"
- Section 23.3.2, "Configuring the Data Source JNDI"
- Section 23.3.3, "Configuring the Application URL"
- Section 23.3.4, "Configuring Viewset Sharing"
- Section 23.3.5, "Configuring the Report Cache Persistence Manager"
- Section 23.3.6, "Configuring Oracle Data Integrator Integration Properties"
- Section 23.3.7, "Configuring the Outbound Email Account"

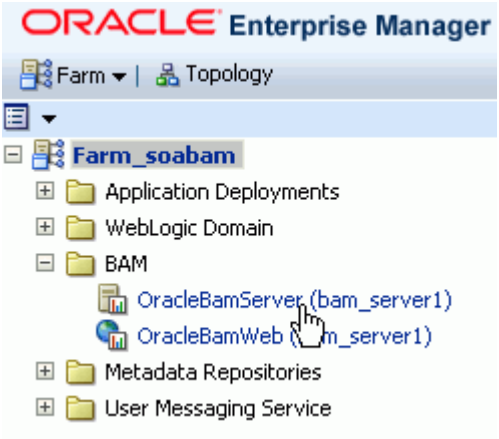
For information about configuring advanced properties, see Section 23.10, "Configuring Advanced Properties" and Section 23.11, "Oracle BAM Configuration Property Reference."

23.3.1 Configuring Oracle BAM Server Properties

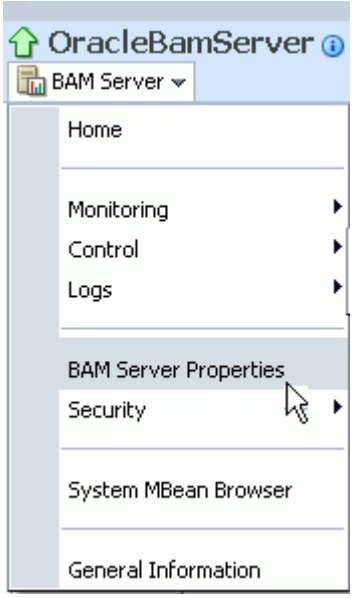
Oracle BAM Server properties are configured in the BAM Server Properties page in Oracle Enterprise Manager Fusion Middleware Control.

To configure Oracle BAM Server properties:

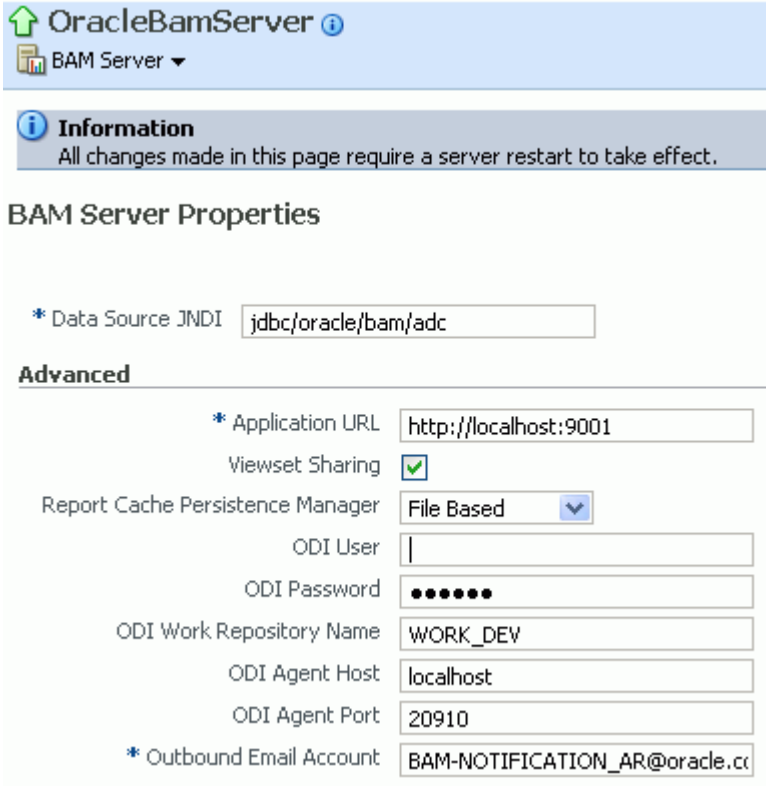
1. Go to the BAM Server Properties page by selecting the **OracleBamServer** node in Fusion Middleware Control **BAM** folder.



2. Choose **BAM Server Properties** from the **BAM Server** menu.



The BAM Server Properties page opens.



3. Enter the appropriate values in each of the fields provided.

   See the following sections for information about configuring each of the properties:

   - Section 23.3.2, "Configuring the Data Source JNDI"
   - Section 23.3.3, "Configuring the Application URL"
   - Section 23.3.4, "Configuring Viewset Sharing"

- Section 23.3.5, "Configuring the Report Cache Persistence Manager"
- Section 23.3.6, "Configuring Oracle Data Integrator Integration Properties"
- Section 23.3.7, "Configuring the Outbound Email Account"

4. Click **Apply**.

   You must restart the Oracle BAM application after any property changes.

23.3.2 Configuring the Data Source JNDI

The **Data Source JNDI** property specifies the database used by Oracle BAM ADC. Enter the data source JNDI name created for Oracle BAM ADC. For general information about configuring data sources see *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server*

23.3.3 Configuring the Application URL

The **Application URL** property is used only by Oracle BAM web applications. This configuration property is no longer used by Oracle BAM Server. For information about configuring this property for Oracle BAM web applications, see Section 23.2.2, "Configuring the Application URL".

23.3.4 Configuring Viewset Sharing

The **Viewset Sharing** property enables view set sharing when possible. A view set is an object that represents a query to a given Oracle BAM data object. A view set can include fields (which can be columns from the data object or lookups from other data objects), aggregates, groups, or calculated fields. A view set can also be sorted, filtered, and row-level security is applied implicitly to it.

Typically, a particular view set can be shared with other users if they are trying to access the same Oracle BAM dashboard, if the view sets are not dissimilar due to factors such as row-level security, prompts or parameters used in filters, and so on.

A snapshot is a query result set that can be created by the view set. Snapshots can be produced from a view set after it is opened. Because snapshot generation is so performance-intensive, view set sharing allows Oracle BAM Server to take only one snapshot of a given view to share across multiple users of the same dashboard.

Sharing view sets between users of the same views improves performance, but requires that new users wait for view sets to be synchronized as each new user begins sharing the view set.

The first user who opens an Oracle BAM dashboard receives a new snapshot (which is cached) and immediately receives active data in real time. Changes for active data, called change lists, are cached.

When additional users open the same dashboard, they receive the cached snapshot that was generated for the first user, and they receive cached change lists (more rapidly than the first user) while the `Synchronizing Active Data` message is displayed.

By default, up to 50 change lists are cached, so if multiple users open a dashboard, secondary users must wait for up to 50 change lists to be sent to the dashboard before they begin receiving real-time active data.

The `ElementsCountLimit` property can be set to allow fewer change lists to be cached. For information about configuring advanced properties, see Section 23.11, "Oracle BAM Configuration Property Reference".

23.3.5 Configuring the Report Cache Persistence Manager

The **Report Cache Persistence Manager** property specifies the persistence location (file based or memory based). Select the appropriate persistence location for your Oracle BAM Report Cache.

23.3.6 Configuring Oracle Data Integrator Integration Properties

The Oracle Data Integrator integration properties specify values for integrating Oracle Data Integrator with Oracle BAM Server.

Enter the appropriate values for the following properties:

ODI User: The user that executes the scenarios in Oracle Data Integrator.

ODI Password: The encoded Oracle Data Integrator password.

ODI Work Repository Name: The name of the Oracle Data Integrator work repository database.

ODI Agent Host: The IP address or hostname of the server where the Oracle Data Integrator agent is running.

ODI Agent Port: The Transmission Control Protocol (TCP) port on which the Oracle Data Integrator agent is listening.

23.3.7 Configuring the Outbound Email Account

This section explains how to configure the email address that appears in the From header of email notifications sent by the Oracle BAM Event Engine.

The **Outbound Email Account** property on Oracle BAM Server must point to a dedicated email account for delivering Oracle BAM alert notification email messages and Oracle BAM report link email messages.

If an Oracle BAM alert configured with a secondary action to send an email notification fails on the primary action, this property must be configured with an email address that Oracle UMS can access at runtime. Creating a dedicated email account for this purpose is recommended.

> **Note:** Administrators should not use personal email accounts to test alerts, because Oracle UMS may delete email notifications in the mail box and continue deleting messages as they arrive.

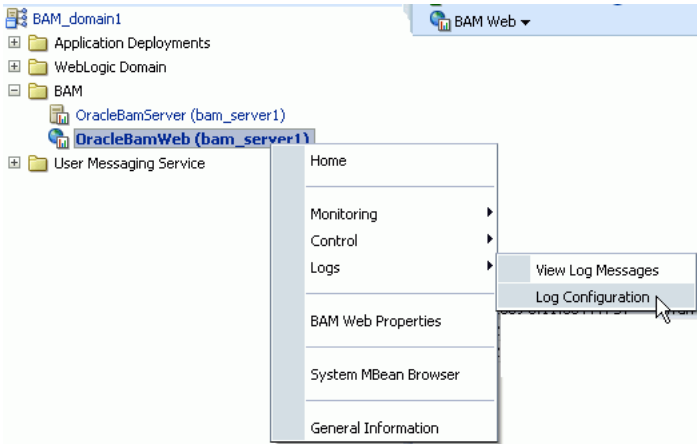
To complete the configuration, you must configure the Oracle UMS `usermessagingdriver-email` driver. For details, see Section 23.5, "Configuring Oracle User Messaging Service".

Oracle BAM Event Engine has another configuration property, `TimeoutPeriodToGetFailedStatusesInMins`, which is the time interval for which the Event Engine waits after delivering an email notification, to see if it receives any failure notification from Oracle UMS. The default value for this property is 10 minutes. If a failure notification does not arrive within this time, the Event Engine assumes that email delivery was successful, and the failover action is discarded. Failure notifications arriving after this time are ignored by the Event Engine. For information about configuring `TimeoutPeriodToGetFailedStatusesInMins`, see Section 23.11, "Oracle BAM Configuration Property Reference".

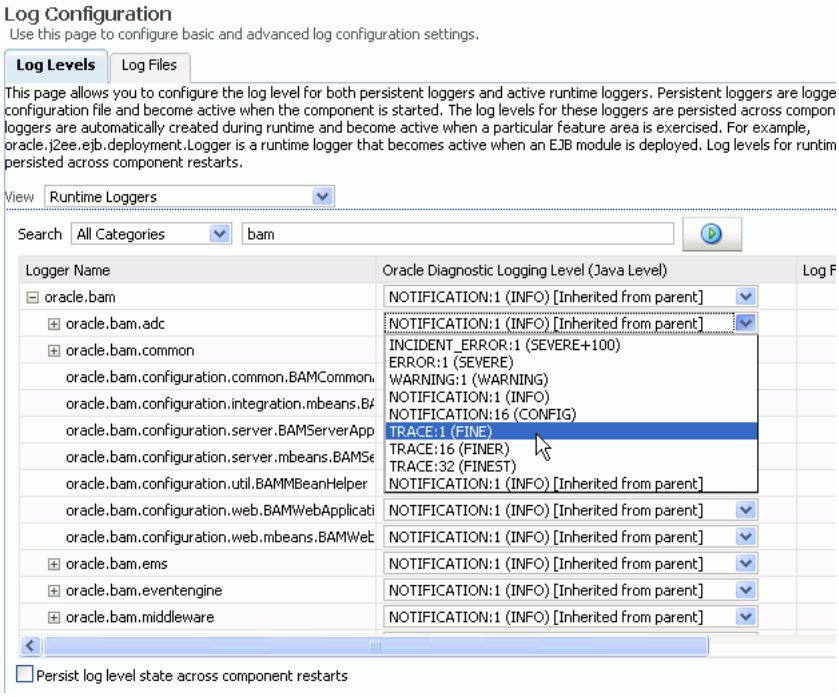
23.4 Configuring the Logger

Use the Log Configuration page to configure logging levels. For information about viewing the logs, see Section 24.6, "Monitoring Oracle BAM Logs." Also, see Section 3.4, "Configuring Log Files" and the *Oracle Fusion Middleware Administrator's Guide* for general information about logging and log levels.

To open the Log Configuration page, right-click the **OracleBamServer** node or **OracleBamWeb** node in the navigation tree and choose **Logs** > **Log Configuration**.



For each logger, select the desired notification level.



23.5 Configuring Oracle User Messaging Service

Oracle User Messaging Service (UMS) must be configured properly in Fusion Middleware Control to send email notifications when alerts are issued.

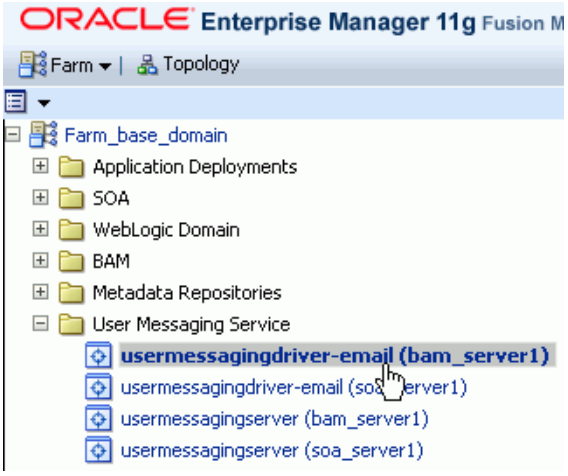
The UMS email driver monitors the outbound email account configured for Oracle BAM Server for any delivery failures, including rejected emails. Email delivery failure

notifications are asynchronous, that is, there is no definite time within which email rejection notification is received.

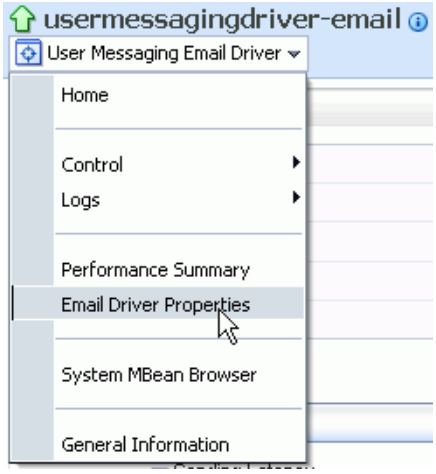
To configure the outbound email account see Section 23.3.7, "Configuring the Outbound Email Account."

To configure the service:

1. In Fusion Middleware Control, go to the usermessagingdriver-email (User Messaging Email Driver) page by expanding the **User Messaging Service** folder, then selecting the **usermessagingdriver-email (bam_server1)** node.



2. From the User Messaging Email Driver shortcut menu, choose **Email Driver Properties**.



3. In the **Driver-Specific Configuration** section, the properties **IncomingMailServer**, **IncomingUserIDs**, and **IncomingUserPasswords** must be provided with the Internet Message Access Protocol (IMAP) server name (Oracle UMS also supports POP3), email account name, and password to access this email account.

Driver-Specific Configuration

Name	Description	Mandatory	Encoded Credential	Value
IncomingMailServer	The host name of the incoming mail server. Required only if e-mail receiving is supported on the driver instance.			
IncomingUserIDs	The list of user names of the mail accounts the driver instance is polling from. Each name must be separated by a comma, for example, foo,bar. Required only if e-mail receiving is supported on the driver instance.			
IncomingUserPasswords	The list of passwords corresponding to the user names. Each password is separated by a comma and must reside in the same position in the list as their corresponding user name appears on the usernames list. Required only if e-mail receiving is supported on the driver instance.		✓	Type of Password: Indirect Password, Create New User Indirect Username/Key: Password:

For more information, see Section 26.4, "Configuring User Messaging Service Drivers". Table 26–7, " Custom Email Properties" contains specific information about the properties you must configure.

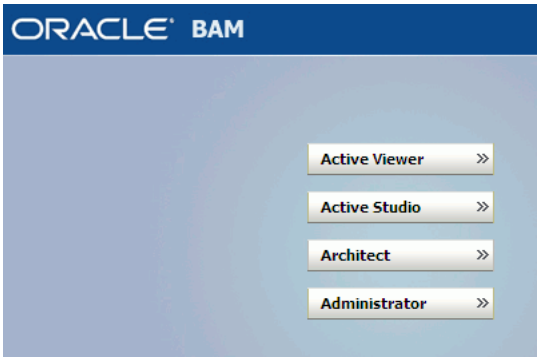
4. Click **Apply** to save the changes.
5. Enter the email account in the **Outbound Email Account** property in the Oracle BAM Server configuration. For details, see Section 23.3.7, "Configuring the Outbound Email Account".

23.6 Configuring Oracle BAM Distribution Lists

Oracle BAM distribution lists are used to send alerts and reports to multiple Oracle BAM users, rather than having to specify several individual recipients. Oracle BAM distribution lists are configured using the Oracle BAM Administrator application.

To configure distribution lists:

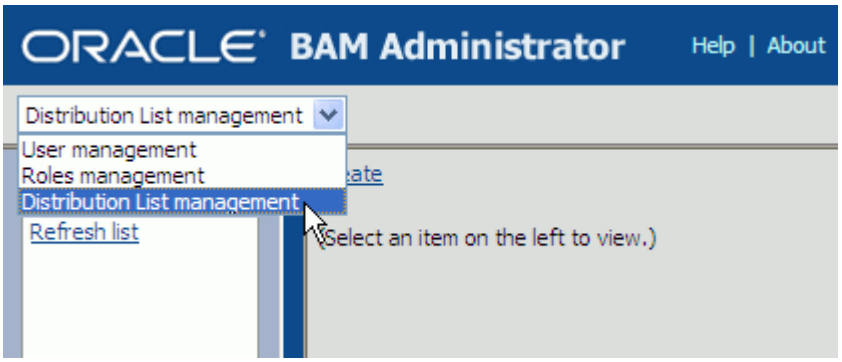
1. Go to the Oracle BAM start page, log in, and select **Administrator**.



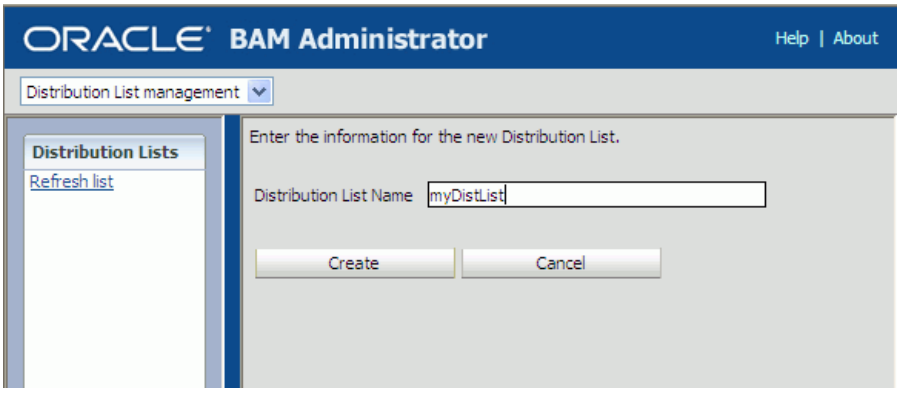
The Oracle BAM start page is located at:

`http://host_name:port_number/OracleBAM/`

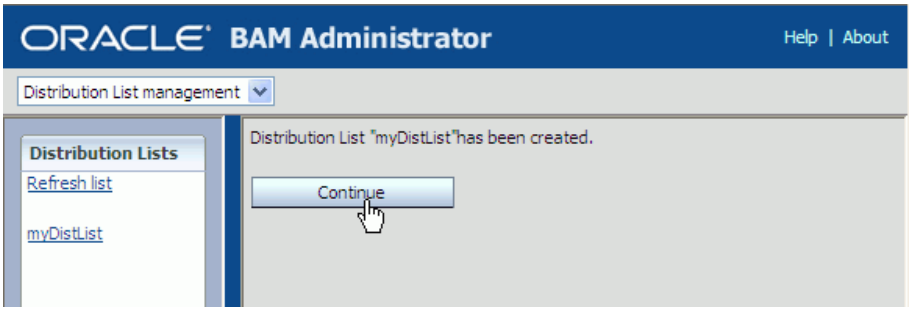
2. Select **Distribution List management** from the list.



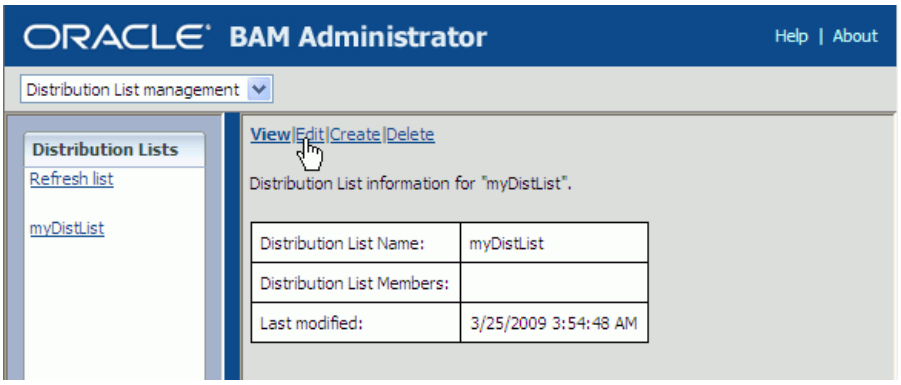
3. Click **Create**.
4. Enter a unique **Distribution List Name** and click **Create**.



5. Click **Continue** to save the distribution list name.

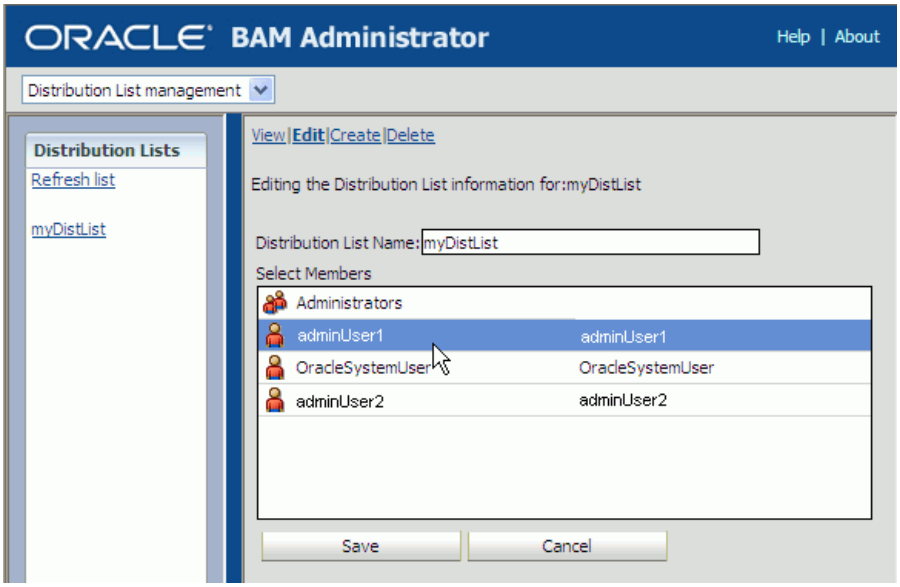


6. Select the new distribution list that is displayed in the **Distribution Lists** list on the left side of the page, and click **Edit**.



7. Select user accounts to add to the distribution list from the **Select Members** list.

   You can select multiple users by clicking on several user names. The highlighted user names are added to the distribution list.



8. Click **Save**.

   The users are added to the distribution list.

23.7 Configuring Oracle BAM Adapter

Oracle BAM Adapter properties, including the adapter batching properties, are configured in Oracle WebLogic Server Administration Console. For details, see Section 23.7.1, "Configuring Oracle BAM Adapter Properties".

Some issues that must be considered when Oracle BAM Adapter (including Oracle BAM sensor actions in a BPEL process) is used in a SOA composite application are configuring connection factories, trusted domains, and credential mapping.

Before deploying applications that use Oracle BAM Adapter, a connection factory to Oracle BAM Server must be configured. You can configure both Remote Method Invocation (RMI) and Simple Object Access Protocol (SOAP) connection factories. For details, see Section 23.7.2, "Configuring Oracle BAM Connection Factories".

When using the RMI connection between a SOA composite application and Oracle BAM Server, that is when they are deployed in different domains, trusted domain configuration must be done in Oracle WebLogic Server Administration Console. For more information, see Section 23.7.3, "Configuring Trusted Domains".

If the Oracle BAM Adapter is using credentials rather than a plain text user name and password, in order for Oracle BAM Adapter (including Oracle BAM sensor actions in a BPEL process) to connect to Oracle BAM Server, the credentials must also be established and mapped. For more information, see Section 23.7.4, "Configuring Credential Mapping".

The `oracle.bam.adapter` logger should be configured to troubleshoot issues with Oracle BAM Adapter. For information about logger configuration, see Section 23.4, "Configuring the Logger".

Notes: Use plain text user names and passwords only in nonproduction mode. Do not mix using credential mapping and plain text user information; apply one at a time to avoid confusion.

Because Oracle BAM and Oracle SOA Infrastructure use different identity stores, an Oracle BAM user must be configured in the service infrastructure identity store if there is not a common service infrastructure user and Oracle BAM user.

Oracle BAM Adapter does not support transactional behavior. Even though there is a section in Oracle WebLogic Server Administration Console for configuring transaction support, Oracle BAM Adapter must remain at the No Transaction level.

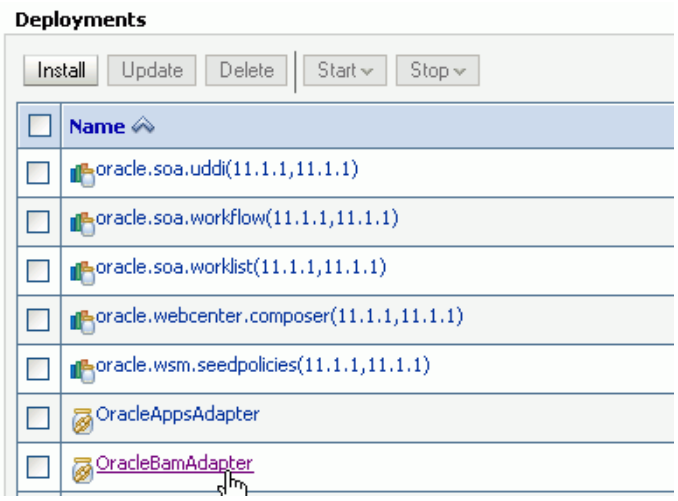
23.7.1 Configuring Oracle BAM Adapter Properties

Oracle BAM Adapter properties are configured in Oracle WebLogic Server Administration Console.

To configure Oracle BAM Adapter properties:

1. Go to the Oracle WebLogic Server Administration Console (`http://host_name:port_number/console`), and log on.
2. In the **Domain Structure** menu, select **Deployments**.

3. In the **Deployments** summary table, search for **OracleBamAdapter** and click it.



4. In the Settings for OracleBamAdapter page, click the **Configuration** tab, and then the **Properties** tab in the second row.

5. Enter the new values, select the checkboxes for each updated value, and click **Save**.

   For information about each property, see Table 23–1, " Oracle BAM Adapter Configuration Properties".

   Also, see Section 23.8, "Configuring Oracle BAM Batching Properties" for general information about configuring batching properties.

   > **Note:** To commit these changes to the deployment plan you must click **Activate Changes** in the Change Center. If the **Activate Changes** button does not appear, see the "Oracle BAM Issues With Oracle WebLogic Server Administrative Console" section in the *Oracle Fusion Middleware User's Guide for Oracle Business Activity Monitoring* for a workaround.

6. Restart the SOA domain server.

   It must be restarted because that is where the Oracle BAM Adapter is deployed.

Table 23–1 lists the Oracle BAM Adapter configuration properties.

Table 23–1 Oracle BAM Adapter Configuration Properties

RMI Property	SOAP Property	Default	Description
Batch_Lower_Limit	**SOAP_Batch_Lower_Limit**	1000	The minimum amount of elements in a batch before it is sent out.
Batch_Upper_Limit	**SOAP_Batch_Upper_Limit**	5000	The maximum amount of elements in a batch before it is sent out.
Batch_Timeout	**SOAP_Batch_Timeout**	5000	The timeout in milliseconds after which the batch is sent out even if it is not full.

Table 23–1 (Cont.) Oracle BAM Adapter Configuration Properties

RMI Property	SOAP Property	Default	Description
Block_On_Batch_Full	**SOAP_Block_On_Batch_Full**	`false`	The default setting for block on batch full, which indicates whether batching operations should block when the batch is full. If `true` then batching calls block until room is freed up in the batch for the operation. If `false` then `BatchIsFullException` exception is thrown.
Number_Batches	**SOAP_Number_Batches**	10	The limit of the number of pending calls allowed.

23.7.1.1 Configuring the Adapter to Retry Sending Messages

If messages from BPEL process monitors are published when Oracle BAM Server is unavailable, the messages are not persisted and therefore are not published to the Oracle BAM data objects when Oracle BAM Server comes back online.

You can change this behavior by setting the Oracle BAM Adapter property **Block_On_Batch_Full** to `true` in Oracle BAM Adapter configuration as described in Section 23.7.1, "Configuring Oracle BAM Adapter Properties."

For more information about BPEL process monitors see the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

23.7.2 Configuring Oracle BAM Connection Factories

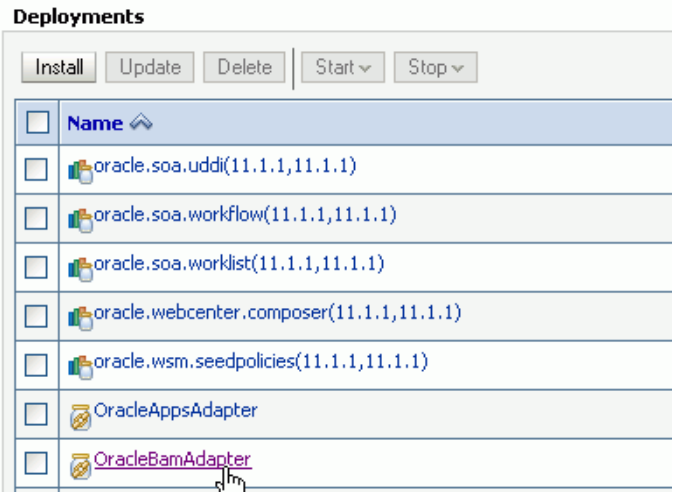
Some configuration in the Oracle WebLogic Server Administration Console is necessary before using Oracle BAM Adapter.

Use the Oracle WebLogic Server Administration Console to configure the Oracle BAM connection factories for connections with Oracle BAM Server.

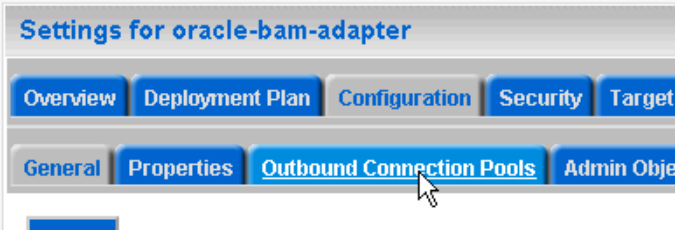
To configure Oracle BAM connection factories:

1. Go to the Oracle WebLogic Server Administration Console (`http://`*`host_name`*`:`*`port_number`*`/console`), and log on.
2. In the **Domain Structure** menu, select **Deployments**.

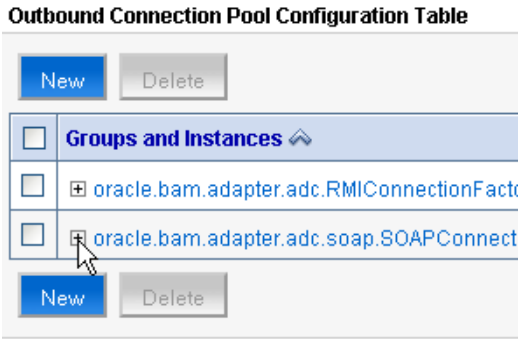
3. In the **Deployments** summary table, search for **OracleBamAdapter** and click it.



4. In the Settings for oracle-bam-adapter page, click the **Configuration** tab, and then the **Outbound Connection Pools** tab in the second row.

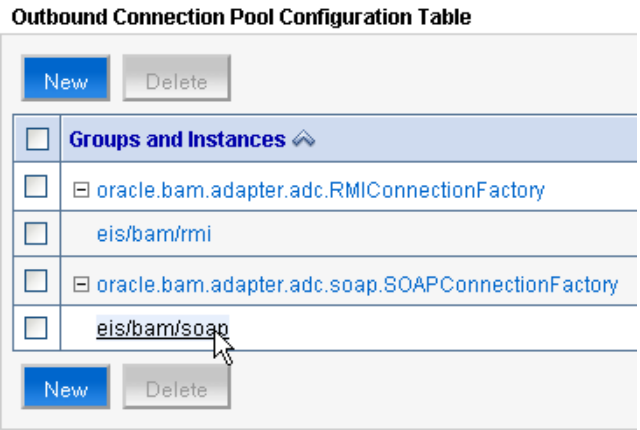


5. Configure the connection factory interface that you intend to use in Oracle JDeveloper, by expanding the appropriate **Group and Instance** target and clicking the Java Naming and Directory Interface (JNDI) name.

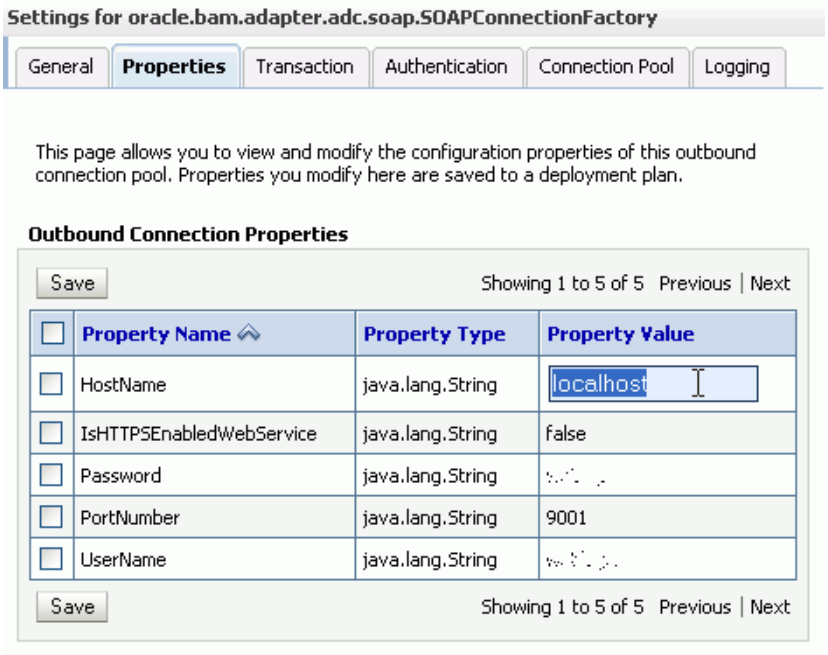


To configure the connection factory for RMI-based calls, click to expand the **oracle.bam.adapter.adc.RMIConnectionFactory** group and select the JNDI name that the user uses in Oracle JDeveloper. The **eis/bam/rmi** connection factory is provided out of the box.

To configure the connection factory for SOAP-based calls, click to expand the **oracle.bam.adapter.adc.SOAPConnectionFactory** group and select the JNDI name that the user uses in Oracle JDeveloper. The **eis/bam/soap** connection factory is provided out of the box.



6. Configure each of the properties by clicking in the table cells and entering the values.

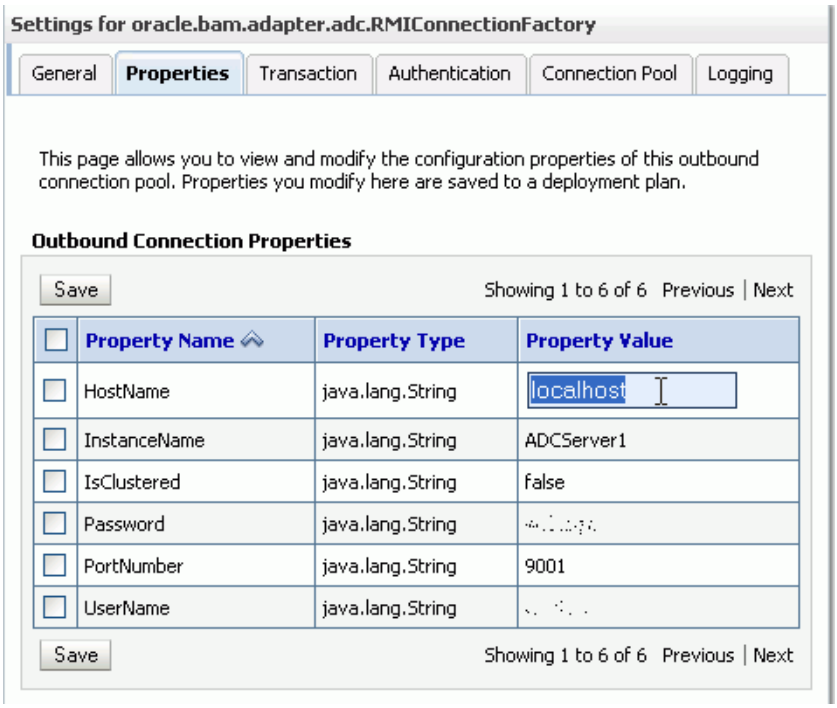


Note: Press Enter after entering a value in the table.

Note: The **UserName** field should contain an Oracle BAM user who is a member of application-level role Administrator or Report Architect. For information about assigning users to Oracle BAM application roles, see Section 25.3.3, "Adding Members to Application Roles" and Section 25.3.4, "Introduction to Oracle BAM Application Roles".

This configuration creates the connection factory with HTTP. To configure an HTTPS connection factory, see Section 23.7.2.1, "Configuring HTTPS for Oracle BAM Adapter".

The RMI protocol has different configuration properties. You must provide **InstanceName** and **IsClustered** in addition to the standard connection properties.



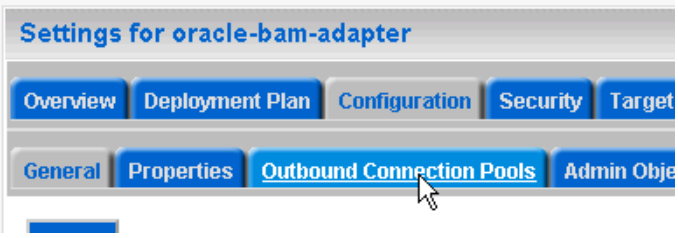
7. Select the checkboxes next to the values you have updated and click **Save**.

Note: To commit these changes to the deployment plan you must click **Activate Changes** in the Change Center. If the **Activate Changes** button does not appear, see the "Oracle BAM Issues With Oracle WebLogic Server Administrative Console" section in the *Oracle Fusion Middleware User's Guide for Oracle Business Activity Monitoring* for a workaround.

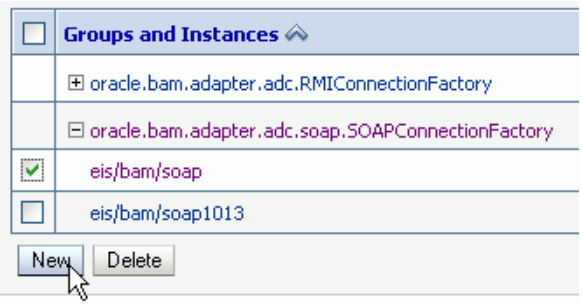
23.7.2.1 Configuring HTTPS for Oracle BAM Adapter

To configure HTTPS:

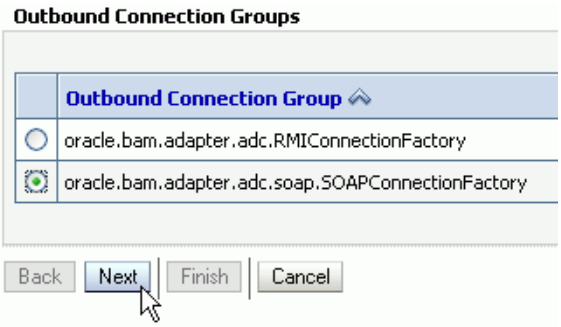
1. Locate the **oracle.bam.adapter.adc.soap.SOAPConnectionFactory** connection factory in the Oracle WebLogic Server Administration Console as described in Section 23.7, "Configuring Oracle BAM Adapter".



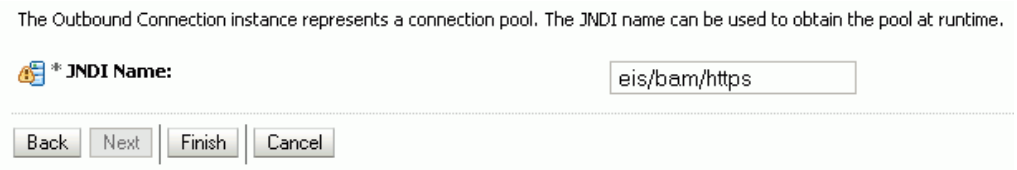
2. Click to expand the **oracle.bam.adapter.adc.soap.SOAPConnectionFactory** node, select the **eis/bam/soap** option, and click **New**.



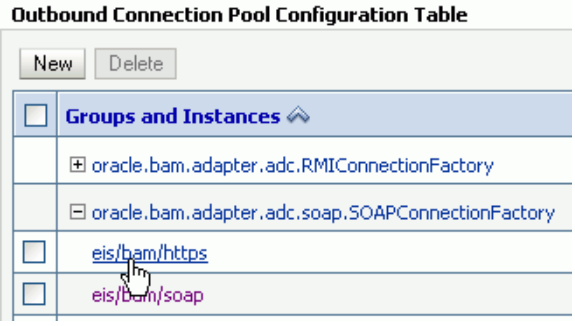
3. Select the **oracle.bam.adapter.adc.soap.SOAPConnectionFactory** option and click **Next**.



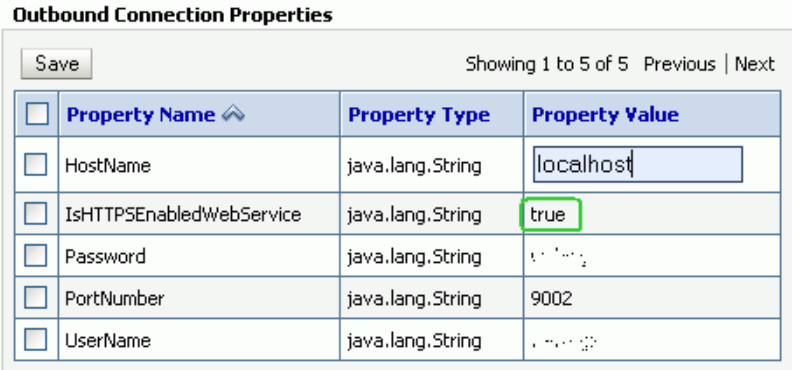
4. Enter a JNDI name for this connection factory and click **Finish**.



5. Select the **eis/bam/https** connection pool instance in the **Groups and Instances** list.



6. Complete the configuration properties as shown in Section 23.7, "Configuring Oracle BAM Adapter," and change the **IsHTTPSEnabledWebService** value to `true`.

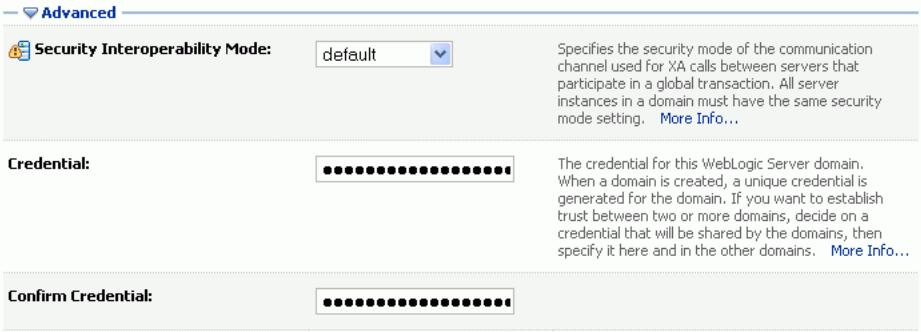


23.7.3 Configuring Trusted Domains

When using the RMI connection between a SOA composite application and Oracle BAM Server, that is when they are deployed in different domains, trusted domain configuration must be done in Oracle WebLogic Server Administration Console.

To configure trusted domains:

1. Using the Oracle WebLogic Server Administration Console, in both SOA and Oracle BAM Server domains, go to ***base_domain*** > **Security** > **General**, and expand the **Advanced** settings.



2. Enter the credential for the Oracle WebLogic Server domain in the **Credential** and **Confirm Credential** fields.

   Enter the same credential in both the SOA and Oracle BAM Server domains to establish trust between the domains.

3. Save and activate the changes, then restart Oracle WebLogic Server.

For more information about trusted domains see "Enabling Global Trust" in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

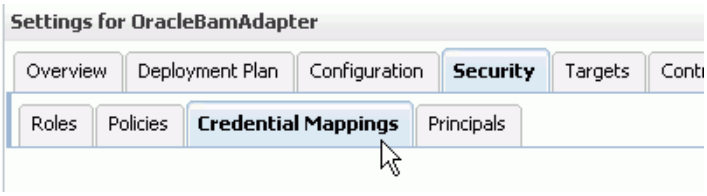
23.7.4 Configuring Credential Mapping

Configure Oracle BAM Adapter to securely store user name and password properties using Oracle WebLogic Server credential mapping functionality.

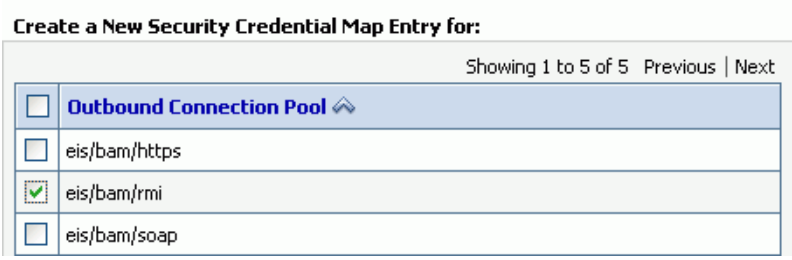
Note: Use plain text user names and passwords only in nonproduction mode. Do not mix using credential mapping and plain text user information; apply one at a time to avoid confusion.

To configure credential mapping:

1. Using the Oracle WebLogic Server Administration Console, go to **Deployment** > **OracleBamAdapter** > **Security** > **Credential Mappings**.

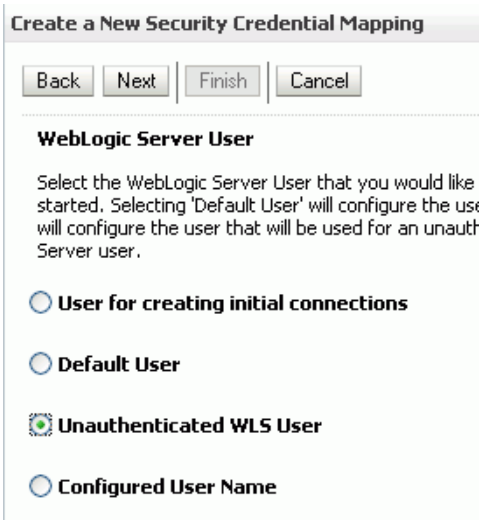


2. Click **New** and select an outbound connection pool instance for which to create a credential mapping entry, then click **Next** (By default, Oracle BAM is installed with two connection pools, **eis/bam/rmi** and **eis/bam/soap**).



> **Note:** Ensure that a separate credential mapping entry for each outbound connection pool is created. If there are any additional outbound connection pools other than the defaults, you must create corresponding credential mappings for them as well.

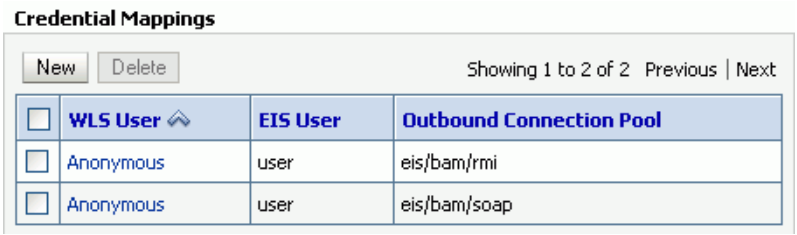
3. Select **Unauthenticated WLS User**, and click **Next** (**Unauthenticated WLS User** is similar to an anonymous user).



4. Specify a corresponding user name and password for connecting to Oracle BAM Server, then click **Finish** to complete the credential mapping.

5. Repeat these steps to create the credential mapping for the other Oracle BAM connection pool entries.

   After you configure the credential mappings for each of the outbound connection pool entries, the mappings appear in the **Credential Mappings** table.



6. For these changes to take effect, Oracle WebLogic Server must be restarted.

23.8 Configuring Oracle BAM Batching Properties

The batch processor batches operations (for example, insert, update, upsert, and delete) between the client and the Active Data Cache server to improve performance by limiting the number of remote calls. For example, 10 update operations could be processed in a single remote call with batching enabled, rather than making 10 remote calls.

> **Note:** Batching is used only on incoming data, not on internal Oracle BAM processes.

For Oracle BAM Adapter, the batching properties are configured using the Oracle BAM Adapter configuration page in the Oracle WebLogic Server Administration Console.

Oracle BAM batching properties are located in the `BAMWebConfig.xml` file for configuring batching to send data from Oracle BAM web applications (that is, to send data using Oracle BAM Architect), and to send data from the Oracle BAM `DataObjectOperation` web service using the batch method.

Oracle BAM configuration files are located in the following directory:

```
WL_HOME/user_projects/domains/base_domain/config/fmwconfig/servers/bam_server
1/applications/oracle-bam_11.1.1/config/
```

Oracle BAM must be restarted after any changes to Oracle BAM properties. For information about restarting Oracle BAM, see Section 25.2, "Managing Oracle BAM Availability".

> **Note:** The Oracle BAM ADC API properties are the properties used by all Oracle BAM clients (such as ICommand and Oracle Data Integrator), and the SOAP properties are only used by Oracle BAM Adapter when configured to use SOAP.

Table 23–2 lists the batching properties for Oracle BAM ADC API and SOAP.

Table 23–2 Active Data Cache API and SOAP (Oracle BAM Adapter) Batching Properties

Oracle BAM ADC API Property	SOAP Property	Default	Description
ActiveDataCache_Datasets_Batching_Limit_PendingCalls	Adapter_SOAP_Batching_Limit_PendingCalls	10	The limit of the number of pending calls allowed.
ActiveDataCache_Datasets_Batching_Limit_Lower	Adapter_SOAP_Batching_Limit_Lower	1000	The minimum number of elements in a batch before it is sent out.
ActiveDataCache_Datasets_Batching_Limit_Upper	Adapter_SOAP_Batching_Limit_Upper	5000	The maximum number of elements in a batch before it is sent out.
ActiveDataCache_Datasets_Batching_Timeout	Adapter_SOAP_Batching_Timeout	5000	The timeout interval in milliseconds after which the batch is sent out even if it is not full.
ActiveDataCache_Datasets_Batching_BlockOnBatchFull	Adapter_SOAP_Batching_BlockOnBatchFull	`false`	The block on batch full properties indicate whether batching operations should block when the batch is full. If set to `true`, then batching calls block until room is freed up in the batch for the operation. If set to `false`, a `BatchIsFullException` exception is thrown.
ActiveDataCache_Datasets_Batching_RetryInterval	Adapter_SOAP_Batching_RetryInterval	30000	The interval in milliseconds at which batch processing should attempt to retry failed flush attempts.
ActiveDataCache_Datasets_Batching_MaxRetryInterval	Adapter_SOAP_Batching_MaxRetryInterval	3600000	The maximum interval in milliseconds at which batch processing should attempt to retry failed flush attempts.

Table 23–2 (Cont.) Active Data Cache API and SOAP (Oracle BAM Adapter) Batching

Oracle BAM ADC API Property	SOAP Property	Default	Description
ActiveDataCache_Datasets_Batching_MaximumRetries	Adapter_SOAP_Batching_MaximumRetries	`Integer.MAX_VALUE`	The maximum number of times batch processing should attempt to retry failed flush attempts. Zero indicates no retry.
ActiveDataCache_Datasets_Batching_SuspendOnDisconnect	Adapter_SOAP_Batching_SuspendOnDisconnect	`false`	Suspend on disconnect, which indicates whether batching operations should be permanently suspended if and when the connection to Oracle BAM Server is lost.

Batching Retry Mechanism

The batching retry mechanism is designed to retry remote calls to the server in the event communication with the server is lost due to, for example, network problems or Oracle BAM Server going offline. The operation of this retry mechanism is governed by the configuration properties detailed in this section. The term *flush* refers to the attempt to send the batch calls to the Oracle BAM Server for processing.

The `RetryInterval` property specifies the interval between retry attempts in milliseconds. If the `MaxRetryInterval` value is unspecified (0), then the code continues retrying at the `RetryInterval` value until the batch flush is successful or the `MaximumRetries` value is exceeded.

If the `MaxRetryInterval` value is specified, then the retry interval value is doubled on each retry up to the `MaxRetryInterval` value. It stays at the `MaxRetryInterval` rate until the batch flush is successful, or the `MaximumRetries` value is exceeded. This allows the retry rate to decline over time to a more reasonable long-term rate. If the server does not recover quickly, do not continue to retry at a rapid rate.

If the `MaximumRetries` value is `0`, then no retries are attempted and failed batches are discarded.

If the `SuspendOnDisconnect` value is `true`, then all batching is permanently suspended until the client application is restarted, regardless of whether communication with Oracle BAM Server is reestablished.

If the pending batch limit is reached and the current batch is full (maximum size reached), then batch calls are blocked. There is no place to queue the operation from the current call, so Oracle BAM blocks the client until the condition resolves itself.

Table 23–2 describes the configuration properties that can be specified for the Active Data Cache API batching (which is used by the RMI connection factory, configured in Section 23.7.2, "Configuring Oracle BAM Connection Factories").

The Oracle BAM Adapter batching mechanism (used by the SOAP connection factory) has an identical set of configuration properties and default values.

23.9 Configuring Security

Security is configured in Oracle WebLogic Server. For information about the topics listed in this section, see *Oracle Fusion Middleware Securing Oracle WebLogic Server*. General information about Oracle WebLogic Server security is available in *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server*.

This section address the following security topics as they relate to Oracle BAM:

- Section 23.9.1, "Configuring Credential Mapping"
- Section 23.9.2, "Configuring Oracle BAM User Permissions"
- Section 23.9.3, "Configuring Secure Socket Layer"
- Section 23.9.4, "Using Oracle Internet Directory With Oracle BAM"
- Section 23.9.5, "Securing Oracle BAM JMS Resources"
- Section 23.9.6, "Calling Secure Web Services"

23.9.1 Configuring Credential Mapping

Credential mapping is used to securely store user name and password properties when using the Oracle BAM Adapter. For information see Section 23.7.4, "Configuring Credential Mapping."

23.9.2 Configuring Oracle BAM User Permissions

To provide secure access to the Oracle BAM applications, users are assigned to roles that provide the necessary permissions. For more information, see Section 25.3, "Managing Oracle BAM Users".

23.9.3 Configuring Secure Socket Layer

Oracle WebLogic Server provides the facilities needed to enable Secure Socket Layer (SSL) on any Oracle SOA Suite and Oracle WebCenter Portal connections into Oracle WebLogic Server.

Use the Java Development Kit (JDK) `keytool` utility to create and manage keystores and certificates, and use the Oracle WebLogic Server Administration Console to configure Oracle WebLogic Server listeners.

For more information, see "Configuring SSL" in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

Although Oracle WebLogic Server lets you configure the Oracle BAM web applications on both SSL and non-SSL ports, and Oracle BAM can also be configured to run on both SSL and non-SSL ports, it causes issues with the URL configured for Oracle BAM alerts. When Oracle BAM is running on both SSL and non-SSL ports, it cannot send two Oracle BAM report URLs to the alert email receiver. It is the responsibility of the administrator to decide which mode of access (SSL or non-SSL) to use for the Oracle BAM report URL recipients.

To start Oracle BAM in SSL mode, and enable Oracle BAM clients (like ICommand) to invoke Oracle BAM ADC, invoke EJBs, JMS resources, and JDBC resources using SSL, you must configure the following Oracle BAM properties:

- `Communication_Protocol` in the `BAMCommonConfig.xml`, `BAMServerConfig.xml`, and `BAMICommandConfig.xml` files, as follows:

Default value: `<Communication_Protocol>t3</Communication_Protocol>`

For SSL using t3s: `<Communication_Protocol>t3s</Communication_Protocol>`

- `ListenPort` in `BAMCommonConfig.xml` file, as follows:

  Default value: `<ListenPort>ListenPort</ListenPort>`

  For SSL: `<ListenPort>SSLListenPort</ListenPort>`

23.9.4 Using Oracle Internet Directory With Oracle BAM

For general information about configuring Oracle Internet Directory, see "Configuring Users and Groups in the Oracle Internet Directory and Oracle Virtual Directory Authentication Providers" in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

To use Oracle Internet Directory with Oracle BAM:

1. Configure Oracle Internet Directory using the Oracle WebLogic Server Administration Console.
2. Create **OracleSystemUser** in Oracle Internet Directory, by connecting to Oracle Internet Directory with a Lightweight Directory Access Protocol (LDAP) browser and creating a user in the same base user Distinguished Name that was provided in the Oracle WebLogic Server Administration Console while configuring Oracle Internet Directory.
3. Run the following SQL statements to nullify the user global unique identifiers (GUID) in the Oracle BAM schema:

```
UPDATE "SysIterUser" SET "SysIterUser"."GUID" = NULL,
"SysIterUser"."Inactive" = NULL;
```

23.9.5 Securing Oracle BAM JMS Resources

This is an optional procedure to restrict access to the Oracle BAM system JMS topics.

To secure Oracle BAM JMS resources:

1. Create a group with any group name using Oracle WebLogic Server Administration Console. For example, `OracleSystemGroup`.
2. Go to *`WL_HOME`*`/AS11gR1SOA/bam/bin` and invoke `secure_jms_system_resource.py` using Oracle WebLogic Scripting Tool (WLST).

   The usage should be:

```
<JAVA_HOME>/bin/java -classpath <weblogic.jar_location> weblogic.WLST
./secure_jms_system_resource.py --username admin_user_name --password
admin_password --url admin_server_t3_url --jmsSystemResource
JMS_system_resource_name --group security_group_to_use
```

3. Assign OracleSystemUser to this group.
4. Assign all those Oracle BAM users to this group.

23.9.6 Calling Secure Web Services

Currently, the Oracle BAM Alerts **Call a Web Service** action supports invoking web services protected by one-way SSL. In one-way SSL, the web service to be invoked

presents a digital certificate to its client (the Oracle BAM Alert action client in this case), and the client verifies whether certificate presented to it can be trusted. If the client trusts the server certificate, then further communication between client and server continues over a secure SSL channel.

To accomplish one-way SSL, the Alert web service client must be pointed to a trust store in which it can perform a lookup, to determine if the certificate presented to it exists in it or not. This can be done by adding the following properties in `BAMCommonConfig.xml`.

```
<SSL.TrustStore.Location>wls.home/lib/DemoTrust.jks</SSL.TrustStore.Location>
<SSL.TrustStore.Type>JKS</SSL.TrustStore.Type>
<SSL.TrustStore.Password>password</SSL.TrustStore.Password>
<SSL.TrustStore.Provider>SUN</SSL.TrustStore.Provider>
```

By default, if these properties are not specified in the configuration file, the values mentioned above are used (except for *`password`*, in which no value is used). You can provide different values to these properties to override the default trust store configuration. Currently you must edit the configuration files to set these properties, as they are not exposed in Oracle Enterprise Manager Fusion Middleware Control.

The default trust store is Java Key Store (JKS), which is file based. Certificates can be added and deleted using the Java `keytool` command-line utility.

For information about configuring the Call a Web Service alert action, see "Call a Web Service" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

23.9.6.1 Example: Protecting Oracle BAM Web Services

The following is an example illustrating how to invoke SSL protected Oracle BAM web services. It is specific to the **Call a Web Service** alert action calling the Oracle BAM `ManualRuleFire` web service. The steps in this section do not need to be done if you are invoking external one-way SSL-protected non-BAM web services.

> **Note:** After this configuration the entire Oracle BAM application is behind the SSL layer and must be accessed using the SSL port number and HTTPS protocol.

To configure properties for calling secure web services:

1. Open Oracle WebLogic Server Administration Console and go to **Domain Structure** > **Environment** > **Servers** > ***BAM_Server*** > **General**.
2. Check **SSL Listen Port** checkbox and specify the **SSL Listen Port** number to use.
3. Modify the Oracle BAM configuration properties.
   a. These properties must be modified in `BAMCommonConfig.xml`:

      ```
      <Communication_Protocol>t3s</Communication_Protocol>
      <ListenPort>SSLListenPort</ListenPort>
      <ApplicationURL>http://SSL_HOST:SSL_PORT</ApplicationURL>
      ```

   b. This property must be modified in `BAMWebConfig.xml`:

      ```
      <ServerPort>SSL_PORT</ServerPort>
      ```

4. Restart Oracle WebLogic Server.
5. Open Oracle BAM Active Studio or Oracle BAM Architect, and go to **Alerts**.

6. Create an alert with the **Call a Web Service** action using a secure web service, such as

   `https://host_name:SSL_port_number/OracleBAMWS/WebServices/ManualRuleFire?wsdl`

   as the WSDL URL for the action.

7. Map the parameters and save the alert.

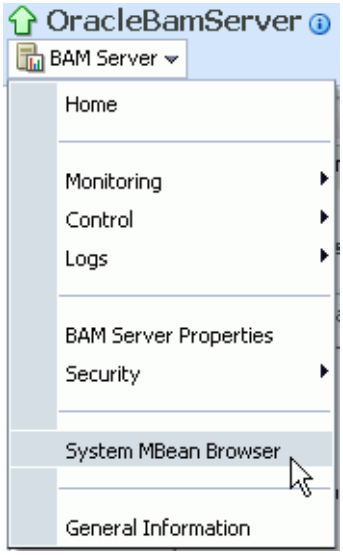
23.10 Configuring Advanced Properties

Oracle BAM provides many advanced properties not available for configuration using the BAM Web Properties and BAM Server Properties pages in Fusion Middleware Control. These advanced properties are configured using the System MBean Browser.

For a listing of all properties available for Oracle BAM configuration, see Section 23.11, "Oracle BAM Configuration Property Reference".

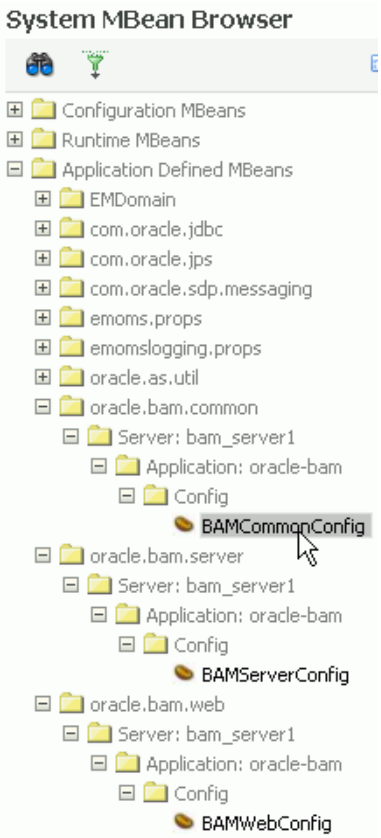
To configure the advanced properties using the System MBean Browser:

1. Choose **System MBean Browser** from the menu on the OracleBAMWeb or OracleBAMServer page.

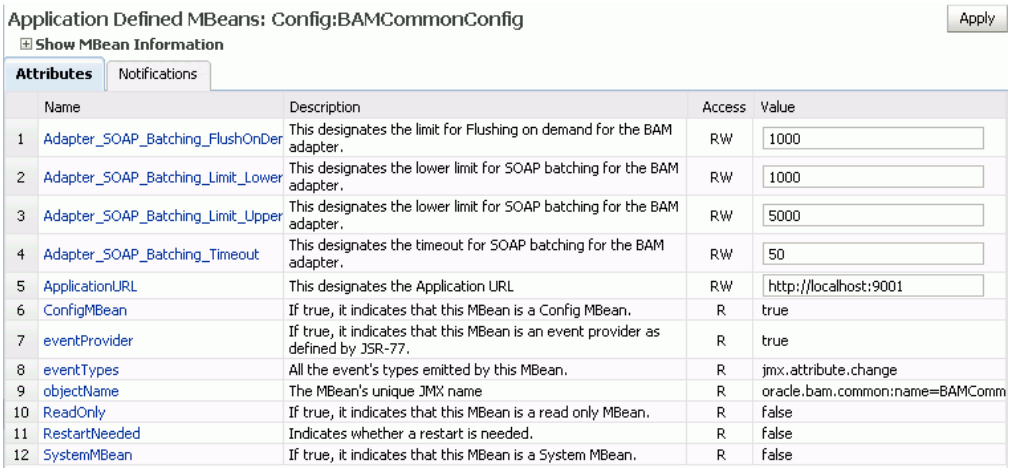


2. Select the path to the appropriate Application Defined MBeans page:
   - For properties common to all Oracle BAM components select **Application Defined MBeans** > **oracle.bam.common** > **Server: bam_server1** > **Application: oracle-bam** > **Config** > **BAMCommonConfig** (see Figure 23–1).
   - For Oracle BAM Server-specific properties select **Application Defined MBeans** > **oracle.bam.server** > **Server: bam_server1** > **Application: oracle-bam** > **Config** > **BAMServerConfig**.
   - For Oracle BAM web applications-specific properties select **Application Defined MBeans** > **oracle.bam.web** > **Server: bam_server1** > **Application: oracle-bam** > **Config** > **BAMWebConfig**.

Figure 23–1 System MBean Browser Navigation Tree



The associated configuration properties are displayed in the System MBean Browser.



Application Defined MBeans: Config:BAMCommonConfig

Apply

Show MBean Information

Attributes | Notifications

	Name	Description	Access	Value
1	Adapter_SOAP_Batching_FlushOnDer	This designates the limit for Flushing on demand for the BAM adapter.	RW	1000
2	Adapter_SOAP_Batching_Limit_Lower	This designates the lower limit for SOAP batching for the BAM adapter.	RW	1000
3	Adapter_SOAP_Batching_Limit_Upper	This designates the lower limit for SOAP batching for the BAM adapter.	RW	5000
4	Adapter_SOAP_Batching_Timeout	This designates the timeout for SOAP batching for the BAM adapter.	RW	50
5	ApplicationURL	This designates the Application URL	RW	http://localhost:9001
6	ConfigMBean	If true, it indicates that this MBean is a Config MBean.	R	true
7	eventProvider	If true, it indicates that this MBean is an event provider as defined by JSR-77.	R	true
8	eventTypes	All the event's types emitted by this MBean.	R	jmx.attribute.change
9	objectName	The MBean's unique JMX name	R	oracle.bam.common:name=BAMComm
10	ReadOnly	If true, it indicates that this MBean is a read only MBean.	R	false
11	RestartNeeded	Indicates whether a restart is needed.	R	false
12	SystemMBean	If true, it indicates that this MBean is a System MBean.	R	false

3. Edit property values in the **Values** column, and click **Apply**.
4. Oracle BAM must be restarted after any changes to Oracle BAM properties. For information about restarting Oracle BAM, see Section 25.2, "Managing Oracle BAM Availability".

Oracle BAM Configuration Property Files

All Oracle BAM properties are located in configuration files. These files are located in the following directory on the host where the Oracle BAM components are installed:

```
WL_HOME/user_projects/domains/base_domain/config/fmwconfig/servers/bam_server
1/applications/oracle-bam_11.1.1/config
```

The configuration properties usually go into the `BAMCommonConfig.xml` file, which contains the properties global to all Oracle BAM components. These properties can also be specified in component-specific configuration files:

- Oracle BAM Server-specific properties are configured in `BAMServerConfig.xml`
- Oracle BAM web applications-specific properties are configured in `BAMWebConfig.xml`
- Oracle BAM ICommand utility-specific properties are configured in `BAMICommandConfig.xml`

The properties set in the `BAMCommonConfig.xml` file are always loaded. Any additional configuration files must be loaded explicitly. Oracle BAM Server explicitly loads the `BAMServerConfig.xml` file, Oracle BAM web applications explicitly load the `BAMWebConfig.xml` file, and ICommand explicitly loads the `BAMICommandConfig.xml` file. When a configuration file is loaded, its properties override any properties previously set, that is properties specified in `BAMServerConfig.xml` override properties set in `BAMCommonConfig.xml`.

23.11 Oracle BAM Configuration Property Reference

This section provides a brief description of each Oracle BAM configuration property that may be used in the Oracle BAM configuration files.

Advanced properties are configured in the Fusion Middleware Control System MBean Browser. For information about configuring properties that do not appear in the Oracle BAM configuration pages in Fusion Middleware Control, see Section 23.10, "Configuring Advanced Properties".

Some of these properties correspond to properties you can set in the Fusion Middleware Control and Oracle WebLogic Server Administration Console, and references to specific sections about these configuration procedures are provided in Table 23–3.

Note: The Oracle BAM ADC API properties are the properties used by all Oracle BAM clients (such as ICommand and Oracle Data Integrator), and the SOAP properties are only used by Oracle BAM Adapter when configured to use SOAP.

Table 23–3 Oracle BAM Configuration Properties

Property Name	Description
`ActiveDataCache.API.Batching.ThreadPool.CoreSize`	Indicates how many threads to keep in the Oracle BAM ADC API thread pools when idle. Default: 5
`ActiveDataCache.API.Batching.ThreadPool.KeepAliveTimeInSecs`	Amount of time (in seconds) to keep an idle thread in the Oracle BAM ADC API thread pools. Default: 20
`ActiveDataCache.API.Batching.ThreadPool.MaxSize`	Maximum number of threads in the Oracle BAM ADC API thread pools. Default: 100
`ActiveDataCache.API.Batching.ThreadPool.QueueSize`	Maximum number of messages that can be queued in the Oracle BAM ADC API thread pools. Default: 30
`ActiveDataCache.API.BufferedDataReader.ThreadPool.CoreSize`	Indicates how many threads to keep in the Oracle BAM ADC buffered data reader thread pools when idle. Default: 10
`ActiveDataCache.API.BufferedDataReader.ThreadPool.KeepAliveTimeInSecs`	Amount of time (in seconds) to keep an idle thread in the Oracle BAM ADC buffered data reader thread pools. Default: 20
`ActiveDataCache.API.BufferedDataReader.ThreadPool.MaxSize`	Maximum number of threads in the Oracle BAM ADC Buffered Data Reader Thread Pools. Default: 100
`ActiveDataCache.API.BufferedDataReader.ThreadPool.QueueSize`	Maximum number of messages that can be queued in the Oracle BAM ADC Buffered Data Reader Thread Pools. Default: 30
`ActiveDataCache.Kernel.ActiveDataQueue.ThreadPool.CoreSize`	Indicates how many threads to keep in the Oracle BAM ADC Active Data Queue Thread Pools when idle. Default: 20
`ActiveDataCache.Kernel.ActiveDataQueue.ThreadPool.KeepAliveTimeInSecs`	Amount of time (in seconds) to keep an idle thread in the Oracle BAM ADC Active Data Queue Thread Pools. Default: 20
`ActiveDataCache.Kernel.ActiveDataQueue.ThreadPool.MaxSize`	Maximum number of threads in the Oracle BAM ADC Active Data Queue Thread Pools. Default: 100
`ActiveDataCache.Kernel.ActiveDataQueue.ThreadPool.QueueSize`	Maximum number of messages that can be queued in the Oracle BAM ADC Active Data Queue Thread Pools. Default: 30
`ActiveDataCache_Datasets_Batching_BlockOnBatchFull`	Block on batch full properties indicate whether batching operations should block when the batch is full. If set to `true`, then batching calls block until room is freed up in the batch for the operation. If set to `false`, a `BatchIsFullException` exception is thrown. Default: `false`

Table 23–3 (Cont.) Oracle BAM Configuration Properties

Property Name	Description
`ActiveDataCache_Datasets_Batching_Limit_Lower`	The minimum amount of elements in a batch before it is sent to the server. Depending on the load, batches may contain more than this minimum (up to the maximum), but they never contain less than this minimum (unless the timeout expires). See `ActiveDataCache_Datasets_Batching_Timeout` for the exception to this rule. See Section 23.8, "Configuring Oracle BAM Batching Properties" for more information. Default: 1000
`ActiveDataCache_Datasets_Batching_Limit_PendingCalls`	The limit of the number of pending batches that are waiting to be sent to the server. When this limit is reached and the current batch is full (at its upper limit), further batched operations become blocking calls until this condition is cleared. This limit throttles the client and keeps it from overrunning the server. See Section 23.8, "Configuring Oracle BAM Batching Properties" for more information. Default: 10
`ActiveDataCache_Datasets_Batching_Limit_Upper`	The maximum amount of elements in a batch before it is sent out. A batch never contains more than this maximum. See Section 23.8, "Configuring Oracle BAM Batching Properties" for more information. Default: 5000
`ActiveDataCache_Datasets_Batching_MaximumRetries`	The maximum number of times batch processing retries failed attempts to send the batch to the server. A value of 0 (zero) indicates no retry. Default: 2147483647
`ActiveDataCache_Datasets_Batching_MaxRetryInterval`	The maximum interval in milliseconds at which batch processing retries failed attempts to send the batch to the server. If this maximum retry interval is specified, the retry interval is doubled on each retry attempt until the interval reaches this maximum value. Further retries use this maximum value. This interval enables the configuration of retry attempts that taper off should the failure not resolve itself quickly. If a maximum retry interval of 0 (zero) is specified, the retry interval does not change. Default: 3600000
`ActiveDataCache_Datasets_Batching_RetryInterval`	The interval in milliseconds at which batch processing retries failed attempts to send the batch to the server. Retry is attempted when the failure is due to server connectivity problems such as the server being restarted. Default: 30000
`ActiveDataCache_Datasets_Batching_SuspendOnDisconnect`	Indicates whether batching operations are permanently suspended if and when the connection to Oracle BAM Server is lost. Default: `false`
`ActiveDataCache_Datasets_Batching_ThreadPoolSize`	The constant size thread pool (that is, containing a fixed number of threads). Default: 25

Table 23–3 (Cont.) Oracle BAM Configuration Properties

Property Name	Description
`ActiveDataCache_Datasets_Batching_Timeout`	The timeout in milliseconds after which the batch is sent out even if it is not full (has less than the configured minimum in `ActiveDataCache_Datasets_Batching_Limit_Lower`). The timeout ensures that operations are not waiting overlong in a batch due to inactivity. A batched operation never waits longer than this timeout to be sent to the server. See Section 23.8, "Configuring Oracle BAM Batching Properties" for more information. Default: 50
`ActiveDataCache_Query_Command_Timeout`	The configuration key for a query command timeout in seconds. If the time to execute a query command exceeds this timeout, an exception is thrown. Default: 60
`ActiveDataCache_Update_Command_Timeout`	The configuration key for an update command timeout in seconds. If the time to execute an update command exceeds this timeout, an exception is thrown. Default: 60
`Adapter_SOAP_Batching_BlockOnBatchFull`	Block on batch full properties indicate whether batching operations should block when the batch is full. If set to `true`, then batching calls block until room is freed up in the batch for the operation. If set to `false`, a `BatchIsFullException` exception is thrown. Default: `false`
`Adapter_SOAP_Batching_Limit_Lower`	The minimum amount of elements in a batch before it is sent out. See Section 23.8, "Configuring Oracle BAM Batching Properties" for more information. Default: 1000
`Adapter_SOAP_Batching_Limit_Upper`	The maximum amount of elements in a batch before it is sent out. See Section 23.8, "Configuring Oracle BAM Batching Properties" for more information. Default: 5000
`Adapter_SOAP_Batching_Timeout`	The timeout interval in milliseconds after which the batch is sent out even if it is not full. See Section 23.8, "Configuring Oracle BAM Batching Properties" for more information. Default: 50
`ADC_PassThrough_Import_BatchSize`	The maximum number of rows that are imported from an external data object. Default: 50
`ADC_PassThrough_Import_MaxRowsToImportAtOnce`	Maximum number. of rows imported by `ExternalDataManager` at one time. Default: 100
`ADC_PassThrough_Import_MaxSize`	Maximum size imported by `ExternalDataManager`. Default: -1

Table 23–3 (Cont.) Oracle BAM Configuration Properties

Property Name	Description
`ADCBatchSize`	Refers to JDBC statement batching. When the Oracle BAM ADC executes SQL statements it uses JDBC batching to reduce the number of calls to the database, improving performance. Each call executes a batch of statements rather than a single statement. Default: 100
`ADCChannelName`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: `invm:topic/oracle.bam.messaging.activedatacache.activedata`
`ADCDataSource`	The data source for the Active Data Cache. See Section 23.3.2, "Configuring the Data Source JNDI" for more information. Default: `jdbc/oracle/bam/adc`
`ADCLogBatchInsertLocks`	Indicates whether to log Oracle BAM ADC batch insert locks. Default: false
`ADCLogFolderLocks`	Indicates whether to log Oracle BAM ADC folder locks. Default: false
`ADCLogSqlStatements`	Indicates whether to log Oracle BAM ADC SQL statements. Default: false
`ADCLogTransactionLocks`	Indicates whether to log Oracle BAM ADC transaction locks. Default: false
`ADCMaxViewsetRowCount`	The default limit for rows of data displayed in a view with this property. The higher you set this value, the more performance is impacted. Default: 64000
`ADCPreloadDataObjectIDs`	Indicates whether data objects should be loaded at Oracle BAM ADC startup rather than on-demand as they are accessed. Pre-loading the data objects increases startup time but decreases initial data object access time. Not pre-loading the data objects decreases startup time but increases initial data object access time. Also, pre-loading all data objects where many data objects are rarely referenced may needlessly increase memory requirements.
`ADCPushInterval`	The rate at which Oracle BAM Active Data Cache pushes events to Oracle BAM Report Server. This is one factor that affects the frequency at which active events occur on Oracle BAM dashboard pages. Increasing this interval reduces the load on Oracle BAM Server. However, larger intervals increase the likelihood of multiple updates in the dashboard being collapsed into single updates. You can override the default `ADCPushInterval` within particular reports by setting the Active Data Retrieval Interval property in the Report Properties dialog for a particular Oracle BAM dashboard or report. Open a report in Active Studio, then select **Edit** > **Change Report Properties** > **Advanced** to set the active data retrieval interval. Nonzero values override the `ADCPushInterval` value for that particular report. Default: 1000
`ADCPushThreadPoolSize`	Indicates the size of the threadpool for Oracle BAM ADC push functionality. Default: 10

Table 23–3 (Cont.) Oracle BAM Configuration Properties

Property Name	Description
`ADCRetryCount`	Number of times to retry the Oracle BAM Active Data Cache connection. Default: 2
`ADCSecurityFiltersEnabled`	Indicates whether Oracle BAM ADC security filters are to be enabled. Default: true
`ADCViewsetTimeoutSec`	Specifies the timeout for retrieving a view set. Default: 0
`ADCWaitTime`	Interval between Active Data Cache connection attempts. Default: 20
`ApplicationURL`	URL for Oracle BAM web applications and Oracle BAM Server. See Section 23.3.3, "Configuring the Application URL" and Section 23.2.2, "Configuring the Application URL" for more information. Default: `DEFAULT`
`Architect_Content_PageSize`	Controls how many rows are shown when viewing data object contents in Oracle BAM Architect. Default: 100
`AutoIncrementBlockSize`	Indicates amount of increment for block size. Default: 100
`CacheDirectory`	Designates the directory used by Report Cache to cache snapshots and active data. Default: `cache`
`ChangeListDelay`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: 1
`ChannelName`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: `OracleBAM`
`CheckViewsFallingBehindInterval`	Indicates the interval to check whether views are falling behind. Default: 20
`CollaborationNILogin`	Whether NI Login is used. Default: false
`ColumnarTimeout`	Indicates the columnar timeout. Default: 0
`Communication_Protocol`	Specifies the communication protocol for JNDI communication. For SSL using t3: `<Communication_Protocol>t3s</Communication_Protocol>` See Section 23.9.3, "Configuring Secure Socket Layer." Default: `DEFAULT`
`ConnectionFactoryName`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: `jms/QueueConnectionFactory`

Table 23–3 (Cont.) Oracle BAM Configuration Properties

Property Name	Description
`DeletesActiveDataSize`	`InsertsActiveDataSize`, `UpdatesActiveDataSize`, `UpsertsActiveDataSize`, and `DeletesActiveDataSize` determine how many inserts, updates, upserts, and deletes, respectively, should be executed before releasing and then requiring the lock on the data object being modified. This keeps the operation from holding the data object lock for long periods of time when there are view sets open on this data object. Default: 50
`DisplayUIStackTrace`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. This property enables user interface stack traces (by setting this property value to `true`). This property is used for temporary diagnostic purposes only because exposing a stack with internal details is a potential security threat. Default: `false`
`DistributionListGroupType`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: *none*
`EDS_Query_Timeout`	The timeout in seconds after which the execution of BAM report queries on external data objects stop because it is taking too long to retrieve the information for display. Also set `Import_MaxSize`, which determines the maximum number of rows that are pulled from the external data object (the larger the number, the slower the performance). Default: 180
`ElementsCountLimit`	The number of change lists that are cached before rewriting the cache file. The default value is 50. When the 51st change list is received, it is cached by replacing the 1st change list. Default: 50
`ElementsSizeLimit`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: 500
`EmailIDForSender`	Email address that appears in the From header of notifications sent by the Event Engine. See Section 23.3.7, "Configuring the Outbound Email Account" for more information. Default: `BAM-NOTIFICATION_AR@oracle.com`
`EMSConnectionRecoveryDuration`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services.
`EnableADCDatasetLockTimeout`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Indicates whether data set lock timeout is enabled. Default: `false`
`EnableDataTruncationMsg`	Indicates whether to enable the data truncation message. Default: `false`
`EnableGetAllViewsInOneRoundTrip`	Allows all views to be retrieved in one round trip. Default: `true`

Table 23–3 (Cont.) Oracle BAM Configuration Properties

Property Name	Description
`EvaluatorMaxExpressionDepth`	Maximum number of nested logical expressions that can be included in one method in the class that gets compiled to evaluate a filter expression. Default: 200 When opening a view that uses a filter with a large number of distinct values, the compiler could run out of memory and you would get the error: "BAM-01317: Compilation for class ... failed with error 3." If this error continues to happen, lower the value for EvaluatorMaxExpressionDepth and restart the server. Although this property does not significantly affect performance, you can turn it off by setting EvaluatorMaxExpressionDepth to zero, which would result in one method in the evaluator class for the entire filter expression.
`EventEngine.Action.ThreadPool.CoreSize`	Indicates how many threads to keep in the Event Engine Action Thread Pools when idle. Default: 4
`EventEngine.Action.ThreadPool.KeepAliveTimeInSec`	Amount of time (in seconds) to keep an idle thread in the Event Engine Action Thread Pools. Default: 20
`EventEngine.Action.ThreadPool.MaxSize`	Maximum number of threads in the Event Engine Action Thread Pools. Default: 100
`EventEngine.Action.ThreadPool.QueueSize`	Maximum number of messages that can be queued in the Event Engine Action Thread Pools. Default: 4
`EventEngine.AlertHistory.ThreadPool.CoreSize`	Indicates how many threads to keep in the Event Engine Alert History Thread Pools when idle. Default: 4
`EventEngine.AlertHistory.ThreadPool.KeepAliveTimeInSecs`	Amount of time (in seconds) to keep an idle thread in the Event Engine Action Thread Pools. Default: 20
`EventEngine.AlertHistory.ThreadPool.MaxSize`	Maximum number of threads in the Event Engine Alert History Thread Pools. Default: 100
`EventEngine.AlertHistory.ThreadPool.QueueSize`	Maximum number of messages that can be queued in the Event Engine Rule Fire Thread Pools. Default: 10
`EventEngineAlertHistoryEventLogging`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: `false`
`EventEngineAlertHistoryRecordsPerWrite`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: 10
`EventEngine.RuleFire.ThreadPool.CoreSize`	Indicates how many threads to keep in the Event Engine Rule Fire Thread Pools when idle. Default: 4

Table 23–3 (Cont.) Oracle BAM Configuration Properties

Property Name	Description
`EventEngine.RuleFire.ThreadPool.KeepAliveTimeInSecs`	Amount of time (in seconds) to keep an idle thread in the Event Engine Rule Fire Thread Pools. Default: 20
`EventEngine.RuleFire.ThreadPool.MaxSize`	Maximum number of threads in the Event Engine Rule Fire Thread Pools. Default: 100
`EventEngine.RuleFire.ThreadPool.QueueSize`	Maximum number of messages that can be queued in the Event Engine Rule Fire Thread Pools. Default: 4
`EventEngineCheckADCIntervalSecs`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: 5
`EventEngineMaxConnections`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: 25
`EventEngineMaxShutdownWaitInterval`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: 10
`EventEngineSaveLastFireIntervalSecs`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: 10
`FileHandleCacheSize`	Used by Oracle BAM Report Cache to cache file *handles* of the files used for caching when file-based persistence is used (see Section 23.3.5, "Configuring the Report Cache Persistence Manager" for information about persistence management). Handle caching increases the speed of reading and writing active data and snapshots. Each view set has two file handles--one for the snapshot and another for active data. The value for this property sets the number of file handles that are cached by default. To disable file handle caching, set this property value to zero (0). The cost of file handle caching is that all of the cached file handles remain open, and so a) on servers configured for a very small number of concurrent open file handles, and b) on servers heavily loaded with open file handle count reaching the maximum value, this could create a problem. For these scenarios, reduce `FileHandleCacheSize` to an appropriate smaller value. Default: 100
`FractionInIntegerDivision`	Setting this property to `true` shows fractions in Integer divisions in calculations. Saved reports must be re-saved after a server restart (following configuration change) to see this behavior. Default: `false`
`GenericSatelliteChannelName`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: `invmjms:topic/oracle.bam.messaging.systemobjectnotification`

Table 23–3 (Cont.) Oracle BAM Configuration Properties

Property Name	Description

`GenericSatelliteSystemObjectNames`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services.
	Default values:
	`SystemObjectName=SysIterReport,MonitorColumnName=SysIterLastModified,MessageColumn1=SysIterID,MessageColumn2=SysIterLastModified;`
	
	`SystemObjectName=SysIterUser,MonitorColumnName=SpecificTimeZoneOffset,MessageColumn1=SysIterID;`
	
	`SystemObjectName=SysIterUser,MonitorColumnName=AdjustTimeZoneSetting,MessageColumn1=SysIterID;`
	
	`SystemObjectName=SysIterUser,MonitorColumnName=AdjustDaylightSavingsSetting,MessageColumn1=SysIterID;`
	
	`SystemObjectName=SysIterDataset,MonitorColumnName=SysIterLastModified,MessageColumn1=SysIterID;`

`HelpAppLocation`	This property is no longer used in Oracle BAM.
`iActiveDataCloseReportsTimeout`	Timeout to close reports. Default: 500
`iActiveDataMaxDiffCounter`	The maximum difference counter for iActiveData. Default: 10
`iActiveDataMinIntervalsToWait`	The minimum interval to wait for iActiveData. Default: 5
`iActiveDataReloadOnSecsBehindThreshold`	The threshold value for reloading iActiveData. Default: 10
`iActiveDataRestartInterval`	The restart interval for iActiveData. Default: 3000
`iActiveDataRestartTimeout`	The restart timeout value for iActiveData. Default: 10000
`iActiveDataScriptsCleanupFactor`	A property to address a Microsoft Internet Explorer memory leak, when active data is coming into the dashboard at a fast pace, by forcing periodic browser refreshes. This value may be further increased when active data is coming to the dashboard at a rate of 25 events per second or greater. You can monitor the Microsoft Internet Explorer memory consumption to determine an appropriate value. Default: 1048576
`ICommand_Default_User_Name`	Specifies default security credentials for running ICommand operations. See *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite* for more information.
`ICommand_Default_Password`	Specifies default security credentials for running ICommand operations. See *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite* for more information.
`Import_BatchSize`	Batch size for import by `ExternalDataManager`. Default: 50

Table 23–3 (Cont.) Oracle BAM Configuration Properties

Property Name	Description
`Import_MaxRowsAtOnce`	Maximum number of rows imported by `ExternalDataManager` at one time. Default: 100
`Import_MaxSize`	Maximum size imported by `ExternalDataManager`. Default: -1
`InsertsActiveDataSize`	`InsertsActiveDataSize`, `UpdatesActiveDataSize`, `UpsertsActiveDataSize`, and `DeletesActiveDataSize` determine how many inserts, updates, upserts, and deletes, respectively, should be executed before releasing and then requiring the lock on the data object being modified. This keeps the operation from holding the data object lock for long periods of time when there are view sets open on this data object. Default: 50
`ListenPort`	For SSL: `<ListenPort>SSLListenPort</ListenPort>` See Section 23.9.3, "Configuring Secure Socket Layer." Default: `ListenPort`
`MaxDBNodeFailoverRetries`	Used in Oracle RAC High Availability configuration. Indicates the number of retries Oracle BAM Server attempts in an event of an Oracle RAC failover. Default: 5
`MTTimerThreadPoolSize`	Oracle BAM has a timer implementation that dispatches timer events using a thread pool. This allows timer events to be processed in parallel. This property determines the thread pool size for this timer implementation. Default: 5
`ODIAgentHost`	The IP address or hostname of the server where the Oracle Data Integrator agent is running. See Section 23.3.6, "Configuring Oracle Data Integrator Integration Properties" for more information. Default: `localhost`
`ODIAgentPort`	The TCP port on which the Oracle Data Integrator agent is listening. See Section 23.3.6, "Configuring Oracle Data Integrator Integration Properties" for more information. Default: 20910
`ODIPassword`	The encoded Oracle Data Integrator password. See Section 23.3.6, "Configuring Oracle Data Integrator Integration Properties" for more information. Default: `SUNOPSIS`
`ODIUser`	The user that executes the scenarios in Oracle Data Integrator. See Section 23.3.6, "Configuring Oracle Data Integrator Integration Properties" for more information. Default: `SUPERVISOR`
`ODIWorkRepositoryName`	The name of the Oracle Data Integrator work repository database. See Section 23.3.6, "Configuring Oracle Data Integrator Integration Properties" for more information. Default: `WORK_DEV`

Table 23–3 (Cont.) Oracle BAM Configuration Properties

Property Name	Description
`oracle.bam.common.messaging.util.MessageConsumerRegistryQueuedPusher_CorePoolSize`	Indicates how many threads to keep in Message Registry Thread Pool when idle. Default: 20
`oracle.bam.common.messaging.util.MessageConsumerRegistryQueuedPusher_KeepAliveTime`	Amount of time (in seconds) to keep an idle thread in the Message Registry Thread Pool. Default: 60
`oracle.bam.common.messaging.util.MessageConsumerRegistryQueuedPusher_MaximumPoolSize`	Maximum number of threads in the Message Registry Thread Pool. Default: 100
`oracle.bam.common.messaging.util.MessageConsumerRegistryQueuedPusher_MaximumQueueSize`	Maximum number of messages that can be queued in the Message Registry Thread Pool. Default: 30
`RecordsLimitForEmail`	The number of rows allowed in a List view or Columnar report used in an email attachment before a limit error is displayed. Attachments containing reports can become large because they include data compared to a report link that accesses the report and data on Oracle BAM Server. Default: 1000
`RecordsLimitForSaveOffline`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: 0
`RecordsLimitForValueBrowser`	Number of values that are presented in values browser for constructing filter expressions. Default: 50
`ReportCache.ViewSets.ThreadPool.CoreSize`	Number of threads to keep in Report Cache Thread Pool when idle. Default: 5
`ReportCache.ViewSets.ThreadPool.KeepAliveTimeInSecs`	Amount of time (in seconds) to keep an idle thread in the Report Cache Thread Pool. Default: 20
`ReportCache.ViewSets.ThreadPool.MaxSize`	Maximum number of threads in the Report Cache Thread Pool. Default: 100
`ReportCache.ViewSets.ThreadPool.QueueSize`	Maximum no. of messages that can be queued in the Report Cache Thread Pool. Default: 30
`ReportCacheChannelName`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: `invmjms:topic/oracle.bam.messaging.reportcache.activedata`
`ReportCacheMaxConnections`	The maximum number of connections allowed by Report Cache to its clients. Default: 25
`ReportCacheMaxShutdownWaitInterval`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: 10

Table 23–3 (Cont.) Oracle BAM Configuration Properties

Property Name	Description
`ReportCachePersistenceManager`	See Section 23.3.5, "Configuring the Report Cache Persistence Manager" for more information. Default: `FileBasedPersistenceManager`
`ReportLoadingSetting`	The report loading indicator to be enabled. See Section 23.2.3, "Configuring the Report Loading Indicator" for more information. Default: `on`
`SensorFactory`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: `oracle.bam.common.statistics.noop.SensorFactoryImpl`
`ServerName`	The Oracle BAM Server hostname. See Section 23.2.4, "Configuring the Server Name" for more information. Default: `DEFAULT`
`ServerPort`	The Oracle BAM Server port number. Default: `DEFAULT`
`UpdatesActiveDataSize`	`InsertsActiveDataSize`, `UpdatesActiveDataSize`, `UpsertsActiveDataSize`, and `DeletesActiveDataSize` determine how many inserts, updates, upserts, and deletes, respectively, should be executed before releasing and then requiring the lock on the data object being modified. This keeps the operation from holding the data object lock for long periods of time when there are view sets open on this data object. Default: 50
`UpsertsActiveDataSize`	`InsertsActiveDataSize`, `UpdatesActiveDataSize`, `UpsertsActiveDataSize`, and `DeletesActiveDataSize` determine how many inserts, updates, upserts, and deletes, respectively, should be executed before releasing and then requiring the lock on the data object being modified. This keeps the operation from holding the data object lock for long periods of time when there are view sets open on this data object. Default: 50
`UseDBFailover`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: `true`
`ViewSetExpiryTimeout`	This property is for Oracle BAM internal use only. Do not change it unless instructed to do so by Oracle Support Services. Default: 120
`ViewSetSharing`	See Section 23.3.4, "Configuring Viewset Sharing" for more information. Default: `true`

24

Monitoring Oracle Business Activity Monitoring

This chapter describes how to use Oracle Enterprise Manager Fusion Middleware Control to view Oracle Business Activity Monitoring (Oracle BAM) performance statistics and logs, and monitor Oracle BAM web services counters, and Oracle BAM server and web components such as Active Data Cache, Event Engine, Report Cache, Enterprise Message Sources, and Report Server.

This chapter includes the following topics:

- Section 24.1, "Introduction to Monitoring Oracle BAM"
- Section 24.2, "Monitoring Oracle BAM Server Components"
- Section 24.3, "Monitoring Oracle BAM Web Applications"
- Section 24.4, "Monitoring Oracle BAM Web Services"
- Section 24.5, "Monitoring Oracle BAM Performance"
- Section 24.6, "Monitoring Oracle BAM Logs"

24.1 Introduction to Monitoring Oracle BAM

You can monitor several aspects of the Oracle BAM components using Oracle Enterprise Manager Fusion Middleware Control.

The OracleBamServer page in Fusion Middleware Control enables you to monitor each of the Oracle BAM Server Components: Active Data Cache, Event Engine, Report Cache, and Enterprise Message Sources. For more information, see Section 24.2, "Monitoring Oracle BAM Server Components".

The OracleBamWeb page in Fusion Middleware Control enables you to monitor Oracle BAM Report Server. For more information, see Section 24.3, "Monitoring Oracle BAM Web Applications".

In addition, the Performance Summary pages for each component allow you to track specific statistics of your choosing. For more information, see Section 24.5, "Monitoring Oracle BAM Performance".

Also, you can configure and monitor Oracle BAM logs using Fusion Middleware Control. For more information, see Section 24.6, "Monitoring Oracle BAM Logs".

24.2 Monitoring Oracle BAM Server Components

Oracle BAM Server components are monitored on the Fusion Middleware Control page for Oracle BAM Server.

To monitor Oracle BAM Server components:

Go to the Oracle BAM Server home page by selecting **BAM** > **OracleBamServer** in the navigation tree.

In the OracleBamServer page, you can select each tab (Active Data Cache, Event Engine, Report Cache, Enterprise Message Sources) to monitor the individual Oracle BAM Server components.

Note: Alternatively, on the farm home page, you can also select the Oracle BAM targets from the right side page content area, where it displays details of the farm.

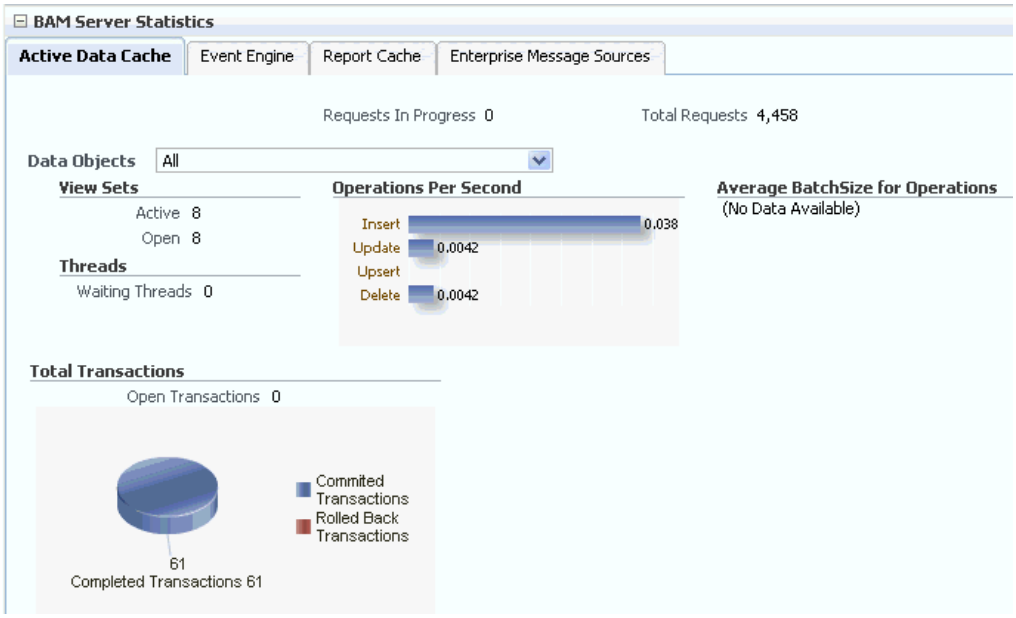
24.2.1 Monitoring Oracle BAM Active Data Cache

Oracle BAM Active Data Cache (Oracle BAM ADC) is designed and optimized to handle large amounts of data in real time. Data coming into Oracle BAM ADC immediately updates all defined calculations and aggregates in real time in an in-memory cache so that this data can be pushed as quickly as possible to Oracle BAM dashboards and reports. Data fed to the Oracle BAM ADC is received from a combination of sources, from Java Message Service (JMS) topics and queues to more traditional data queries and databases. Oracle BAM ADC ensures that no matter the source, when Oracle BAM Server is aware of the incoming data, it is streamed to Oracle BAM alerts and dashboards as incremental changes.

There are two key components to the Oracle BAM ADC, data sets and view sets. Oracle BAM ADC receives transactions (insert, update, upsert, delete) into the data sets. These data sets are constructed based on the design that an Oracle BAM developer defines for data relationships when data objects are defined in Oracle BAM Architect. Data objects can be reflective of flat tables of data, or more complex, star-schema relationships between data objects, which are represented in Oracle BAM as data object look-ups.

After data is updated in the data sets, view sets that are listening on these data sets go into action. The view sets have knowledge of all of the open Oracle BAM dashboards and alerts that users are viewing in the Oracle BAM system. View sets ensure that data updates to these open objects are incrementally updated first, using a push-based mechanism to publish updates to open dashboards and alerts, providing users with the latest information.

Use the Oracle BAM pages in Fusion Middleware Control to monitor statistics for all data objects in Oracle BAM ADC as a group, or select a particular data object from the **Data Objects** list to monitor.



Note: When you select a data object from the list, cached data is displayed to preserve performance. You must refresh the page (using the Refresh icon) to display the latest data associated with the selected data object.

View Sets

Active view sets are those which are typically seen in Oracle BAM reports. Open view sets are the sum of active and static open view sets.

Oracle BAM reports and alerts use active view sets. Oracle BAM Server uses static view sets for internal process. There are also internal parts of the code that use static view sets on system objects (metadata tables). It is possible that if Oracle BAM Server is not closing the static view sets, you observe different values for them.

Static view sets do not support active data generation and are used to fetch data from data objects, therefore, they are kept open for only a very short time when data is fetched. For that reason, in the **Open** counter, the values appearing most of the time are from active view sets.

View Set Count Variation

You can monitor the view set count for Oracle BAM components Active Data Cache, Report Cache, and Report Server. You may see that view set counts do not match between Active Data Cache, Report Cache, and Report Server metrics.

The Report Server opens view sets through the Report Cache for all of the views contained in the reports opened in browsers. The Active Data Cache opens more view sets than the Report Cache. One such view set opened is in the Active Data Cache through the Event Engine.

Whenever the Event Engine starts, it loads all the defined alerts, and for those alerts that are defined to monitor data changes in the Active Data Cache, corresponding view sets are opened in the Active Data Cache. That is why more view sets are

displayed in the Active Data Cache monitoring page than on the Report Cache and Report Server pages.

For most of the views in a report, there is a corresponding view set in the Active Data Cache. View sets opened by the report are shared with the subsequent instances of that report, except for Crosstab views. In reports using the Crosstab view, the view set call is different than the one used by other views. When a report containing a Crosstab view is opened, three view sets are opened in the Active Data Cache. Two of the view sets are closed immediately after fetching the data, while one is kept open to monitor the incoming changes.

Operations Per Second

This statistic includes all of the insert, update, upsert, and delete operations occurring on the selected data object. The accompanying graph displays statistics for the last 5 minutes.

Average Batch Size for Operations

This statistic is the average number of records included in each batch operation on the selected data object. The accompanying graph displays statistics for the last 5 minutes.

Waiting Threads

Certain operations must be synchronized, requiring a data object to be locked. This statistic displays the number of threads that are blocked, waiting for the data object to be unlocked to perform subsequent operations.

Total Transactions

This statistic displays the number of transactions on Oracle BAM data objects. It is the sum of committed and rolled back transactions. The accompanying graph also displays the number of open and completed transactions.

For example, if you add a row in an Oracle BAM data object and save it (commit), it is one transaction. This metric is useful only if the client of Oracle BAM Server is using transactions.

Transactions can also be used by an EMS, when connecting to a JMS topic or queue.

You can configure this metric to track transactions in the Oracle BAM sensor actions in BPEL and in the Oracle Data Integrator knowledge modules.

In regard to **Rolled Back Transactions**, if an invalid insert operation is attempted on an Oracle BAM data object, Oracle BAM Server rolls back the transaction.

24.2.2 Monitoring the Event Engine Component

The Event Engine is used by Oracle BAM Server to evaluate and execute Oracle BAM alerts. The Event Engine monitors complex data conditions and implements specified rules. Rules can include a series of conditions and actions attached to an event. The Event Engine continuously monitors the information in the Active Data Cache for certain conditions and executes the related actions defined in associated rules.

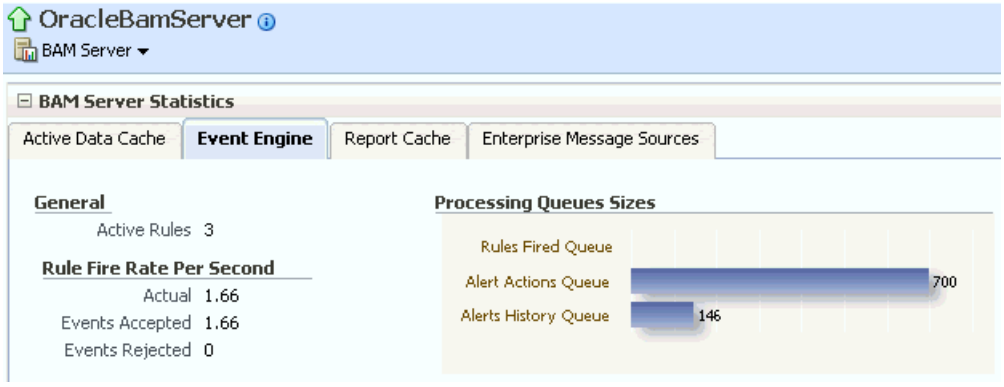
The Event Engine is responsible for tracking events based on date, time, or data changes. The Event Engine design employs a satellite concept, in which there are four different systems (satellites) within which event clauses can be registered and tracked.

The Date and Time satellites are both based on a scheduler, and they are used for time based alerts, for example, "Email a snapshot of this Oracle BAM dashboard to me every morning at 8 a.m."

The Manual satellite is use to *manually* send alerts by name (using a web service call).

The Data satellite is used to track alerts that are based on specific changes in the data of a given data object, such as "Call this web service when SUM(Sales) > 10000 in the MediaSales data object," or "Let me know when anything in this dashboard changes."

You can monitor statistics for active Event Engine rules, condition met rates, and processing queues.



Active Rules

This statistic displays the number of alert rules that are active (not expired or invalid) in Oracle BAM Event Engine. It displays the total number of alert rules in Oracle BAM Event Engine, which is the sum of alerts created for all of the Oracle BAM users.

Rule Fire Rate Per Second

This group of statistics displays the rates of events over the last 5 minutes for **Actual**, **Events Accepted**, and **Events Rejected** rates.

Actual rule fire rate per second is the number of events fired per second in Event Service. It is the number of events fired per second, taking into account number of times alerts fired events in the previous 5 minutes. For example, if when the OracleBamServer page was loaded, only 2 alerts fired in previous 5 minutes, the value shown would be 2/(5 X 60), that is 0.0067. So, if fired twice in 300 seconds, it would fire 0.0067 times in one second.

Events Accepted is the number of events fired per second from the satellite. That it, the number of events fired per second from alerts configured, that were accepted (for further processing), taking into account events that were accepted in the previous 5 minutes. The value shown is calculated in the same way as **Actual**. Unless the Event Engine's queue thresholds are exceeded, all events fired are accepted and processed further.

Events Rejected is the number of rejecting events per second. That is, the number of events fired per second from alerts configured, that were rejected (for further processing), taking into account events that were rejected in the previous 5 minutes. The value shown is calculated the same way as **Actual**. Fired events are rejected when the Event Engine's queue thresholds are exceeded. Ideally this value should be zero.

Processing Queues Sizes

This graph displays the size of the event processing queue, that is, the number of events waiting to be processed. These queues are emptied as soon as the events are processed, so there would have to be a large number of events being generated simultaneously to see alerts waiting for processing in the queues.

The **Processing Queues Sizes** graph displays statistics for the last 5 minutes. The graph displays **Rules Fired Queue**, **Alert Actions Queue**, and **Alerts History Queue** health.

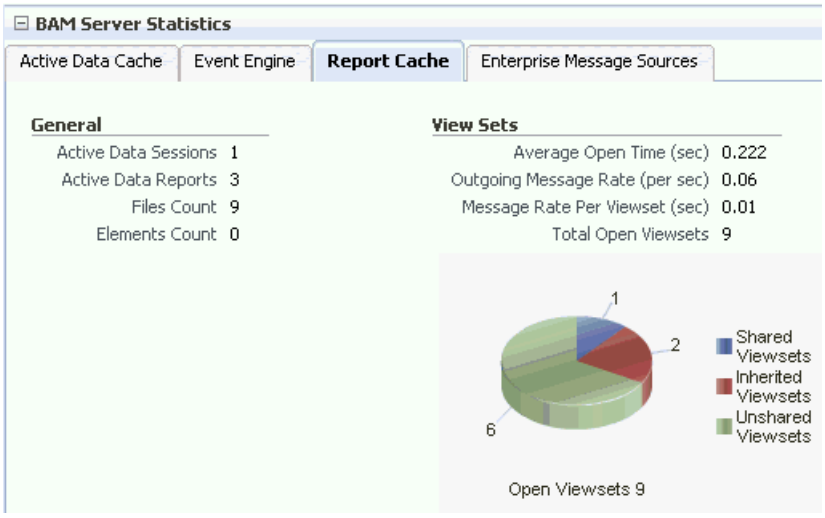
The **Rules Fired Queue** displays the number of events that were fired that are waiting to be processed. The Event Engine maintains internal in-memory queue for fired events where all the events are placed temporarily, before being pulled out from here for further processing, as soon as the Event Engine resources are available. This metric shows the number of events fired waiting in queue to be picked by the system.

The **Alert Actions Queue** displays the number of actions waiting to be processed. The Event Engine maintains internal in-memory queue for actions where all the actions are placed temporarily, before being pulled out from here for further processing, as soon as the Event Engine resources are available. This metric shows the number of actions waiting in queue to be picked by the system.

The **Alerts History Queue** displays the number of alert history items to be saved. The Event Engine maintains internal in-memory queue for all history items, where all the items are temporarily placed, and before picking them up to save to back end. This metric shows the number of history items waiting in queue to be picked by the system.

24.2.3 Monitoring the Report Cache Component

The Report Cache assists the Active Data Cache with maintenance of the view set snapshots in memory. The Report Cache opens view sets in the Active Data Cache for the Report Server (an Oracle BAM web applications component). It then caches the snapshot (in small parts) and the active data before sending it to the Report Server. This allows for random access into the snapshot and recovery from losing the connection to Oracle BAM Server. The Report Cache also, along with the Active Data Cache, supports view set sharing.



Active Data Sessions

This statistic displays the total number of Report Cache sessions with active data.

When a user opens multiple browser windows on a single computer to view Oracle BAM reports, all of the open browser windows share the same active data session. It does not matter whether the user opens the same report or different reports in each browser window, the **Active Data Sessions** count is always 1.

Active Data Reports

This statistic displays the total number of open reports requesting active data.

Files Count

This statistic displays the total number of files currently managed by `FileBasedPersistenceManager` when the Oracle BAM Server is configured to use a file-based persistence manager. For more information, see Section 23.3.5, "Configuring the Report Cache Persistence Manager".

Elements Count

This statistic displays the total number of elements currently managed by `InMemoryPersistenceManager` when the Oracle BAM Server is configured to use a memory-based persistence manager. For more information, see Section 23.3.5, "Configuring the Report Cache Persistence Manager".

View Sets

This group of statistics displays the following counters:

Average Open Time (sec) is the average time taken to open a view set.

Outgoing Message Rate (per sec) is the number of change lists delivered per second by Report Cache to all of its clients.

Message Rate Per Viewset (sec) the number of change lists delivered per second per view set.

Total Open Viewsets is the total number of view sets (that is, the sum of shared, unshared, and inherited view sets) maintained in Report Cache for all of the views contained in the reports opened in browsers.

The **Open Viewsets** graph displays shared, inherited, and unshared view sets.

Shared Viewsets are the parent view sets (these are the view sets that are opened first).

Inherited Viewsets are the child view sets (these view sets are opened later).

Unshared Viewsets (independent view sets) are neither shared or inherited.

View set sharing occurs whenever possible for performance reasons (the consumers require the exact same view sets, so the view sets can be shared). If the view sets cannot be shared, they are unshared.

When users who do not have any differences in row-level security open the same report at close to the same time, the view sets that are created for them in the Active Data Cache are shared in the Report Cache.

Note: View set counts may not match between Active Data Cache, Report Cache, and Report Server metrics. For more information, see "View Set Count Variation" on page 24-3.

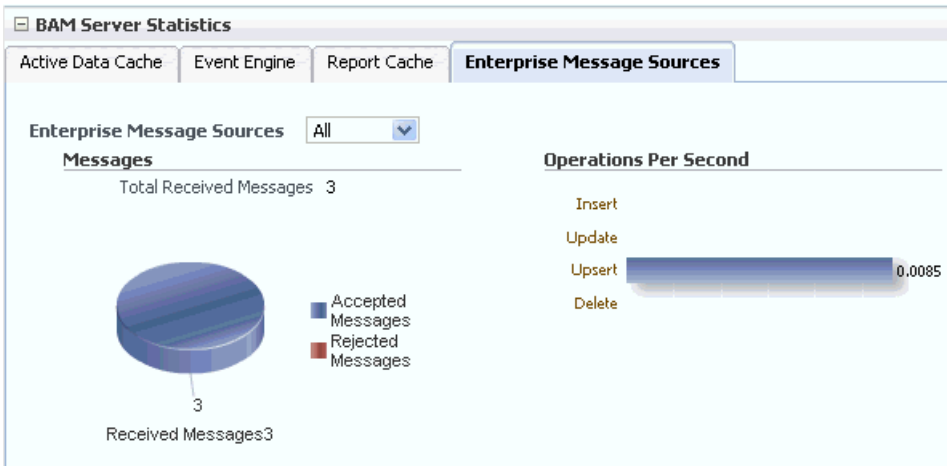
24.2.4 Monitoring the Enterprise Message Sources

Enterprise Message Sources (EMS) are used by applications to provide direct Java Message Service (JMS) connectivity to Oracle BAM Server by mapping messages directly to Oracle BAM data objects. Oracle BAM Server can read data directly from any JMS-based message queue or topic. This option offers guaranteed messaging. It is more difficult to configure and not as fast to perform rigorous data transformations in

XML Stylesheet Language (XSL) than in an Extract Transform and Load (ETL) tool like Oracle Data Integrator.

The EMS feature does not configure ETL scenarios, but rather maps from a message directly to a data object on Oracle BAM Server; however, you can still use XSL transformations before the data is inserted (updated, upserted, or deleted) into the data object. Each EMS reads from a specific JMS topic or queue, and the information is delivered into a data object in the Active Data Cache. The Oracle BAM Architect web application is used to configure EMS definitions.

For more information about configuring EMS definitions, see the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.



Note: Data is displayed in the Enterprise Message Source page only when EMS definitions are created and started in Oracle BAM Architect. Fusion Middleware Control may take some time to fetch the data and display the statistic in the Enterprise Message Source page.

Refreshing the page displays the latest EMS data. You can refresh the page using the Refresh icon.

Enterprise Message Sources

Select the EMS to monitor from the list. You can choose to display aggregated statistics for all EMS definitions, or select a particular EMS to monitor from the list provided.

Messages

Monitor the number of messages received by the selected EMS. You can find rejection statistics in the Performance Summary metrics page.

The accompanying graph displays the number of **Received Messages** broken down into **Accepted Messages** and **Rejected Messages**, and at the top of the graph the number of **Total Received Messages** is displayed.

Operations Per Second

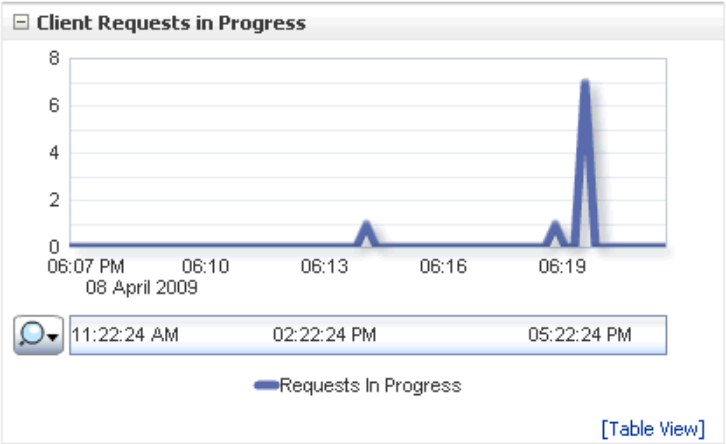
Monitor the rate of insert, update, upsert, and delete operations performed by the selected EMS.

The Operations Per Second graph displays statistics for last 5 minutes.

For more granular detail about the rates for each type of operation, go to the Performance Summary page. For more information, see Section 24.5, "Monitoring Oracle BAM Performance".

24.2.5 Monitoring the Client Requests in Progress

The Oracle BAM **Client Requests in Progress** chart is displayed in the lower right corner of the Oracle BAM Server home page. Client requests include all of the requests made to the Oracle BAM Server Enterprise Java Beans (EJB).



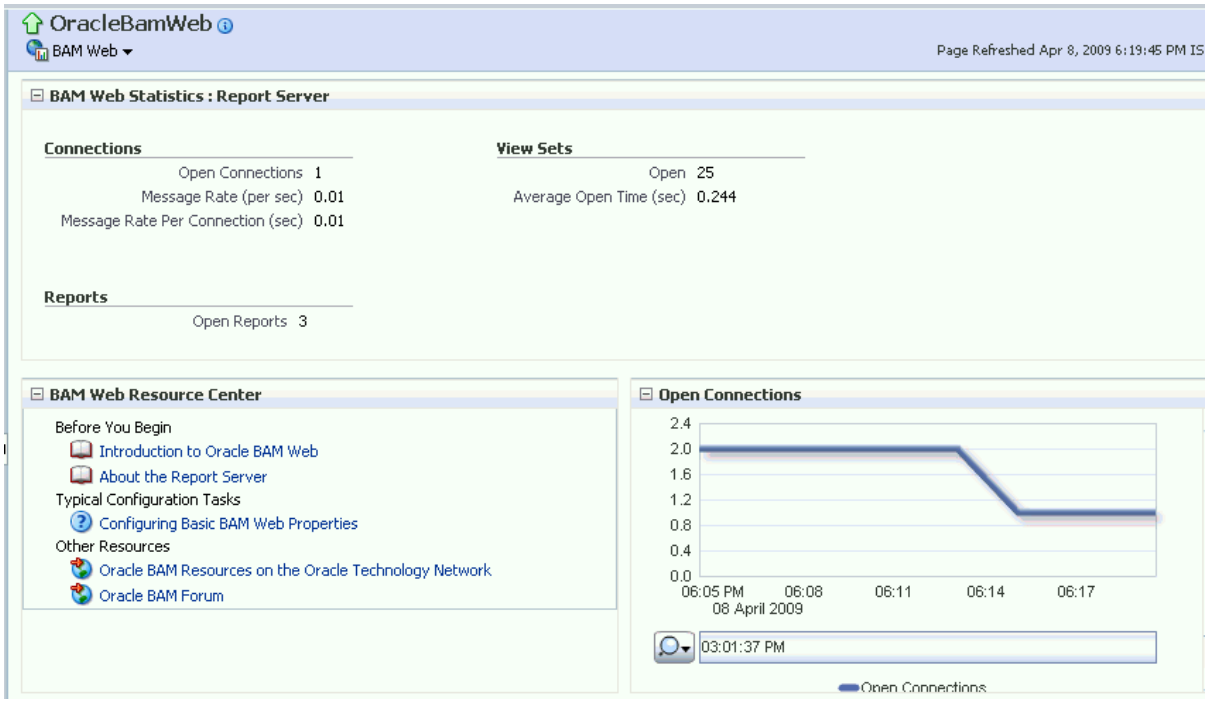
There are a lot of internal requests made when opening a single report (for permissions, the report metadata, the views data, and so on); however, the user may or may not see those internal requests displayed in this chart.

24.3 Monitoring Oracle BAM Web Applications

Oracle BAM web applications are monitored in the OracleBAMWeb page of Fusion Middleware Control.

To monitor Oracle BAM Report Server go to the OracleBAMWeb page by selecting **BAM** > **OracleBamWeb** in the navigation tree.

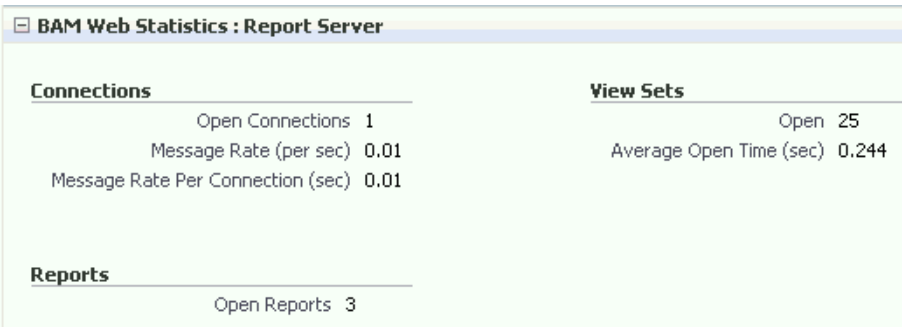
Figure 24–1 BAM Web Statistics



24.3.1 Monitoring Oracle BAM Report Server

Oracle BAM Report Server applies the report definitions to the data sets retrieved from the Oracle BAM ADC for presentation in a browser. It manages information paging for viewing and printing reports. After reports are created, they are stored in the Oracle BAM ADC so that report creation is not repeated each time. Most reporting views are designed to support live, active displays of data changing in real time.

You can monitor statistics for Oracle BAM Report Server connections, view sets, and reports.



Connections

This group of statistics displays the following:

Open Connections is the number of connections open on the Report Server. An open connection corresponds to each user per session. It is different from the number of reports opened, because one user could at most have one connection open at a time regardless of how many reports the user has opened.

Message Rate (per sec) is the total number of messages (view set change lists) delivered per second to all users (connections).

Message Rate Per Connection (sec) is the number of messages (view set change lists) delivered per second per user (connection).

View Sets

This group of statistics displays the number of open view sets, and the average amount of time (in seconds) that it takes to open the view sets on the server.

Average Open Time (sec) is a measure of the average length of time the server requires to construct the view set and send it to the web browser. This statistic does not consider the amount of time required to render the HTML and load the Javascript on the client-side web browser.

Note: View set counts may not match between Active Data Cache, Report Cache, and Report Server metrics. For more information, see "View Set Count Variation" on page 24-3.

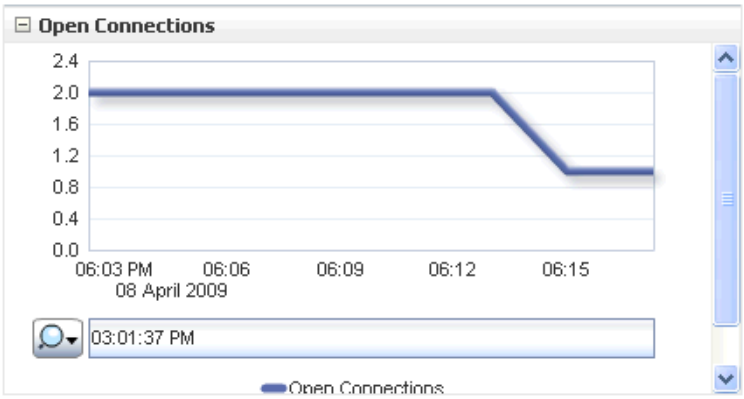
Reports

This statistic displays the total number of open reports. The **Open Connections** graph displays open connections to Oracle BAM web applications. You can use the zoom axis to zoom in on a particular time in which you are interested to see the open connections statistic.

The total number of reports includes the count of duplicate reports opened by the same user. Even if the same report is opened in multiple browser windows, each instance is considered a separate report, because a new view set is opened in Report Server (through Oracle BAM Report Cache) for each report instance, though the view sets are shared.

24.3.2 Monitoring Open Connections

The Oracle BAM **Open Connections** chart is displayed in the lower right corner of the Oracle BAM Server home page.



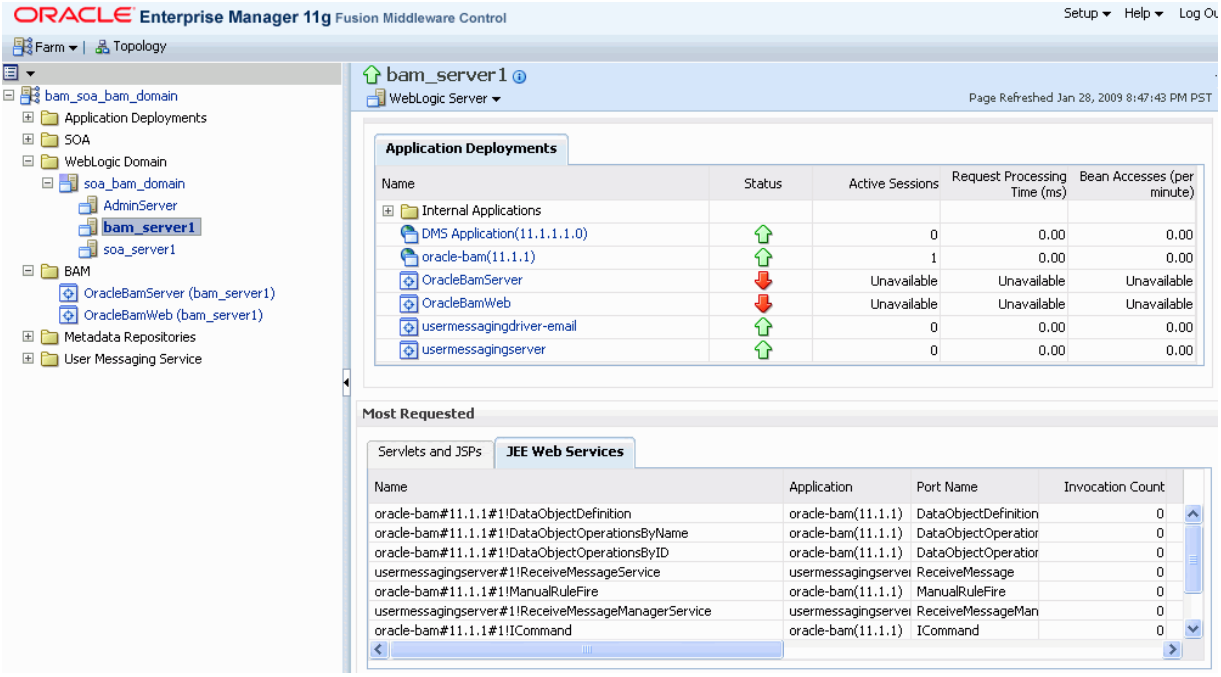
24.4 Monitoring Oracle BAM Web Services

Oracle BAM web services details are not listed in the Oracle BAM pages in Fusion Middleware Control. Oracle BAM web services counters are available by selecting **WebLogic Domain** > **soa_bam_domain** > ***BAM_Server*** in the navigation tree, and selecting the **JEE Web Services** tab in the **Most Requested** region as shown in Figure 24–2.

Oracle BAM provides DataObjectDefinition, DataObjectOperationsByName, DataObjectOperationsByID, DataObjectOperations10131, ManualRuleFire, and ICommand web services. For more information, see "Using Oracle BAM Web Services" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

You can see the following statistics in JEE Web Services table: **Invocation Count**, **Response Count**, **Response Error Count**, **Average Response Time (ms)**, **Average Execution Time (ms)**, and **Dispatch Time Total**.

Figure 24–2 Oracle BAM Web Services Counters



24.5 Monitoring Oracle BAM Performance

The performance of Oracle BAM applications is reflected in metrics and statistics. There are separate Performance Summary pages with appropriate metrics for Oracle BAM Server and Oracle BAM web applications.

When you select the **Monitoring** > **Performance Summary** menu item in each of the Oracle BAM component menus (as shown in Figure 24–3), the Performance Summary page appears as shown in Figure 24–4.

Figure 24–3 Monitoring Shortcut Menu

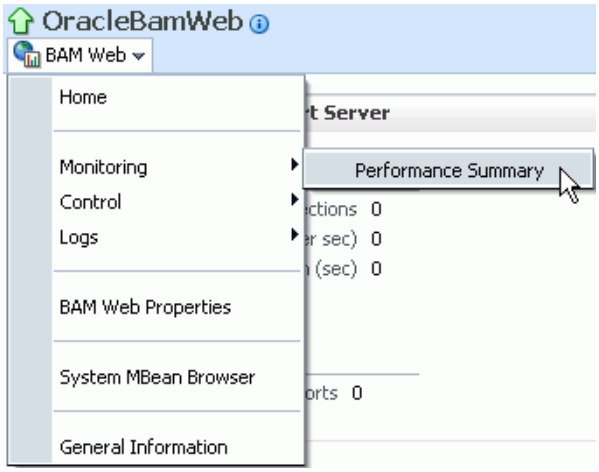
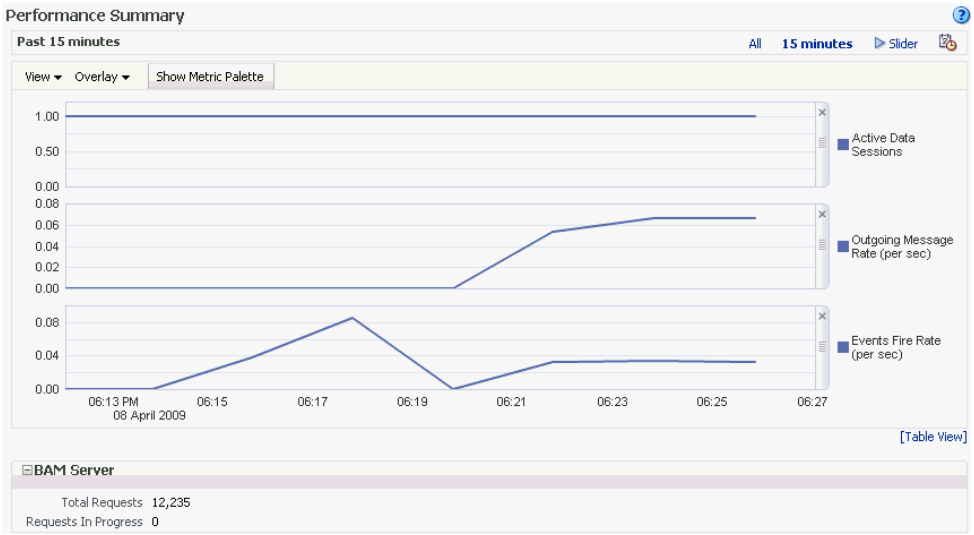


Figure 24–4 Performance Summary Page

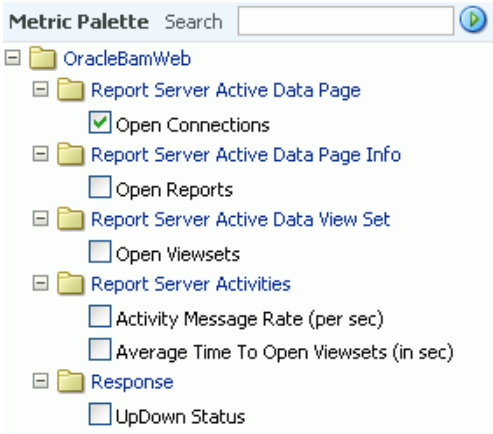


Many metrics are available for capture and display (most of which are also displayed in the Oracle BAM component home pages). Descriptions of each metric are available in the metrics help pages.

To get the most valuable, focused information, use the Metric Palette. Click **Show Metric Palette** to display the Metric Palette, shown in Figure 24–5.

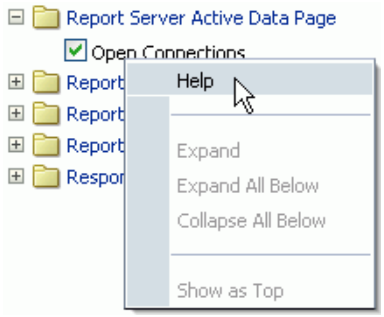
Choose the metrics in which you are most interested. As you select or deselect metrics from the palette, the metrics graph at the left is updated automatically.

Figure 24–5 Metric Palette for Oracle BAM Web Applications



Right-click the metric label and select **Help**, as shown in Figure 24–6, to find more information about each of the metrics.

Figure 24–6 Metrics Shortcut Menu



24.6 Monitoring Oracle BAM Logs

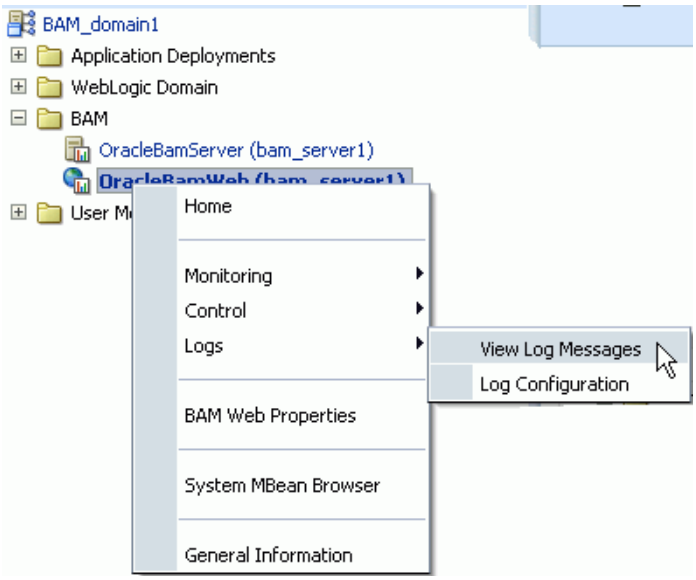
Oracle BAM logs are configured and viewed using Fusion Middleware Control.

Note: Despite having separate log pages for Oracle BAM Server and Oracle BAM web applications, each page displays the logs for all Oracle BAM application components.

For information about using the logger features see the online Help page. For information about configuring the logger, see Section 23.4, "Configuring the Logger."

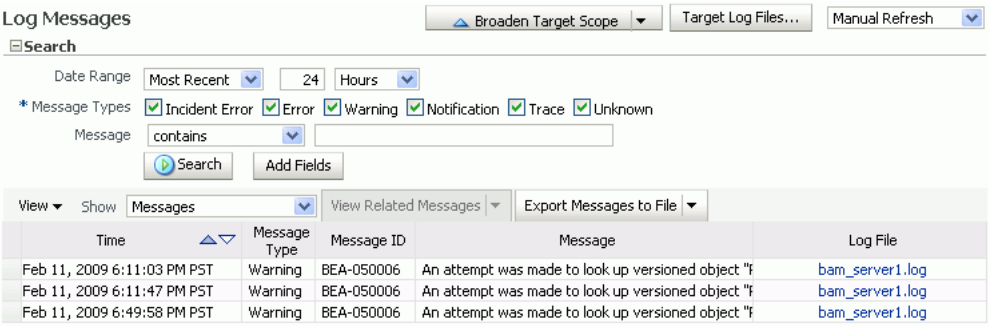
To view the logs, right-click the **OracleBamServer** node or **OracleBamWeb** node in the navigation tree and select **Logs** > **View Log Messages** as shown in Figure 24–7.

Figure 24–7 Logs Shortcut Menu



The log viewer page opens as shown in Figure 24–8. Use this page to query for information about the component. Fields and lists are used to customize the query.

Figure 24–8 Log Messages Page



After entering your search criteria, click **Target Log Files**. The Log Files page appears as shown in Figure 24–9. You can view specific log information (shown in Figure 24–10) or download the log.

Figure 24–9 Log Files Page

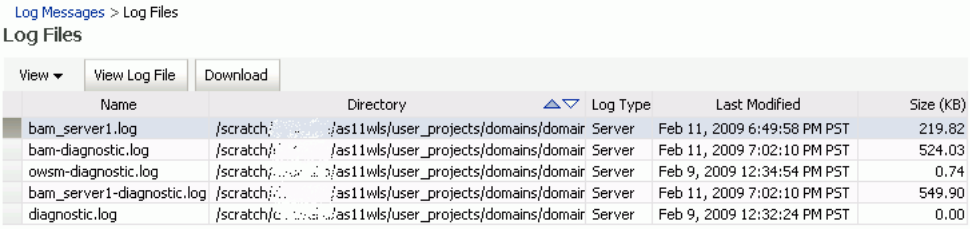
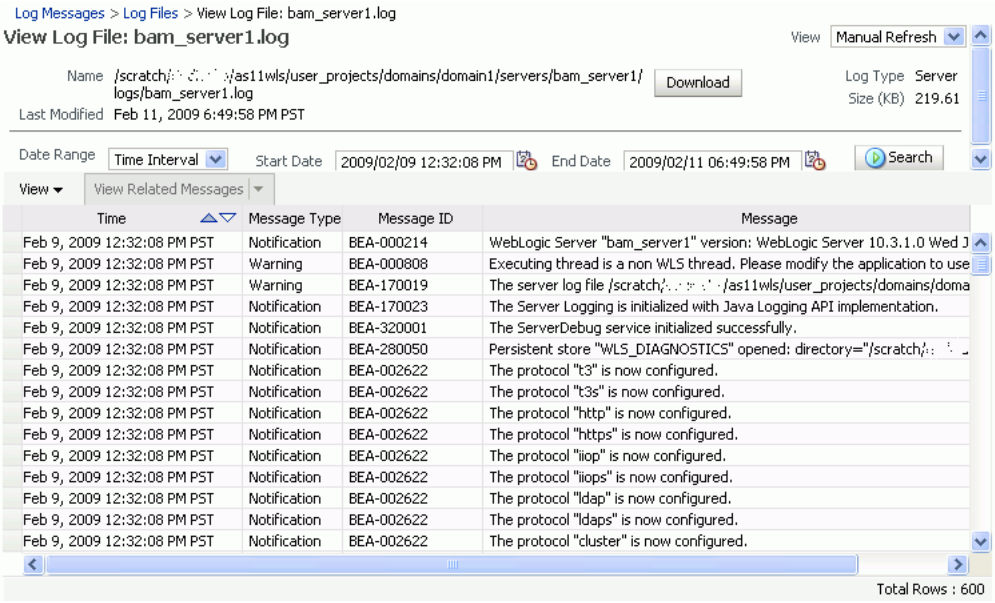


Figure 24–10 Log File bam_server1.log



25

Managing Oracle Business Activity Monitoring

This chapter describes how to start and stop Oracle Business Activity Monitoring (Oracle BAM) components, create users and groups, grant Oracle BAM application permissions using application-level roles, and set up data access permissions on data objects and folders, and reports.

This chapter includes the following topics:

- Section 25.1, "Introduction to Managing Oracle BAM"
- Section 25.2, "Managing Oracle BAM Availability"
- Section 25.3, "Managing Oracle BAM Users"

25.1 Introduction to Managing Oracle BAM

Use Oracle Enterprise Manager Fusion Middleware Control to manage Oracle BAM availability. For more information, see Section 25.2, "Managing Oracle BAM Availability".

Oracle BAM users and groups are created in Oracle WebLogic Server Administration Console (or in a security provider configured for your Oracle WebLogic Server); application-level roles are administered and new groups/roles and policies/grants can be created in Fusion Middleware Control (in **Application Policies**); and the users' Oracle BAM objects are managed in Oracle BAM Administrator. For more information, see Section 25.3, "Managing Oracle BAM Users".

25.2 Managing Oracle BAM Availability

Oracle BAM Server and web applications can be started and stopped using Fusion Middleware Control, or, as a convenience, the entire Oracle BAM application (or the Oracle WebLogic Server to which it is deployed) can be restarted in the Oracle WebLogic Server Administration Console.

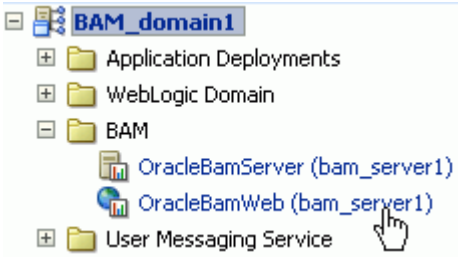
Note: On all UNIX installations of Oracle BAM, due to a behavior of the Java AWT APIs that Oracle BAM uses, Oracle BAM Server must be started with the `DISPLAY` environment variable **un**set.

```
$ unsetenv DISPLAY
```

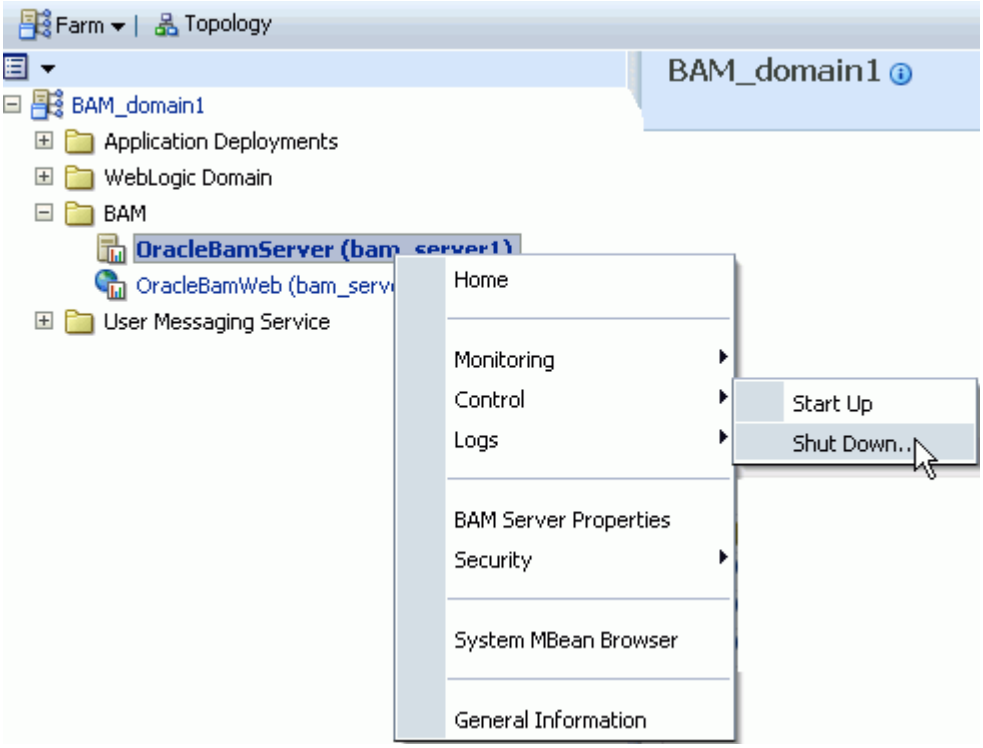
Restarting is required for any configuration changes made in the Fusion Middleware Control properties pages for Oracle BAM or its configuration files.

To start or stop Oracle BAM components in Fusion Middleware Control:

1. In Fusion Middleware Control locate the **OracleBamServer** or **OracleBamWeb** node in the navigation tree.



2. Right-click the node to open the shortcut menu, and select **Control**.



Start Up starts the component.

Shut Down shuts down the component.

Note: Do not use **Start Up** alone to restart the component. To restart the component, you must stop the component using **Shut Down** before using **Start Up** to start the component or an error occurs.

25.3 Managing Oracle BAM Users

These are the main steps for managing Oracle BAM users:

1. Configure the Security Provider

Users, groups, or both, to be used with Oracle BAM users must be present in a security provider configured for your Oracle WebLogic Server. For the main steps and links to detailed information about configuring security for Oracle WebLogic Server, see the *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

> **Note:** You can use Oracle WebLogic Server to configure the Active Directory Authentication provider for authenticating Oracle BAM users instead of using the default embedded LDAP (also known as the default authenticator). To use Active Directory, you must also do some additional steps to change the OracleSystemUser default user. For more information, see "Changing the OracleSystemUser Default User" in the *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

2. Create Users and Groups

Users and groups are defined in the configured security provider (for example, in the Oracle WebLogic Server embedded LDAP server). Refer to your specific security provider documentation for details on defining users and groups. For more information, see Section 25.3.1, "Defining Users and Groups" and Section 25.3.2, "Using Previously Seeded Group Members".

For example instructions about using Oracle WebLogic Server Administration Console to create users and groups, see Section 25.3.5, "Configuring Oracle WebLogic Server Embedded LDAP Server".

3. Assign Users and Groups to Application Roles

In turn, these users, groups, or both, are assigned to Oracle BAM application-level roles that grant those users, groups, or both, specific permissions for using Oracle BAM applications. Users and groups are granted Oracle BAM application permissions based on their Oracle BAM role membership.

For a detailed description of the Oracle BAM application roles and their associated Oracle BAM application permissions, see Section 25.3.3, "Adding Members to Application Roles".

Membership in Oracle BAM application roles is administered from the Application Roles page for Oracle BAM provided by Fusion Middleware Control. This page allows users and groups to be added as members to the various Oracle BAM application roles and allows creation of new application roles. For more information, see Section 25.3.4, "Introduction to Oracle BAM Application Roles".

> **Note:** Oracle BAM does not support assigning Oracle BAM application permissions directly to users and groups. Oracle BAM application permissions can only be granted to Oracle BAM application roles.
>
> The only way to grant Oracle BAM application permissions to users and groups is to make those users and groups members of an Oracle BAM application role associated with the desired Oracle BAM application permissions.

With the exception of the Administrator role, membership in an Oracle BAM application role does not imply any Oracle BAM data access permissions. The Oracle

BAM application roles only grant the user access to the associated Oracle BAM user interface as described in Section 25.3.4, "Introduction to Oracle BAM Application Roles."

When the user logs on to the Oracle BAM start page, there is a button for each of the Oracle BAM applications. Whether these buttons are enabled or not is based on the user's Oracle BAM application role membership.

Note: Changes to a user's group and role membership could take as long as 5 minutes to propagate throughout the system.

---

4. Populate Users In Oracle BAM Applications

Users are not visible from Oracle BAM Administrator until they have logged into Oracle BAM for the first time. Oracle BAM also provides a utility that you can run to populate the users in Oracle BAM Administrator. For more information, see Section 25.3.6, "Populating Users in Oracle BAM Administrator".

5. Set Up Data Access Permissions on Oracle BAM

Specific data access permissions can be granted to users and groups using Oracle BAM Architect and Oracle BAM Active Studio. Users and groups can be granted read, update, and delete operation permissions on specific data objects and folders. For more information, see "Creating Permissions on Data Objects" and "Using Data Object Folders" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

Data access permissions can also be granted to users and groups at the row level for data objects. For information about row-level data security, see "Creating Security Filters" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

Individual report authors can control which Oracle BAM users have access to reports. For more information, see "Setting Folder Permissions" in the *Oracle Fusion Middleware User's Guide for Oracle Business Activity Monitoring*.

6. Manage Oracle BAM Object Ownership

When Oracle BAM users are removed from the security provider, the user accounts still appear in Oracle BAM Administrator because they may *own* Oracle BAM objects that must be transferred to other users before the user is completely removed from Oracle BAM. Object ownership is managed using Oracle BAM Administrator (see Section 25.3.7, "Managing Oracle BAM Object Ownership").

7. Remove Users From Oracle BAM

The administrator must also remove users from Oracle BAM Administrator after they are deactivated in the security provider (see Section 25.3.8, "Removing Invalid Users from Oracle BAM Administrator").

OracleSystemUser

OracleSystemUser is the default owner of all Oracle BAM objects. It is required by Oracle BAM Server and must not be deleted.

25.3.1 Defining Users and Groups

Users are defined in the configured security provider's identity store (for example, Oracle WebLogic Server embedded LDAP server).

Groups, also referred to as enterprise-level roles, are also defined in this identity store. Groups are referred to as enterprise-level roles to distinguish them from application-level roles.

The enterprise-level roles are global to Oracle WebLogic Server, and they are applicable to all applications running on that server, including Oracle BAM. The application-level roles are specific to each application.

For example instructions about using Oracle WebLogic Server Administration Console to create users and groups, see Section 25.3.5, "Configuring Oracle WebLogic Server Embedded LDAP Server".

Note: When you delete a user you cannot reuse that user name until you resolve the old instance with Oracle BAM. Oracle BAM marks the deleted user inactive when you delete the user in Oracle WebLogic Server.

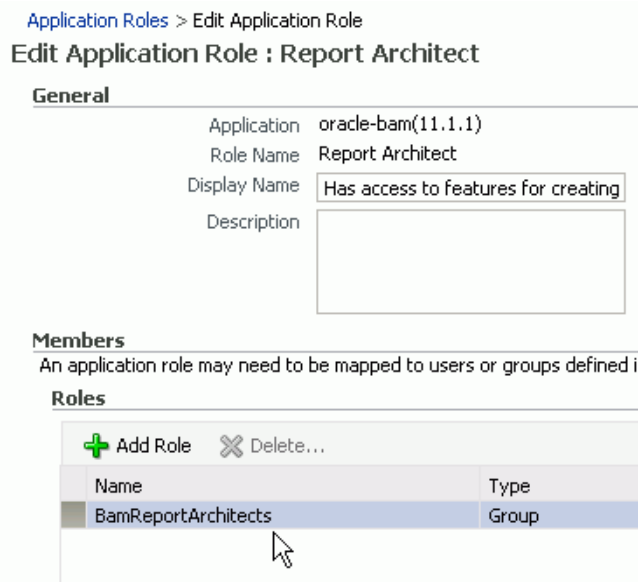
If you attempt to reuse the same user name for a new account, Oracle BAM does not consider the newly added user to be the same user as the one previously deleted. You must delete the original user from Oracle BAM using Oracle BAM Administrator (and optionally transfer the user's Oracle BAM objects to another valid user). Once the inactive user is removed from Oracle BAM you can reuse the user name.

For information about removing users from Oracle BAM, see Section 25.3.8, "Removing Invalid Users from Oracle BAM Administrator".

25.3.2 Using Previously Seeded Group Members

The following Oracle WebLogic Server groups have been previously seeded in the Oracle BAM application policy:

- BamAdministrators: Member of application role Administrator.
- BamReportArchitects: Member of application role Report Architect.
- BamReportCreators: Member of application role Report Creator.
- BamReportViewers: Member of application role Report Viewer.

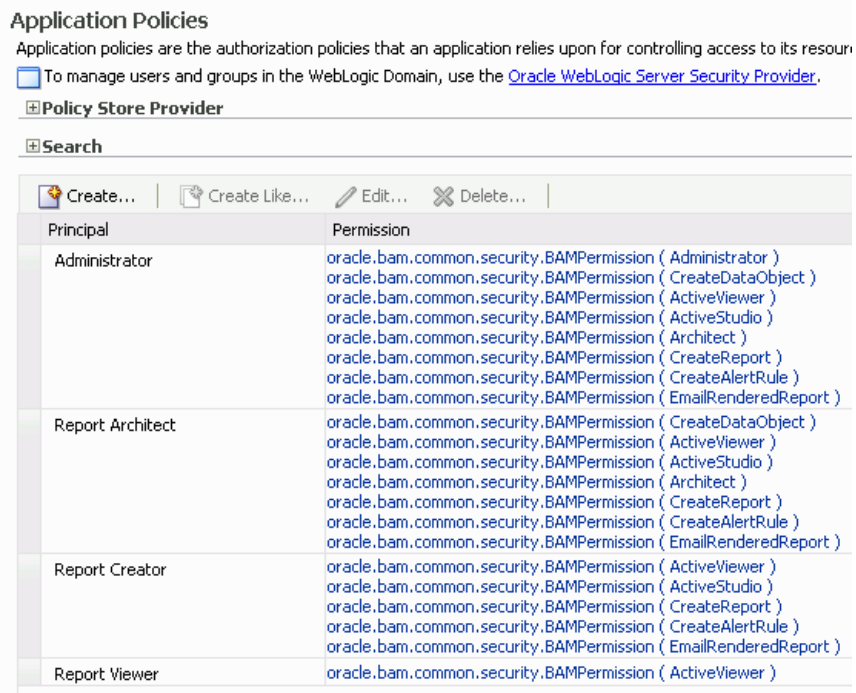


These members are a convenience. If you define these groups in your configured security provider, you can then assign Oracle BAM application-level roles to specific users and groups by placing them into these groups. All of this can be done from your security provider and does not require any Oracle BAM application policy modifications.

You must create these groups manually in the security provider because Oracle BAM does not automatically seed users or groups in the configured security provider.

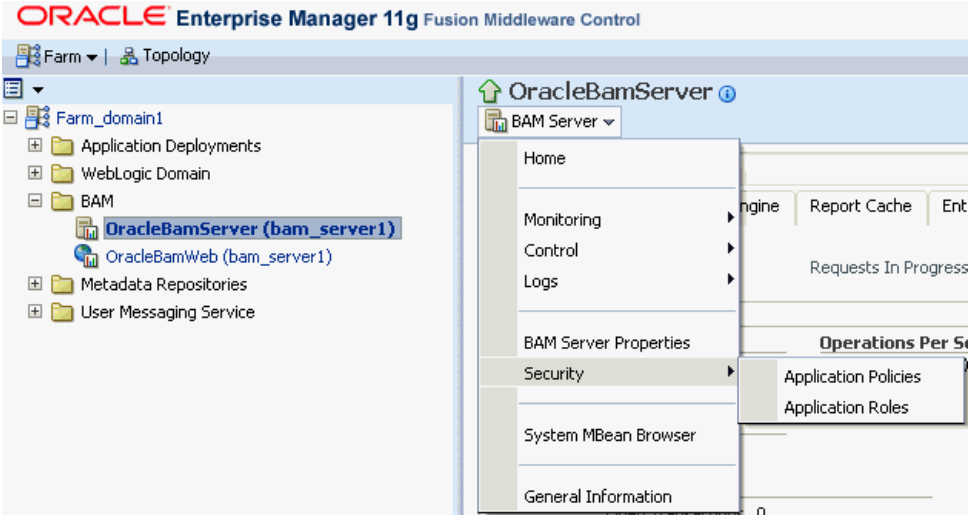
25.3.3 Adding Members to Application Roles

The Oracle BAM application policy defines the Oracle BAM application-level roles described in Section 25.3.4, "Introduction to Oracle BAM Application Roles" including role membership. The Oracle BAM application policy is managed in Fusion Middleware Control. The default policy store provider is the XML file-based policy store.

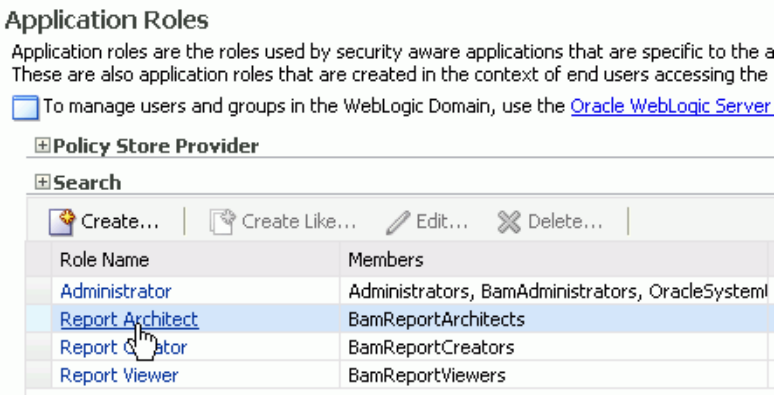


To add members to the Oracle BAM application-level roles, you must add entries to the membership list of the desired role using Fusion Middleware Control.

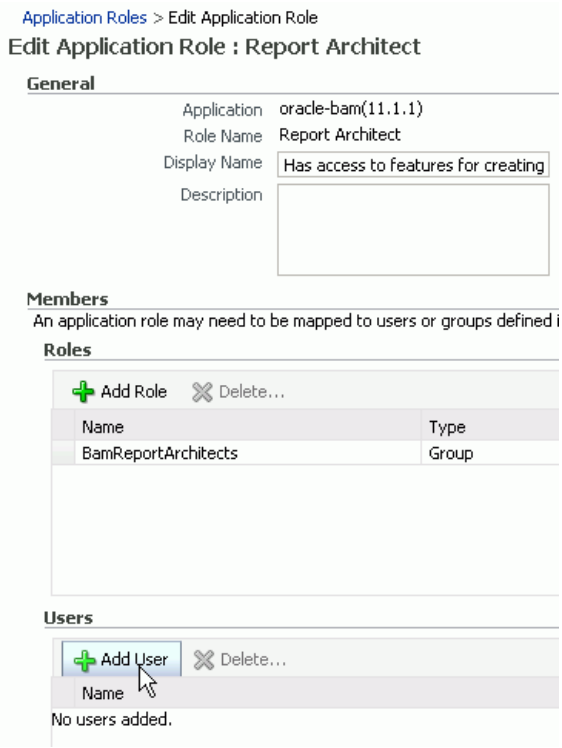
Shown here is the navigation required to open the Oracle BAM Application Roles page in Fusion Middleware Control:



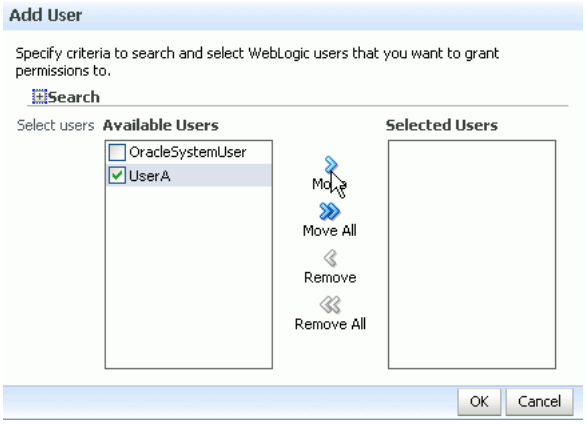
Select a role in the **Role Name** list:



Add a member to the role:



Select an available user and move it to the **Selected Users** list:



25.3.4 Introduction to Oracle BAM Application Roles

Oracle BAM defines the following application-level roles:

- Administrator: Has access to all features.
- Report Architect: Has access to features for creating data objects and reports.
- Report Creator: Has access to features for creating reports.
- Report Viewer: Has access to features for viewing reports.

The application roles determine the permissions granted to specific users or groups. If a user or group is a member of one of these Oracle BAM application roles, then they are granted the associated Oracle BAM permissions.

The Oracle BAM application roles are granted the following permissions:

Administrator Permissions

- Administrator: Has full access to Oracle BAM Administrator application functionality.
- CreateDataObject: Can create data objects in Oracle BAM Architect.
- ActiveViewer: Has full access to Oracle BAM Active Viewer application functionality.
- ActiveStudio: Has full access to Oracle BAM Active Studio application functionality.
- Architect: Has full access to Oracle BAM Architect application functionality.
- CreateReport: Can create reports in Oracle BAM Active Studio.
- CreateAlertRule: Can create alerts in Oracle BAM Architect and Oracle BAM Active Studio.
- EmailRenderedReport: Can email report attachments to Oracle BAM users.

Report Architect Permissions

- CreateDataObject: Can create data objects in Oracle BAM Architect.
- ActiveViewer: Has full access to Oracle BAM Active Viewer application functionality.
- ActiveStudio: Has full access to Oracle BAM Active Studio application functionality.
- Architect: Has full access to Oracle BAM Architect application functionality.
- CreateReport: Can create reports in Oracle BAM Active Studio.
- CreateAlertRule: Can create alerts in Oracle BAM Architect and Oracle BAM Active Studio.
- EmailRenderedReport: Can email report attachments to Oracle BAM users.

Report Creator Permissions

- ActiveViewer: Has full access to Oracle BAM Active Viewer application functionality.
- ActiveStudio: Has full access to Oracle BAM Active Studio application functionality.
- CreateReport: Can create reports in Oracle BAM Active Studio.

- CreateAlertRule: Can create alerts in Oracle BAM Active Studio.
- EmailRenderedReport: Can email report attachments to Oracle BAM users.

Report Viewer Permissions

- ActiveViewer: Has full access to Oracle BAM Active Studio application functionality.

25.3.5 Configuring Oracle WebLogic Server Embedded LDAP Server

The Oracle WebLogic Server embedded LDAP server is the default security provider for Oracle WebLogic Server. This section describes the procedures for adding new users and groups to the Oracle WebLogic Server embedded LDAP server.

For information about configuring the embedded LDAP server, see "Managing the Embedded LDAP Server" in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

25.3.5.1 Using the Oracle WebLogic Server Administration Console

Oracle WebLogic Server administration is performed using the Oracle WebLogic Server Administration Console at

`http://host_name:port_number/console`

Oracle WebLogic Server must be running to access the administration console. This console requires the user to log in with Oracle WebLogic Server administrator credentials. After successfully logging in, the user is placed at the Oracle WebLogic Server Administration Console home page.

25.3.5.2 Adding a Group

To add a group:

1. Go to **Home** > **Security Realms** > ***myrealm*** > **Users and Groups** in Oracle WebLogic Server Administration Console.
2. Select the **Groups** tab, and click **New**.
3. Enter the group name and description. Do not change the provider.
4. Click **OK**.

25.3.5.3 Adding a User

To add a user:

1. Go to **Home** > **Security Realms** > ***myrealm*** > **Users and Groups** in Oracle WebLogic Server Administration Console.
2. Select the **Users** tab, and click **New**.
3. Enter the user name, description, and login password. Do not change the provider.
4. Click **OK**.

25.3.5.4 Adding a User to a Group

To add a user to a group:

1. Go to **Home** > **Security Realms** > ***myrealm*** > **Users and Groups** in Oracle WebLogic Server Administration Console.
2. Select the **Users** tab, and select the user.

3. Select the **Groups** tab.
4. Move the groups from the **Available** list to the **Chosen** list. The **Chosen** list represents the list of groups of which the user is now a member.
5. Click **Save**.

25.3.6 Populating Users in Oracle BAM Administrator

Users are not automatically populated in Oracle BAM Administrator by the security provider. The system administrator must either run the `registerusers` utility, or have users log in to the Oracle BAM start page by using the provided credentials, before they can be managed in Oracle BAM Administrator.

Oracle BAM is a monitoring and analytics application that in implementation can support requirements to secure data, not only at the user and permissions level, but also at the row level for specific analytic data. Because of this requirement, Oracle BAM users require some management in Oracle BAM Administrator that is not provided in the standard Oracle WebLogic Server user management tool set. In order to configure these additional security features for a specific user, that user must be defined in Oracle BAM Administrator.

See the following topics for more information:

- Section 25.3.6.1, "Using the Registerusers Utility"
- Section 25.3.6.2, "Populating By User Login"

25.3.6.1 Using the Registerusers Utility

The `registerusers` utility is a standalone Java application for registering users with Oracle BAM Server. Normally, Oracle BAM is not aware of a particular defined user until that user logs into Oracle BAM for the first time. On initial login, Oracle BAM looks up the user in the configured security provider, and synchronizes Oracle BAM's definition for that user with that of the configured security provider. This includes the user name, role membership, group membership, user properties (for example, email address), and so on.

Because Oracle BAM is not aware of defined users until they have logged in at least one time, it is not possible to perform certain Oracle BAM user configuration related to that user. For example, when defining data object permissions in Oracle BAM, the user is not present in the list of known users presented by the permission editor.

The `registerusers` utility allows a user with Oracle BAM administrator-level privileges to register a list of users with Oracle BAM. The result of this registration is that these users are fully defined in Oracle BAM and available for further Oracle BAM user configuration, such as defining data object permissions.

The users are only added to Oracle BAM if they are successfully authenticated with the security provider. The input to this utility is a simple list of white-space delimited user names.

The `registerusers` utility confirms the successful registration of each user to standard out, and the failed registration of any user to standard error. The following is sample output from the `registerusers` utility:

```
>registerusers -file cmd_file.txt
Enter Password:
Connecting to BAM server as user weblogic.
Registering users...
Registration Succeeded For User: james.
Registration Failed    For User: jane. Reason: BAM-00400: Authentication failed.
```

```
Registration Succeeded For User: adam.
Registration Failed    For User: bob. Reason: BAM-00400: Authentication failed.
Registration Succeeded For User: bill.
Registration Succeeded For User: barbara.
Registration Succeeded For User: lynn.
```

If the utility is executed without any arguments, then the command syntax is displayed to standard out.

The syntax of the command is:

```
registerusers -adminuser adminusername -host host
-port port -protocol protocol [-file filename]... [username]...
```

where:

adminusername specifies the name of a user with Oracle BAM administrator privileges.

host specifies the hostname or IP address of the target Oracle BAM Server. Default: `localhost`

port specifies the port number of the target Oracle BAM server. Default: `7001`

protocol specifies the communication protocol to be used. Can be `t3` or `t3s` for SSL. Default: `t3`

filename specifies a file containing command line arguments. The `-file` option is replaced by the contents of the specified file.

username specifies the name of a user to be registered with Oracle BAM.

For example:

```
>registerusers -adminuser weblogic -host localhost -port 7001 -protocol t3 -file
cmd_file.txt smith jones
```

If the `-adminuser` option is omitted, then it is prompted for from standard input. The password for the BAM Administrator specified by `-adminuser` is prompted for from standard input using a password prompt where typed characters are hidden.

The file represented by `<filename>` should contain white-space delimited command-line arguments. White space is defined as blanks, EOL characters, and commas. Both single- and double-quoted strings are supported.

The file can also contain double-slash and slash-asterisk comments for documentation purposes. These comments are ignored.

If a command-line argument must contain any white-space characters (including commas), forward slashes (/) (as in file paths), or asterisks (*), then it must be quoted. For example, `"/dir1/dir2/file.txt"`.

The file can contain further `-file` options. The file should be in UTF-8 format to support extended character sets.

The following is a sample command file:

```
// Specify Oracle BAM administrator account to execute command. Password is not
// provided so that the user is securely prompted for the password.
-adminuser user_name
// Configure the target BAM server.
-host localhost -port 7001 -protocol t3
// Register the following users.
jones /* This is Mr. Jones. */
smith /* This is Ms. Smith. */
```

```
// Register users from another file.
-file "/dir1/dir2/user_list.txt"
```

Default values for `-host`, `-port`, and `-protocol` arguments are included in the `UserRegisterConfig.xml` file. These defaults are only used if the option is not specified on the command line. If these options are not specified on the command line, or in the `UserRegisterConfig.xml` configuration file, then the defaults are used. The command syntax output shows the default values that are in effect.

The defaults are:

```
-host localhost -port 7001 -protocol t3
```

The following is an example of a `UserRegisterConfig.xml` file.

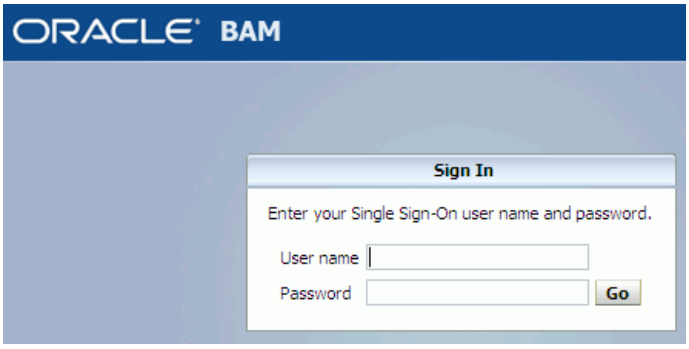
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<UserRegister>
  <ServerName>localhost</ServerName>
  <ServerPort>7001</ServerPort>
  <Communication_Protocol>t3</Communication_Protocol>
  <ServerPlatform>weblogic</ServerPlatform>
</UserRegister>
```

The `-host` default is picked up from the `ServerName` property, the `-port` default is picked up from the `ServerPort` property, and the `-protocol` default is picked up from the `Communication_Protocol` element. Information about these properties can be found in Table 23–3. `ServerPlatform` configuration is required for standalone applications. The value of this is `weblogic` or `websphere`, depending on the type of application server Oracle BAM Server is running on.

25.3.6.2 Populating By User Login

To populate users in Oracle BAM Administrator:

1. Have each Oracle BAM user open the Oracle BAM start page and log in using the credentials specified by the security provider.

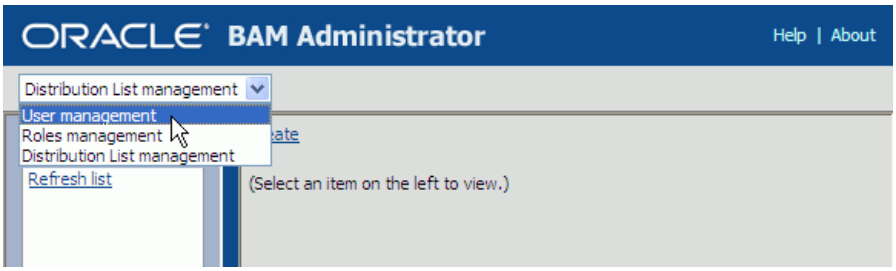


   The Oracle BAM start page can be found at:

   `http://`*`host_name`*`:`*`port_number`*`/OracleBAM/`

2. The administrator opens the Oracle BAM start page, logs in, and selects **Administrator**.
3. Select **User management** from the list, if the User Management page is not displayed in Oracle BAM Administrator.

Figure 25–1 Oracle BAM Administrator Function List



4. Verify that each Oracle BAM user appears in the **Users** list (use the **Refresh list** link to show the latest data).

25.3.7 Managing Oracle BAM Object Ownership

When Oracle BAM users are removed from the security provider, the user accounts continue to appear in the Oracle BAM Administrator application with an exclamation mark (!) icon to indicate that they are not valid.

These users do not have access to Oracle BAM applications, but remain visible because they may *own* objects in Oracle BAM that must be transferred to other users before the user is completely removed from Oracle BAM.

Object ownership is managed using Oracle BAM Administrator.

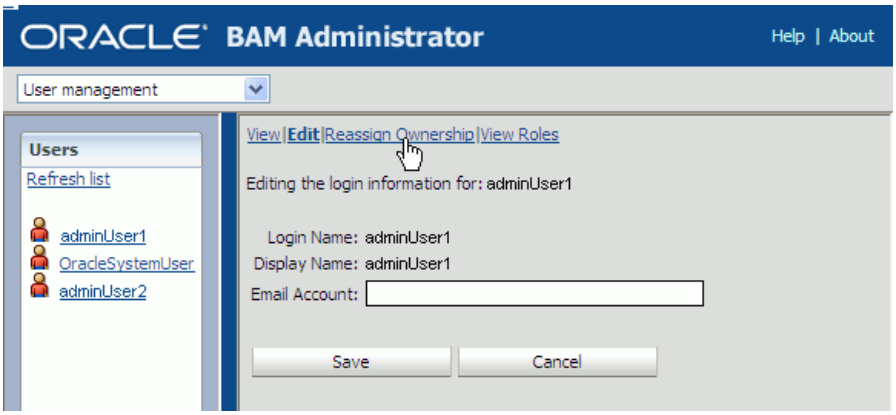
To transfer object ownership:

1. Go to the Oracle BAM start page, log in, and select **Administrator**.

   The Oracle BAM start page can be found at:

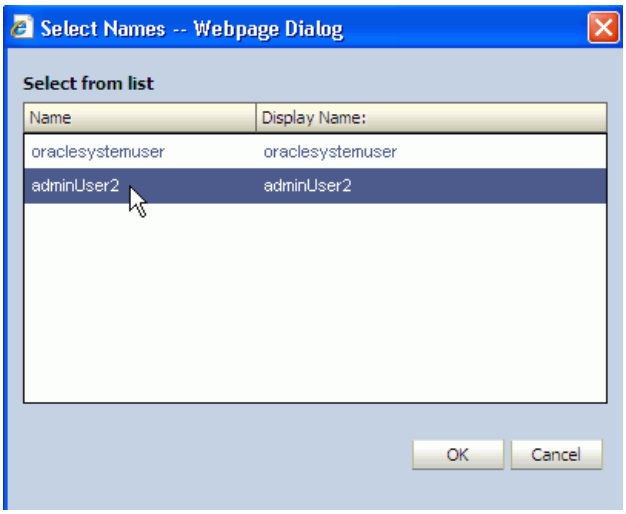
   `http://host_name:port_number/OracleBAM/`

2. Select **User management** from the list (see Figure 25–1).
3. Select the user in the **Users** list that currently owns the objects you can reassign.
4. Click **Reassign Ownership**.



   The **Select Names** dialog displays a list of users.

5. Select a user account in the list that becomes the new owner of the objects.



6. Click **OK**.

   Reports are moved to a subfolder named after the selected user name.

   Alerts are moved, and a zero (0) is appended to the alert name if the selected owner has an alert with the same name. If the alert needs an item updated or specified, an exclamation mark (!) is displayed on the alert icon in Oracle BAM Architect and Oracle BAM Active Studio Alerts pages.

   Shared reports and folders change ownership but are not moved.

25.3.8 Removing Invalid Users from Oracle BAM Administrator

Invalid users are not automatically removed from Oracle BAM because they may own reports and alerts that must be transferred to active Oracle BAM users. After those objects are transferred, an invalid user may be deleted from Oracle BAM Administrator.

To remove invalid users:

1. Go to the Oracle BAM start page, log in, and select **Administrator**.

   The Oracle BAM start page can be found at:

   `http://host_name:port_number/OracleBAM/`

2. Select **User management** from the list (see Figure 25–1).
3. Select an invalid user in the **Users** list whose objects have been reassigned.

   For information about reassigning objects, see Section 25.3.7, "Managing Oracle BAM Object Ownership".

   An inactive user has an exclamation point (!) icon next to the user name.

4. Click **Delete**.

Part X

Administering Oracle User Messaging Service

This part describes how to administer Oracle Messaging Service.

This part includes the following chapters:

- Chapter 26, "Configuring Oracle User Messaging Service"
- Chapter 27, "Monitoring Oracle User Messaging Service"
- Chapter 28, "Managing Oracle User Messaging Service"

26

Configuring Oracle User Messaging Service

This chapter describes the features and architecture of Oracle User Messaging Service (UMS). It also describes how to configure and secure Oracle UMS in your environment.

This chapter includes the following topics:

- Section 26.1, "Introduction to User Messaging Service"
- Section 26.2, "Introduction to Oracle User Messaging Service Configuration"
- Section 26.3, "Accessing User Messaging Service Configuration Pages"
- Section 26.4, "Configuring User Messaging Service Drivers"
- Section 26.5, "Configuring User Messaging Service Access to the LDAP User Profile"
- Section 26.6, "Securing the Oracle User Messaging Service"
- Section 26.7, "Troubleshooting Oracle User Messaging Service"

26.1 Introduction to User Messaging Service

Oracle User Messaging Service enables two-way communication between users and deployed applications. Key features include:

- Support for a variety of messaging channels: Messages can be sent and received through email, instant messaging (IM) (XMPP), short message service (SMS) (SMPP), and voice. Messages can also be delivered to a user's SOA/Oracle WebCenter Portal worklist.
- Two-way messaging: In addition to sending messages from applications to users (referred to as *outbound* messaging), users can initiate messaging interactions (inbound messaging). For example, a user can send an email or text message to a specified address; the message is routed to the appropriate application that can then respond to the user or invoke another process according to its business logic.
- User messaging preferences: End users can use a web interface to define preferences for how and when they receive messaging notifications. Applications immediately become more flexible; rather than deciding whether to send to a user's email address or IM client, the application can simply send the message to the user, and let UMS route the message according to the user's preferences.

> **Note:** The User Messaging Preferences UI is available at:
>
> `http://host:port/sdpmessaging/userprefs-ui`
>
> The User Messaging Preferences UI is also embedded in Oracle BPM Worklist. You can access it by choosing **Preferences** > **Notification**.

- Robust message delivery: UMS keeps track of delivery status information provided by messaging gateways, and makes this information available to applications so that they can respond to a failed delivery. Or, applications can specify one or more *failover* addresses for a message in case delivery to the initial address fails. Using the failover capability of UMS frees application developers from having to implement complicated retry logic.
- Pervasive integration within Oracle Fusion Middleware: UMS is integrated with other Fusion Middleware components providing a single consolidated bi-directional user messaging service.
  - Integration with Oracle BPEL Process Manager: Oracle JDeveloper includes prebuilt BPEL activities that enable messaging operations. Developers can add messaging capability to a SOA composite application by dragging and dropping the desired activity into any workflow.
  - Integration with human workflow: UMS enables the human workflow service engine to send actionable messages to and receive replies from users over email.
  - Integration with Oracle BAM: Oracle BAM uses UMS to send email alerts in response to monitoring events.
  - Integration with Oracle Oracle WebCenter Portal: UMS APIs are available to developers building applications for Oracle WebCenter Portal: Spaces. The API is a realization of Parlay X Web Services for Multimedia Messaging, version 2.1, a standard web service interface for rich messaging.

26.1.1 Components

There are three types of components that comprise Oracle User Messaging Service. These components are standard Java EE applications, making it easy to deploy and manage them using the standard tools provided with Oracle WebLogic Server.

- UMS Server: The UMS Server orchestrates message flows between applications and users. The server routes outbound messages from a client application to the appropriate driver, and routes inbound messages to the correct client application. The server also maintains a repository of previously sent messages in a persistent store, and correlates delivery status information with previously sent messages.
- UMS Drivers: UMS Drivers connect UMS to the messaging gateways, adapting content to the various protocols supported by UMS. Drivers can be deployed or undeployed independently of one another depending on what messaging channels are available in a given installation.
- UMS Client applications: UMS client applications implement the business logic of sending and receiving messages. A UMS client application might be a SOA application that sends messages as one step of a BPEL workflow, or a WebCenter Portal Spaces application that can send messages from a web interface.

In addition to the components that comprise UMS itself, the other key entities in a messaging environment are the external gateways required for each messaging

channel. These gateways are not a part of UMS or Oracle WebLogic Server. Since UMS Drivers support widely-adopted messaging protocols, UMS can be integrated with existing infrastructures such as a corporate email servers or XMPP (Jabber) servers. Alternatively, UMS can connect to outside providers of SMS or text-to-speech services that support SMPP or VoiceXML, respectively.

26.1.2 Architecture

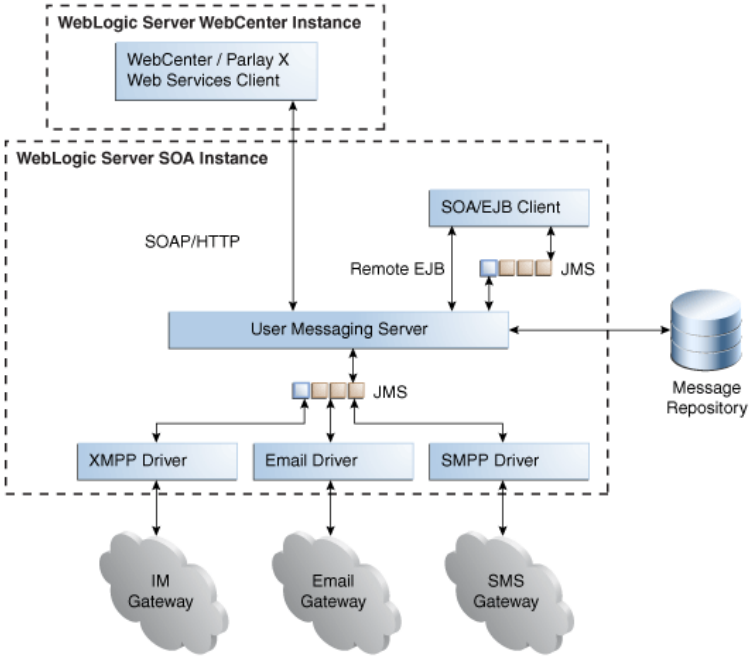
The system architecture of Oracle User Messaging Service is shown in Figure 26–1.

For maximum flexibility, the components of UMS are separate Java EE applications. This allows them to be deployed and managed independently of one another. For example, a particular driver can be stopped and reconfigured without affecting message delivery on all other channels.

Exchanges between UMS client applications and the UMS Server occur as SOAP/HTTP web service requests for web service clients, or through remote Enterprise JavaBeans (EJB) and JMS calls for BPEL messaging activities. Exchanges between the UMS Server and UMS drivers occur through JMS queues.

Oracle UMS server and drivers are installed alongside Oracle SOA Suite or Oracle BAM in their respective Oracle WebLogic Server instances. An Oracle WebCenter Portal installation includes the necessary libraries to act as a UMS client application, invoking a server deployed in a SOA instance.

Figure 26–1 UMS Architecture



26.2 Introduction to Oracle User Messaging Service Configuration

Oracle User Messaging Service enables users to receive notifications sent from SOA applications that are developed and deployed to the Oracle WebLogic Server using Oracle JDeveloper.

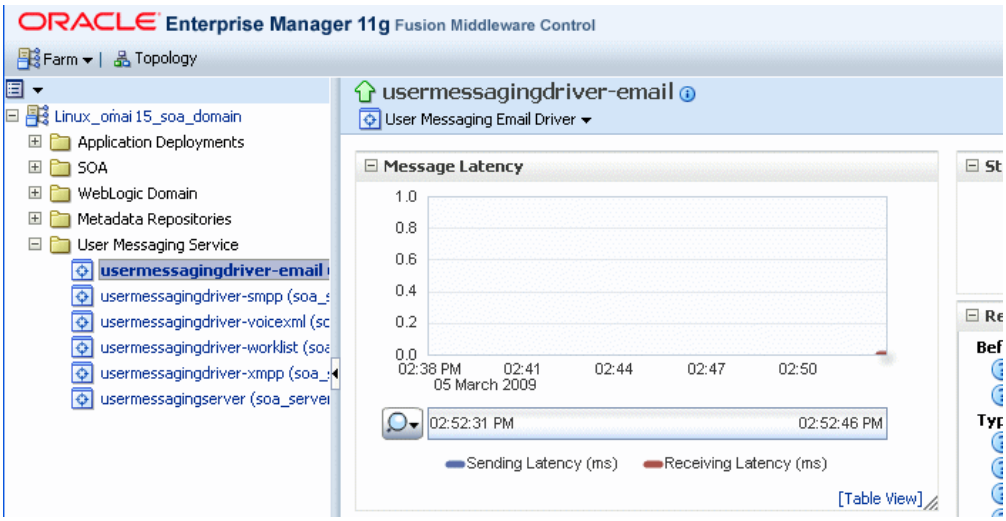
At the application level, there is notification activity for a specific delivery channel (such as SMS or email). For example, when you build a SOA application that sends

email notification, you drag and drop an Email activity from the Oracle JDeveloper Component Palette to the appropriate location within a workflow. The application connects then sends notifications.

For more information about Oracle JDeveloper, see your Oracle JDeveloper documentation.

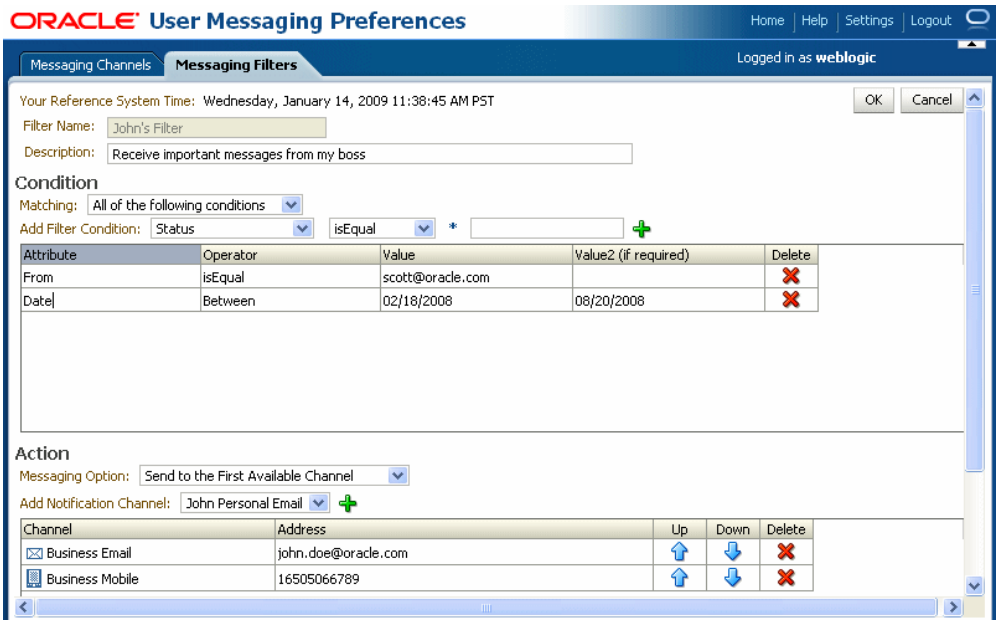
To enable the workflow participants to receive and forward notifications, use Oracle Enterprise Manager Fusion Middleware Control to set the Oracle User Messaging Service environment by configuring the appropriate driver instances that reside on the same Oracle WebLogic Server on which you deploy the workflow application (Figure 26–2). Oracle User Messaging Service includes drivers that support messaging through email, IM, SMS, and voice channels. For more information, see Section 26.4, "Configuring User Messaging Service Drivers."

Figure 26–2 Oracle Enterprise Manager Fusion Middleware Control



For workflow participants to receive the notifications, they must register the devices that they use to access messages through User Messaging Preferences (Figure 26–3).

Figure 26–3 User Messaging Preferences



26.3 Accessing User Messaging Service Configuration Pages

You configure Oracle User Messaging Service through Oracle Enterprise Manager Fusion Middleware Control.

26.3.1 Setting the Storage Method

Use the Server Properties page to set the deployment type for the Messaging Server (that is, select the storage method for runtime and management data) and add (or remove) the User Messaging Preference Business Terms that are used for creating message filters.

Select **Persistent** (the default) to enable entries and the Messaging Store to persist when the server has been restarted. In the Transient mode (which is recommended for lightweight deployments), the Messaging Server does not maintain any data stored in the Messaging Store after a restart.

26.3.2 Adding or Removing User Messaging Preferences Business Terms

The Server Properties page enables you to add or remove the business terms used to construct the message filters in User Message Preferences. For more information about building messaging filters with business terms, see Section 26.3.2.1, "Adding Business Terms."

26.3.2.1 Adding Business Terms

Note: Business terms are stored per server instance. If there are multiple instances (as in a cluster), then new business terms must be added to each instance individually.

To add a business term to User Messaging Preferences:

1. Click **Add**.
2. Enter a descriptive name for the business term.
3. Select a data type (string, number, or date).
4. Click **Apply**.

26.3.2.2 Removing Business Terms

To remove a business term from User Messaging Preferences:

1. Select the business term.
2. Click **Delete**.
3. Click **Apply** to confirm the new term.

26.4 Configuring User Messaging Service Drivers

Oracle User Messaging Service includes the following drivers.

- Configuring the Messaging Extension Driver
- Configuring the Email Driver
- Configuring the SMPP Driver
- Configuring the XMPP Driver
- Configuring the VoiceXML Driver
- Configuring the Worklist Driver
- Configuring the Proxy Driver

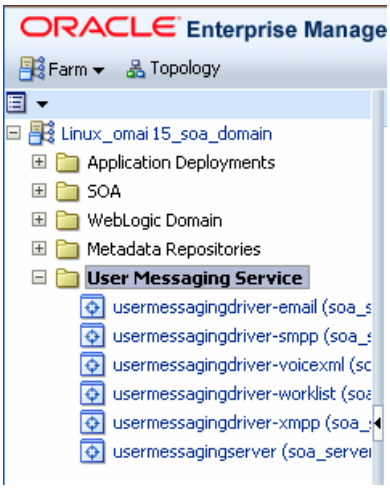
Note: For the cluster environment, when you use separate messaging drivers for separate managed server nodes, all the drivers must be configured separately.

UMS Messaging Drivers are configured per instance. Configuring only one does not populate the configuration values to the drivers on the other cluster nodes.

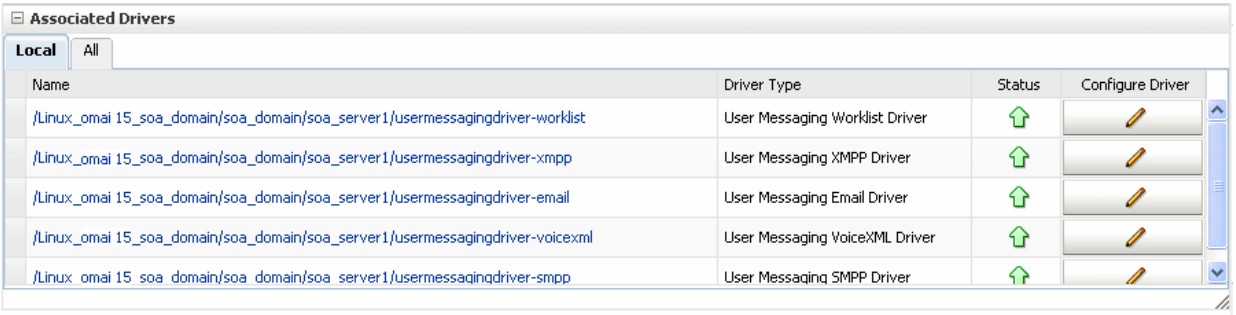
26.4.1 Configuring a Driver

To configure a driver:

1. Log in to Oracle Enterprise Manager Fusion Middleware Control as an administrator.
2. Expand the **Fusion Middleware** folder.
3. Navigate to the User Messaging Service home page.



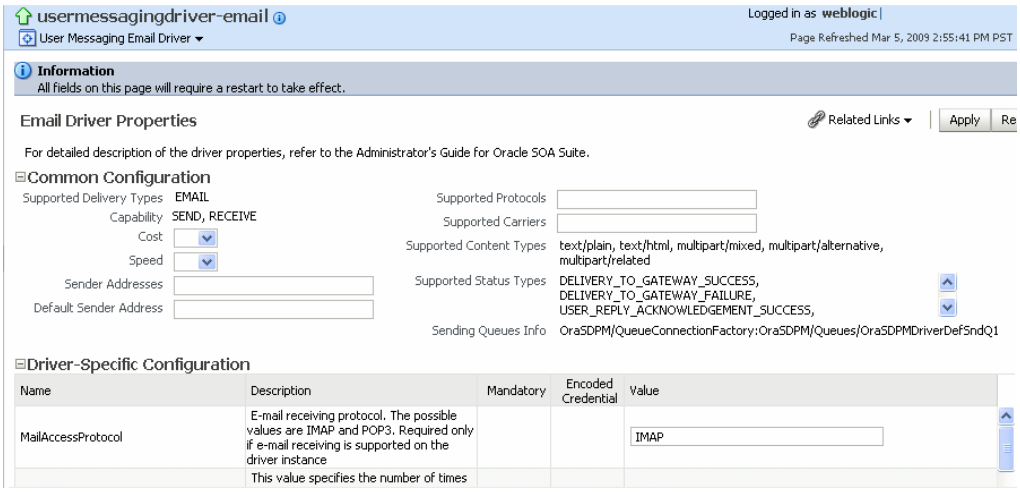
4. Click **usermessagingserver(soa_server1)**. The Associated Drivers page appears.



5. Select the **Local** tab to access the drivers collocated with the UMS server instance. These drivers may or may not be registered with the UMS server depending on whether they are properly configured. The **ALL** tab lists all drivers that are deployed in the domain and registered to all the UMS server instances.

6. Find the email driver in the list, and then click the adjacent **Configure Driver** icon.

   The configuration page is displayed.



7. If needed, expand the **Driver-Specific Configuration** section and configure the driver parameters. For more information, see Section 26.4.1.1, "Introduction to Driver Properties."

26.4.1.1 Introduction to Driver Properties

Oracle User Messaging Service drivers share common properties (listed in Table 26–1) that are used by the Messaging Engine when routing outbound messages. Typically, administrators set such Quality of Service (QoS) properties as driver cost (Cost) and driver speed (Speed), supported carriers (SupportedCarriers), and supported protocols (SupportedProtocols). Driver developers configure properties that typically do not require modification by the administrator, such as supported delivery types (SupportedDeliveryTypes), and supported content types (SupportedContentTypes).

> **Note:** Properties such as SendingQueuesInfo are for advanced use and only require modification for advanced deployment topologies.

Table 26–1 Common Driver Properties

Name	Description	Mandatory Property?
Capability	Sets the driver's capability to send or receive messages. The values are SEND, RECEIVE, and BOTH.	Yes
Cost	The cost level of the driver (from 0 - 10). 0 is least expensive; 10 is most expensive. If the value is not in this range, cost is considered to be 0.	No
DefaultSenderAddress	The default address of the sender. The driver uses these addresses when sending a message that has no sender address specified, or when the specified sender address is not in the sender addresses list and the driver does not support using the application-provided sender address.	No
SenderAddresses	The list of sender addresses that the driver supports. If provided by the driver, the messaging engine can use this to route a sending message to the driver by matching against the sender address of the message.	No
SendingQueuesInfo	The information for the driver sending queue.	Yes
Speed	The speed level of the driver (from 0-10, with 10 being the fastest).	No
SupportedCarriers	A comma-delimited list of supported carriers.	No
SupportedContentTypes	The content type supported by the driver.	Yes
SupportedDeliveryTypes	The delivery types supported by the driver.	Yes
SupportedProtocols	A comma-delimited list of supported protocols. Enter an asterisk (*) for any protocol.	No
SupportedStatusTypes	The status types supported by the driver.	No
SupportsCancel	Supports a cancel operation on a message.	No
SupportsReplace	Supports a replace operation on a message.	No
SupportsStatusPolling	For certain protocols, an active polling of the remote gateway must be performed to check the status of a message previously sent. This property indicates whether the driver supports such status polling. If set to true, the messaging engine invokes the driver connection's `getStatus()` operation.	No
SupportsTracking	Supports a tracking operation on a message.	No

26.4.1.2 Securing Passwords

Sensitive driver properties (namely, passwords) can be stored securely in the credential store using Oracle Enterprise Manager Fusion Middleware Control. Properties are marked with the flag **Encoded Credential** and have a custom entry form field.

To store a sensitive driver property securely:

1. Go to the driver configuration page of the selected driver.
2. In the **Driver-Specific Configuration** section, locate the property with the **Encoded Credential** flag set.
3. Select the credential type. (Depending on the selected credential type, you are prompted to enter the username and/or password.) There are three options:
   - Indirect password, create new user (default option): specify the username and real password; the password is stored in the credential store with the username as part of the key. The key and a fixed folder (map name) are stored in the driver deployment's `driverconfig.xml` file.
   - Indirect password, use existing user: choose an existing username/key in the credential store (to reference the password you stored previously).
   - User a clear text password: specify the password, and it is stored directly in the `driverconfig.xml` file.
4. Click **Apply** to save the changes.
5. Restart the driver application or the container for the changes to take effect.

You can check the password in the driver deployment directory's `driverconfig.xml` file. For an indirect password, the format is:

```
value="->mapName:keyName"    (mapName is the driver target name, and the key is
<parameter_name>.<username>)
```

For example, here is a sample entry in `driverconfig.xml` for an email driver's `OutgoingPassword` property:

```
<Property value="-&gt;
/Farm_base_domain/base_domain/server_soa/usermessagingdriver-email:
OutgoingPassword.ouser" encodedCredential="true"
type="java.lang.String" mandatory="no" name="OutgoingPassword"
description="oracle.sdp.messaging.EmailDriverConfig.outgoingPassword"/>
```

26.4.1.3 Configuring the Messaging Extension Driver

The extension driver extends the messaging capability of the User Messaging Service by enabling support for arbitrary administrator-defined channels (protocols) and delivering the notifications for such channels to an administrator-defined web service listener endpoint.

> **Note:** An instance of this driver is deployed, but not targeted to any servers in the default installation. To enable this driver instance, it must be targeted to the appropriate servers where UMS (`usermessagingserver`) is running.

26.4.1.3.1 Driver Application Archive (EAR) The EAR file is `$oracle_home/communications/applications/sdpmessagingdriver-extension.ear`.

26.4.1.3.2 Common Properties These are common driver properties that are indicative of the capabilities of this driver for use by the messaging engine when routing outbound messages. Some properties are set by the driver developer and do not normally require modification, while others can be modified by the administrator to change the routing behavior. Some properties (such as `sendingqueuesinfo`) are for advanced use and only require modification for advanced deployment topologies. See Table 26–2. For a complete description of these properties and available values, see the JavaDoc for `driverconfigpropertynames`.

Table 26–2 Extension Driver Common Properties

Name	Description	Mandatory?	Default Value
InstanceName	Instance name (for internal use only)	Yes	Extension-Driver
Capability	Message sending and receiving capability	Yes	SEND
SupportedDeliveryTypes	Supported delivery types	Yes	URI
SupportedContentTypes	Supported content types	Yes	text/plain, text/html, text/xml
SupportedStatusTypes	Supported status types	No	DELIVERY_TO_GATEWAY_SUCCESS, DELIVERY_TO_GATEWAY_FAILURE
Cost	Cost	No	
Speed	Speed	No	
SupportedCarriers	Supported carriers	No	
SupportedProtocols	Supported protocols	No	popup
SupportsCancel	Supports cancel operation on the message	No	False
SupportsReplace	Supports replace operation on the message	No	False
SupportsTracking	Supports tracking operation on the message	No	False
SupportsStatusPolling	Supports status polling operation on the message	No	False
SenderAddresses	Sender addresses	No	
DefaultSenderAddress	Default sender address	No	

Table 26–2 (Cont.) Extension Driver Common Properties

Name	Description	Mandatory?	Default Value
SendingQueuesInfo	Driver sending queue info	Yes	`OraSDPM/QueueConnectionFactory:OraSDPM/Queues/OraSDPMDriverDefSndQ1`

26.4.1.3.3 Custom Properties This driver supports multiple configuration groups called extension endpoint groups. An extension endpoint group holds the configuration for a remote endpoint at which to deliver extension notifications. Each endpoint must have a distinct combination of protocol and mapped domain. The properties of the extension endpoint group are listed in Table 26–3:

Table 26–3 Extension Driver Custom Properties

Name	Description	Mandatory?
Group Name	The name of this extension endpoint configuration group.	Yes
Endpoint URL	Remote endpoint listener URL.	Yes
Mapped Domain	The extension endpoint used to deliver messages where the domain part of the recipient URI matches this value.	No
Protocol	The extension endpoint used to deliver messages where the protocol (scheme) part of the recipient URI matches this value.	Yes
Security Policies	Comma-separated list of WS-Security policies to apply to this endpoint.	No
Username	Username to propagate through WS-Security headers.	No
Keystore Alias	Keystore alias to use for looking up WS-Security policy public keys.	No
Credential Store Key	Key to use for looking up the WS-Security username and password from the Oracle Web Services Management credential store map.	No

26.4.1.3.4 Extension Driver Security If the remote extension endpoint is secured using WS-Security, then additional configuration of the extension driver is required. There are two typical WS-Security configurations that are supported. The extension driver can either use SAML tokens or username tokens.

To use extension driver security:

1. To use SAML tokens, the Security Policies configuration property should contain value `oracle/wss11_saml_token_identity_switch_with_message_protection_client_policy`, and the Keystore Alias configuration property should contain a valid alias for keystore entries that is accepted by the remote extension endpoint.
2. To use username tokens, the Security Policies configuration property should contain value `oracle/wss11_username_token_with_message_protection_client_policy`, and the Credential Store Key configuration property should contain a valid alias for a credential store entry that is accepted by the remote extension endpoint.

For more details about using WS-Security policies, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

26.4.1.3.5 Client API Messageinfo Support Table 26–4 describes whether the protocol or driver implementation honors the following message delivery-related properties that are specified through the client API.

Table 26–4 Client API MessageInfo Support

Name	Description	Supported?
Expiration	Expiration means how much later in seconds for the message to expire.	False
Delay	Delay means how much later to send the message out.	False

26.4.1.3.6 Usage Instructions Perform the following steps to use the extension driver:

To use the extension driver:

1. Implement and deploy a web service listener endpoint based on the `MessagingNotifyService` WSDL (`umsnotify.wsdl`):

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
                  xmlns:tns="http://xmlns.oracle.com/ucs/messaging/extension"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  name="MessagingNotifyService"

targetNamespace="http://xmlns.oracle.com/ucs/messaging/extension">

    <wsdl:types>

    <xsd:schema
targetNamespace="http://xmlns.oracle.com/ucs/messaging/extension">
      <xsd:element name="notification">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="messageId" type="xsd:string" minOccurs="0"
 maxOccurs="1">
              <xsd:annotation>
                <xsd:documentation>Unique message identifier from User
 Messaging Service.</xsd:documentation>
              </xsd:annotation>
            </xsd:element>
            <xsd:element name="sender" type="xsd:string">
              <xsd:annotation>
                <xsd:documentation>The sender address.</xsd:documentation>
              </xsd:annotation>
            </xsd:element>
            <xsd:element name="recipient" type="xsd:string">
              <xsd:annotation>
                <xsd:documentation>The recipient address (typically
 username).</xsd:documentation>
              </xsd:annotation>
            </xsd:element>
            <xsd:element name="subject" type="xsd:string" minOccurs="0"
 maxOccurs="1">
              <xsd:annotation>
                <xsd:documentation>The subject of the message, if
 available.</xsd:documentation>
              </xsd:annotation>
            </xsd:element>
            <xsd:element name="contentType" type="xsd:string"
```

```
default="text/plain">
            <xsd:annotation>
              <xsd:documentation>The MIME type of the message. e.g.
text/plain, text/html, text/xml.</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="content" type="xsd:string">
            <xsd:annotation>
              <xsd:documentation>The main body of the message. Textual
content only (no binary content).</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="parameters" type="tns:parameter" minOccurs="0"
maxOccurs="unbounded">
            <xsd:annotation>
              <xsd:documentation>Additional key-value pairs. This interface
does not define any specific key-value pair meanings. Use of such parameters
is defined on a private basis by particular implementations of this interface.
  </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:complexType name="parameter">
      <xsd:sequence>
        <xsd:element name="name" type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>Parameter name</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="value" type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>Parameter value</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="notificationResponse">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="messageId" type="xsd:string" minOccurs="0"
maxOccurs="1">
            <xsd:annotation>
              <xsd:documentation>A message identifier returned in response to
successfully accepting the message. If returned, the identifier should be
unique. Note: A fault is raised if the message cannot be
accepted.</xsd:documentation>
            </xsd:annotation></xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="notificationFault">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="code" type="xsd:string"/>
          <xsd:element name="message" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
```

```
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="notifyRequest">
    <wsdl:part element="tns:notification" name="parameters" />
  </wsdl:message>
  <wsdl:message name="notifyResponse">
    <wsdl:part element="tns:notificationResponse" name="parameters"/>
  </wsdl:message>
  <wsdl:message name="notifyException">
    <wsdl:part element="tns:notificationFault" name="parameters"/>
  </wsdl:message>
  <wsdl:portType name="Notify">
    <wsdl:operation name="invoke">
      <wsdl:input message="tns:notifyRequest"/>
      <wsdl:output message="tns:notifyResponse"/>
      <wsdl:fault message="tns:notifyException" name="NotifyException"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="NotifySOAPBinding" type="tns:Notify">
    <soap:binding style="document"
 transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="invoke">
      <soap:operation
        soapAction="http://www.oracle.com/ucs/messaging/extension" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
      <wsdl:fault name="NotifyException">
        <soap:fault name="NotifyException" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>

  <wsdl:service name="NotifyService">
    <wsdl:port binding="tns:NotifySOAPBinding" name="Notify">
      <soap:address location="http://localhost:8001/NotifyService"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

2. Configure the extension driver.

   a. Target the predeployed extension driver called `usermessagingdriver-extension` or a new deployment to the appropriate servers where UMS (`usermessagingserver`) is running and start the driver.

   b. In Enterprise Manager Fusion Middleware Control, navigate to the **usermessagingserver** home page.

   c. Click **User Messaging Service > Driver Properties**.

   d. Select and Edit the driver `usermessagingdriver-extension` or create a new driver with the same name as your new driver deployment.

   e. Under **Driver-Specific Configuration**, add a new extension endpoint configuration group and specify the correct properties: **EndpointURL** is the URL to the web service listener endpoint created in Step 1. **Protocol** is the

value of the new messaging channel for which you want to add notification support (for example, **popup**).

f. Under **Common Configuration**, update **Supported Protocols** with a comma-separated list of protocols defined in each Extension Endpoint group.

g. Click **OK** to save the configuration.

This completes the configuration and integration of a new messaging channel (protocol) in UMS using the extension driver.

To send notifications to this new channel (protocol), recipients must be specified for the URI delivery type using the URI addressing format:

```
URI:scheme:scheme-specific-address-value
```

where *scheme* is the protocol. The URI delivery type is optional. For example, if the extension driver was configured to support the protocol popup, an application can compose a message to `popup:john.doe@example.com`.

End users can also declare their messaging preferences by creating a new messaging channel for the new channel type in the Worklist/UMS Preferences UI. User preferences are only applied when applications send user-based notifications (that is, to recipients of the form *USER:username*).

Note: Proper configuration of SSL/TLS in the Oracle WebLogic Server container is a prerequisite for secure connections between UMS and the email server. See "Configuring SSL" in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

26.4.1.4 Configuring the Email Driver

The email driver both sends and receives messages (that is, its **capability** property is set to **both** by default). The email driver sends messages over SMTP and uses either IMAP or POP3 for receiving messages.

26.4.1.4.1 Email Driver Interoperability This section details interoperability features of the email driver.

The email driver is compatible with these protocols: POP3, IMAP4, and SMTP.

Email driver features include:

- Automatic connection retry
- SMTP for message sending
- IMAP4 and POP3 for message receiving (using polling)
- scalable, highly available
- Message loss prevention and duplication avoidance

The gateway vendors and versions in Table 26–5 have been verified.

Table 26–5 Email Driver Gateway Vendors and Versions

Vendor	Version
Oracle Beehive	Release 1 (1.4.3)
Oracle Collaboration Suite	10*g* Release 1 (10.1.2)
Microsoft Exchange	2003

Table 26–5 (Cont.) Email Driver Gateway Vendors and Versions

Vendor	Version
Dovecot (IMAP4/POP3)	0.99.11
sendmail (SMTP)	8.13.1

26.4.1.4.2 Common Properties Table 26–6 lists common driver properties that are indicative of the capabilities of this driver for use by the messaging engine when routing outbound messages. Some properties are set by the driver developer and do not normally require modification, while others can be modified by the administrator to change the routing behavior. Some properties such as **sendingqueuesinfo** are for advanced use and only require modification for advanced deployment topologies. For a complete description of these properties and available values, see the JavaDoc of `driverconfigpropertynames`.

Table 26–6 Common Email Properties

Name	Description	Mandatory	Default Value
InstanceName	Instance name (for internal use only)	Yes	Email-Driver
Capability	Message sending and receiving capability	Yes	Both
SupportedDeliveryTypes	Supported delivery types	Yes	Email
SupportedContentTypes	Supported content types	Yes	text/plain, text/html, multipart/mixed, multipart/alternative, multipart/related
SupportedStatusTypes	Supported status types	No	DELIVERY_TO_ GATEWAY_SUCCESS, DELIVERY_TO_ GATEWAY_FAILURE, USER_REPLY_ ACKNOWLEDGEMENT _SUCCESS, USER_ REPLY_ ACKNOWLEDGEMENT _FAILURE
Cost	Cost	No	N/A
Speed	Speed	No	N/A
SupportedCarriers	Supported carriers	No	N/A
Supported Protocols	Supported protocols	No	N/A
SupportsCancel	Supports cancel operation on the message	No	False
SupportsReplace	Supports replace operation on the message	No	False
SupportsTracking	Supports tracking operation on the message	No	False
SupportsStatusPolling	Supports status polling operation on the message	No	False
SenderAddresses	Sender addresses	No	N/A
DefaultSenderAddress	Default sender address	No	N/A

Table 26–6 (Cont.) Common Email Properties

Name	Description	Mandatory	Default Value
SendingQueuesInfo	Driver sending queue info	Yes	OraSDPM/QueueConnectionFactory:OraSDPM/Queues/OraSDPMDriverDefSndQ1

26.4.1.4.3 Email Custom Properties Table 26–7 lists properties specific to this driver and generally associated with configuring access to the remote gateway and certain protocol or channel-specific behavior.

Table 26–7 Custom Email Properties

Name	Description	Mandatory?	Default Value
MailAccessProtocol	Email receiving protocol. The possible values are IMAP and POP3. Required only if email receiving is supported on the driver instance.	No	IMAP
RetryLimit	This value specifies the number of times to retry connecting to the incoming mail server, if the connection is lost for some reason. The default value is **-1**, which means there is no limit to the number of tries.	No	-1
MailDelFreq	The frequency to permanently remove deleted messages. The unit is in seconds and the default value is 600 seconds. A negative value indicates the messages should not be expunged. For the POP3 protocol, the message is expunged after it is processed.	No	600
AutoDelete	This value indicates if the driver should mark the messages deleted after they have been processed. The default is Disabled. For the POP3 protocol, the messages are always deleted right after they are processed.	No	Disabled
Debug	This value indicates if the driver is running in Debug mode. When enabled, JavaMail prints out requests and responses between the email driver and the mail server to Fusion Middleware Control. The default is Disabled.	No	Disabled
CheckMailFreq	The frequency with which to retrieve messages from the mail server. The unit is in seconds and the default value is 30 seconds.	No	30
ReceiveFolder	The name of the folder from which the driver is polling messages. The default value is INBOX.	No	INBOX

Table 26–7 (Cont.) Custom Email Properties

Name	Description	Mandatory?	Default Value
OutgoingMailServer	The name of the SMTP server. This is mandatory only if email sending is required.	No	N/A
OutgoingMailServerPort	The port number of the SMTP server; typically **25**.	No	25
OutgoingMailServerSecurity	The security setting used by the SMTP server. Possible values are None, TLS, and SSL. The default value is None.	No	None
OutgoingDefaultFromAddr	The default FROM address (if one is not provided in the outgoing message).	No	N/A
OutgoingUsername	The username used for SMTP authentication. Required only if SMTP authentication is supported by the SMTP server.	No	N/A
OutgoingPassword	The password used for SMTP authentication. This is required only if SMTP authentication is supported by the SMTP server. This includes Type of Password (choose from Indirect Password/Create New User, Indirect Password/Use Existing User, and Use Cleartext Password) and Password.	No	N/A
IncomingMailServer	The hostname of the incoming mail server. Required only if email receiving is supported on the driver instance.	No	N/A
IncomingMailServerPort	Port number of IMAP4 (that is, **143** or **993**) or POP3 (that is, **110** or **995**) server.	No	N/A
IncomingMailServerSSL	Indication to enable SSL when connecting to IMAP4 or POP3 server. The default is Disabled.	No	Disabled
IncomingMailIDs	The email addresses corresponding to the user names. Each email address is separated by a comma and must reside in the same position in the list as their corresponding user name appears on the usernames list. Required only if email receiving is supported on the driver instance.	No	N/A
IncomingUserIDs	The list of user names of the mail accounts from which the driver instance is polling. Each name must be separated by a comma, for example, foo,bar. This is required only if email receiving is supported on the driver instance.	No	N/A

Table 26–7 (Cont.) Custom Email Properties

Name	Description	Mandatory?	Default Value
IncomingUserPasswords	The list of passwords corresponding to the user names. Each password is separated by a comma and must reside in the same position in the list as their corresponding user name appears on the usernames list. This is required only if email receiving is supported on the driver instance. This includes Type of Password (choose from Indirect Password/Create New User, Indirect Password/Use Existing User, and Use Cleartext Password) and Password.	No	N/A
ProcessingChunkSize	The number of messages processed during each message polling. The default is 100.	No	100
ImapAuthPlainDisable	Indication to disable or enable plain text authentication (`AUTHENTICATE PLAIN` command) for IMAP user authentication. The default is Disabled.	No	Disabled. When this property is disabled, that means that plain text is allowed.

26.4.1.4.4 Client API Messageinfo Support These properties are message delivery-related that are specified through client API. Table 26–8 describes if the protocol or driver implementation honors such properties.

Table 26–8 Client API MessageInfo Support

Name	Description	Support
Expiration	Expiration means how long the message may exist until it expires.	False
Delay	Delay means the amount of time that must elapse before the message is sent.	False

26.4.1.5 Configuring the SMPP Driver

Short Message Peer-to-Peer (SMPP) is a popular GSM SMS protocols. User Messaging Service includes a prebuilt implementation of the SMPP protocol as a driver that can send and receive short messages. If the sending feature is enabled, the SMPP driver opens one TCP connection to the Short Message Service Center (SMS-C) as a transmitter for sending. If the driver's receiving feature is enabled, it opens another connection to the SMS-C as a receiver for receiving. Only two TCP connections (both initiated by the driver) are needed for all communication between the driver and the SMS-C.

> **Note:** The SMPP Driver implements Version 3.4 of the SMPP protocol and only supports connections to an SMS-C or an SMS gateway that supports this version.

26.4.1.5.1 SMPP Driver Interoperability This section details interoperability features of the SMPP Driver.

The SMPP driver is compatible with this protocol: SMPP v3.4.

SMPP driver features include:

- Automatic connection retry
- HTTP proxy for firewall traversal
- Authentication configuration
- Configurable chunk size
- Bulk Sending
- Encoding: UCS2, IA5, GSM_DEFAULT
- Priority Setting
- Configurable Window size
- Plain text content only

The Gateway vendors in Table 26–9 have been verified.

Table 26–9 SMPP Driver Gateway Vendors

Vendor
Syniverse
Clickatell
Logica CMG
OpenSMPP (simulator)

26.4.1.5.2 Common Properties Table 26–10 lists common driver properties that are indicative of the capabilities of this driver for use by the messaging engine when routing outbound messages. Some properties are set by the driver developer and do not normally require modification, while others can be modified by the administrator to change the routing behavior. Some properties such as SendingQueuesInfo are for advanced use and only require modification for advanced deployment topologies. For a complete description of these properties and available values see the JavaDoc of `DriverConfigPropertyNames`.

Table 26–10 Common SMPP Properties

Name	Description	Mandatory	Default Value
InstanceName	Instance name (for internal use only)	Yes	SMPP-Driver
Capability	Message sending and receiving capability	Yes	Both
SupportedDeliveryTypes	Supported delivery types	Yes	SMS
SupportedContentTypes	Supported content types	Yes	text/plain
SupportedStatusTypes	Supported status types	No	DELIVERY_TO_ GATEWAY_SUCCESS, DELIVERY_TO_ GATEWAY_FAILURE
Cost	Cost	No	N/A
Speed	Speed	No	N/A
SupportedCarriers	Supported carriers	No	N/A

Table 26–10 (Cont.) Common SMPP Properties

Name	Description	Mandatory	Default Value
Supported Protocols	Supported protocols	No	N/A
SupportsCancel	Supports cancel operation on the message	No	False
SupportsReplace	Supports replace operation on the message	No	False
SupportsTracking	Supports tracking operation on the message	No	False
SupportsStatusPolling	Supports status polling operation on the message	No	False
SenderAddresses	Sender addresses	No	N/A
DefaultSenderAddress	Default sender address	No	N/A
SendingQueuesInfo	Driver sending queue info	Yes	OraSDPM/QueueConnectionFactory:OraSDPM/Queues/OraSDPMDriverDefSndQ1

26.4.1.5.3 Custom Properties Table 26–11 lists properties specific to this driver and generally associated with configuring access to the remote gateway and certain protocol or channel-specific behavior.

Table 26–11 Custom SMPP Properties

Name	Description	Mandatory?	Default Value
SmsAccountId	The Account Identifier on the SMS-C.	Yes	N/A
SmsServerHost	The name (or IP address) of the SMS-C server.	Yes	N/A
TransmitterSystemId	The account ID that is used to send messages.	Yes	N/A
ReceiverSystemId	The account ID that is used to receive messages.	Yes	N/A
TransmitterSystemType	The type of transmitter system. The default is Logica.	Yes	The default value is Logica.
ReceiverSystemType	The type of receiver system. The default is Logica.	Yes	The default value is Logica.
TransmitterSystemPassword	The password of the transmitter system. This includes Type of Password (choose from Indirect Password/Create New User, Indirect Password/Use Existing User, and Use Cleartext Password) and Password.	Yes	N/A

Table 26–11 (Cont.) Custom SMPP Properties

Name	Description	Mandatory?	Default Value
ReceiverSystemPassword	The password for the receiver system. This includes Type of Password (choose from Indirect Password/Create New User, Indirect Password/Use Existing User, and Use Cleartext Password) and Password.	Yes	N/A
ServerTransmitterPort	The TCP port number of the transmitter server.	Yes	N/A
ServerReceiverPort	The TCP port number of the receiver server.	Yes	N/A
DefaultEncoding	The default encoding of the SMPP driver. The default is IA5. Choose from the drop-down list: IA5, UCS2, and GSM_DEFAULT.	No	IA5
EncodingAutoDetect	If enabled, the SMPP driver encodes automatically. The default is Enabled.	No	Enabled
LocalSendingPort	The local TCP port used by the SMPP driver to send messages to the SMS-C.	No	N/A
LocalReceivingPort	The local TCP port used by the SMPP driver to receive messages from the SMS-C.	No	N/A
LocalAddress	The hostname (or IP address) of the server that hosts the SMPP driver.	No	N/A
WindowSize	The window size for SMS. This value must be a positive number. Default is **1**.	No	1
EnquireInterval	The interval, in seconds, to send an enquire message to the SMS-C. The default is 30 seconds.	No	30
ThrottleDelay	The delay, in seconds, between throttles. Default is 30.	No	30
BindRetryDelay	The minimum delay, in seconds, between bind entry attempts. Default is 30.	No	30
ResponseTimer	Time lapse allowed between SMPP request and response, in seconds. The default is 30.	No	30
RegisteredDeliveryMask	The registered delivery bit mask. The default is 0xFF, which does not change the delivery flag value.	No	0xFF

Table 26–11 (Cont.) Custom SMPP Properties

Name	Description	Mandatory?	Default Value
RangeSetNull	Set to true to set the address range field of BIND_ RECEIVER to null. Set to false (the default value) to set the address range field to SmsSystemId. The default is Disabled.	No	Disabled
PriorityAllowed	The highest priority allowed for the SMPP driver. The range is 0 (normal) to 3 (highest). The default is 0.	No	0
BulkSending	Setting this value to enabled (the default) enables sending messages in bulk to the SMS-C.	No.	Enabled
PayloadSending	Determines if the message_ payload parameter is always used when sending a message to the SMS-C. The default is Disabled.	No	Disabled
SourceTon	The type of number (TON) for ESME address(es) served through SMPP receiver session. The default is 0.	No	0
SourceNpi	The numbering plan indicator (NPI) for ESME address(es) served through the SMPP receiver session. The default is 0.	No	0
DestinationTon	The TON for destination. The default is 0.	No	0
DestinationNpi	The NPI for destination. The default is 0.	No	0
ExtraErrorCode	A comma-separated list of error codes.	No	N/A
MaxChunks	The maximum SMS chunks for a message. The default is **-1** (no maximum).	No	-1 (no maximum)
ChunkSize	The size of each SMS message chunk. Default is *160*.	No	160
LongMessageSending	Supports sending long messages. The default is Disabled.	No	Disabled
DatagramMessageMode	Supports datagram message mode. The default is Disabled.	No	Disabled

26.4.1.5.4 Client API MessageInfo Support These properties are message delivery-related that are specified through client API. Table 26–12 describes if the protocol or driver implementation honors such properties.

Table 26–12 Client API MessageInfo Support

Name	Description	Support
Expiration	Expiration means how long the message may exist until it expires.	True
Delay	Delay means the amount of time that must elapse before the message is sent.	False

26.4.1.6 Configuring the XMPP Driver

The XMPP Driver provides unidirectional and bidirectional access from Oracle Fusion Middleware to end users for real-time IM through the Extensible Messaging and Presence Protocol (XMPP). This driver enables end users to receive alert notifications or interactively chat with applications through their IM client of choice.

26.4.1.6.1 Introduction to XMPP XMPP is an open, XML-based protocol for IM and Presence. XMPP-based software is deployed on thousands of servers across the Internet and is used by millions of people worldwide. XMPP consists of a client/server architecture, which resembles the ubiquitous email network. XMPP servers are completely decentralized, allowing anyone to set up their own server. Messaging is achieved as in the email network, where recipients are addressed by an XMPP ID (or Jabber ID or JID) with the following form: `[username]@domain[/resource]`. See RFC 3920 for details on the addressing scheme.

In an XMPP network, users identified by their XMPP IDs as mentioned above (which typically consist of a username and the domain of the XMPP server to which the user connects). An end user of XMPP connects to an XMPP server using an XMPP client to send instant messages to other XMPP users. XMPP, however, is not the only protocol network available for IM. XMPP has an extensible and modular architecture. It integrates with proprietary IM networks, enabling XMPP users to communicate with those on other networks.

To use the XMPP Driver in UMS, you must have access to a Jabber/XMPP server and an XMPP account for the UMS XMPP Driver instance with which to log in.

26.4.1.6.2 XMPP Driver Interoperability This section details interoperability features of the XMPP Driver.

The XMPP driver is compatible with these protocols: XMPP (RFC 3920, 3921).

XMPP Driver features include:

- Automatic connection retry
- HTTP proxy for firewall traversal
- Plain text content only

The gateway vendors and versions in Table 26–13 have been verified.

Table 26–13 XMPP Driver Gateway Vendors and Versions

Vendor	Version
ejabberd	2.1.3
jabberd2	2.2.14
jabberd14	1.6.1.1-p1
Oracle Beehive	2.0.1.2.1

26.4.1.6.3 Third-Party Software The XMPP Driver uses or requires the following third-party software (you may optionally choose to install and configure your own XMPP server):

Table 26–14 Required Third-Party Software

Name	Instructions	Version(s)
Apache Smack	This driver uses the Apache Smack XMPP Java library to connect to a Jabber/XMPP IM Server. This driver includes a licensed copy of Smack (version 3.0.4).	3.0.4

Note: You are not required to install your own XMPP Server if you have access to an existing server. For a list of public servers, see `http://www.jabber.org`.

26.4.1.6.4 Driver Application Archive (EAR) The EAR file is `$ORACLE_HOME/communications/applications/sdpmessagingdriver-xmpp.ear`.

26.4.1.6.5 Common Properties Table 26–15 lists common driver properties that are indicative of the capabilities of this driver for use by the messaging engine when routing outbound messages. Some properties are set by the driver developer and do not normally require modification, while others can be modified by the administrator to change the routing behavior. Some properties such as SendingQueuesInfo are for advanced use and only require modification for advanced deployment topologies. For a complete description of these properties and available values, see the JavaDoc of `DriverConfigPropertyNames`.

Table 26–15 Common XMPP Properties

Name	Description	Mandatory	Default Value
InstanceName	Instance name (for internal use only)	Yes	XMPP-IM-Driver
Capability	Message sending and receiving capability	Yes	Both
SupportedDeliveryTypes	Supported delivery types	Yes	IM
SupportedContentTypes	Supported content types	Yes	text/plain
SupportedStatusTypes	Supported status types	No	DELIVERY_TO_GATEWAY_SUCCESS, DELIVERY_TO_GATEWAY_FAILURE
Cost	Cost	No	N/A
Speed	Speed	No	N/A
SupportedCarriers	Supported carriers	No	N/A
Supported Protocols	Supported protocols	No	N/A
SupportsCancel	Supports a cancel operation on the message	No	False
SupportsReplace	Supports a replace operation on the message	No	False

Table 26–15 (Cont.) Common XMPP Properties

Name	Description	Mandatory	Default Value
SupportsTracking	Supports a tracking operation on the message	No	False
SupportsStatusPolling	Supports a status polling operation on the message	No	False
SenderAddresses	Sender addresses	No	N/A
DefaultSenderAddress	Default sender address	No	N/A
SendingQueuesInfo	Driver sending queue information	Yes	OraSDPM/QueueConnectionFactory:OraSDPM/Queues/OraSDPMDriverDefSndQ1

26.4.1.6.6 XMPP Custom Properties The XMPP Driver includes the custom properties shown in Table 26–16.

Table 26–16 Custom XMPP Properties

Name	Description	Mandatory	Default Values
IMServerHost	Jabber/XMPP server hostname.	No	N/A
IMServerPort	Corresponding Jabber/XMPP server port. The default is *5222*.	Yes	5222
IMServerUsername	Jabber/XMPP username with which you log in. You may also enter a complete Jabber ID if its domain name is different from the Jabber/XMPP server hostname (for example: myUserName or myUserName@*xmpp-domain*). **Note:** An attempt is made to register this user account if it does not exist and the server supports account registration.	No	N/A
IMServerPassword	Corresponding password for the username listed above. Includes Type of Password (choose from Indirect Password/Create New User, Indirect Password/Use Existing User, Use Cleartext Password) and Password.	No	N/A
SecurityMode	Security mode to use when making a connection to the server. Available options are: None (Security is disabled and only unencrypted connections are used), TLS (Security through TLS encryption is used whenever it is available), and SSL (Security through SSL encryption is used). The default is TLS.	No	TLS
SASLAuthenticationEnabled	Whether to use SASL authentication when logging into the server. If SASL authentication fails, then the driver tries to use non-SASL authentication. By default, SASL is enabled.	No	Enabled

26.4.1.6.7 Client API MessageInfo Support These properties are message delivery-related that are specified through the client API. Table 26–17 describes if the protocol or driver implementation honors such properties.

Table 26–17 Client API MessageInfo Support

Name	Description	Support
Expiration	Expiration means how long the message may exist until it expires.	False
Delay	Delay means the amount of time that must elapse before the message is sent.	False

26.4.1.7 Configuring the VoiceXML Driver

The VoiceXML Driver supports the Genesys VoiceGenie gateway's outbound call protocol to send messages authored in VoiceXML. The gateway delivers the message using text-to-speech synthesis.

26.4.1.7.1 VoiceXML Driver Interoperability This section details interoperability features of the VoiceXML Driver.

The VoiceXML driver is compatible with this protocol: VoiceXML over HTTP (VoiceGenie gateway protocol).

The VoiceXML driver features include:

- VoiceXML content only

The gateway vendor and version in Table 26–18 has been verified.

Table 26–18 VoiceXML Driver Gateway Vendor and Version

Vendor	Version
Genesys VoiceGenie	6.4.2

26.4.1.7.2 Common Properties Table 26–19 lists common driver properties that are indicative of the capabilities of this driver for use by the messaging engine when routing outbound messages. Some properties are set by the driver developer and do not normally require modification, while others can be modified by the administrator to change the routing behavior. Some properties such as SendingQueuesInfo are for advanced use and only require modification for advanced deployment topologies. For a complete description of these properties and available values, see the JavaDoc of `DriverConfigPropertyNames`.

Table 26–19 Common VoiceXML Properties

Name	Description	Mandatory	Default Value
InstanceName	Instance name (for internal use only)	Yes	VoiceXML-Driver
Capability	Message sending and receiving capability	Yes	SEND
SupportedDeliveryTypes	Supported delivery types	Yes	VOICE
SupportedContentTypes	Supported content types	Yes	text/vxml, text/x-vxml

Table 26–19 (Cont.) Common VoiceXML Properties

Name	Description	Mandatory	Default Value
SupportedStatusTypes	Supported status types	No	DELIVERY_TO_ GATEWAY_ SUCCESS, DELIVERY_TO_ GATEWAY_FAILURE
Cost	Cost	No	N/A
Speed	Speed	No	N/A
SupportedCarriers	Supported carriers	No	N/A
Supported Protocols	Supported protocols	No	N/A
SupportsCancel	Supports cancel operation on the message	No	False
SupportsReplace	Supports replace operation on the message	No	False
SupportsTracking	Supports tracking operation on the message	No	False
SupportsStatusPolling	Supports status polling operation on the message	No	False
SenderAddresses	Sender Addresses	No	N/A
DefaultSenderAddress	Default Sender Address	No	N/A
SendingQueuesInfo	Driver Sending Queue Info	Yes	OraSDPM/QueueConnectionFactory:OraSDPM/Queues/OraSDPMDriverDefSndQ1

26.4.1.7.3 VoiceXML Custom Properties The VoiceXML Driver includes the custom properties shown in Table 26–20.

Table 26–20 Custom VoiceXML Properties

Name	Description	Mandatory	Default Values
VoiceXMLOutboundServlet URI	The URL of the VoiceXML gateway.	Yes	N/A
VoiceXMLOutboundServlet UserName	The user name of the VoiceXML gateway.	No	N/A
VoiceXMLOutboundServlet Password	The password of the user of the VoiceXML gateway.	No	N/A
VoiceXMLOutboundServlet DNIS	The number that should appear in the recipient's caller ID display.	No	N/A

Table 26–20 (Cont.) Custom VoiceXML Properties

Name	Description	Mandatory	Default Values
VoiceXMLReceiveURL	The URL of this driver's servlet that handles incoming requests from the VoiceXML Gateway. The format is `http://host:port/usermessagingdriver-voicexml/receive`. The default behavior, if this property is not set, is to use the local container's HTTP listener host and port. The default behavior only works for the first driver instance. For additional instances, the context root is different and this property must be configured using the correct context root replacement for `/sdpmessagingdriver-voicexml`.	No	N/A

Note: In a clustered (high-availability) environment with Oracle HTTP Server (OHS) configured, do not use the OHS port to configure the VoiceXML driver receive URLs. Using the OHS port to configure the VoiceXML driver receive URLs causes a conflict with the drivers.

Each VoiceXML driver must be configured with its own WLS server's port.

26.4.1.7.4 Client API MessageInfo Support These properties are message delivery related which are specified through client API. Table 26–21 describes if the protocol or driver implementation honors such properties.

Table 26–21 Client API MessageInfo Support

Name	Description	Support
Expiration	Expiration means how long the message may exist until it expires.	False
Delay	Delay means the amount of time that must elapse before the message is sent.	False

26.4.1.8 Configuring the Worklist Driver

The Worklist driver enables notifications from all sources to be sent to users in the form of worklist tasks for integration into the users' WebCenter Portal Unified Worklist.

Note: Worklist message tasks are accessible both through an Oracle WebCenter Portal that has been configured to search the BPEL connection to which the worklist message driver is sending messages, and through Oracle BPM Worklist. Oracle BPM Worklist also shows these message-based tasks as worklist items.

This integration is achieved by exposing a worklist channel (delivery type) to applications and end users. Messages sent through the user's worklist channel are processed by the worklist driver.

> **Note:** The worklist channel supports all rich text tags supported by the `af:richTextEditor` for the `text/html` content type. For more information about the Rich Text Editor, see *<af:richTextEditor>*.

The User Messaging Service API semantics are the same as those for existing channels such as IM or email. *This driver handles sending messages only.* The driver Application Archive (EAR) is located at: `$ORACLE_HOME/communications/applications/sdpmessagingdriver-worklist.ear`

You can install the Worklist Driver onto an Oracle WebLogic platform, or onto an IBM WebSphere platform. Choose the appropriate installation instructions below.

26.4.1.8.1 Install the Worklist Driver on the Oracle WebLogic Server Platform To enable the messaging worklist feature, the WebLogic SOA domain must be extended using the extension template available at `$ORACLE_HOME/common/templates/applications/oracle.ums.driver.worklist_template_11.1.1.jar`.

To extend a SOA domain using the Oracle Fusion Middleware Configuration Wizard:

1. Launch Oracle Fusion Middleware Configuration Wizard (`$ORACLE_HOME/common/bin/config.sh` or `%ORACLE_HOME%\common\bin\config.cmd`).
2. Select the *Extend an existing WebLogic domain* option.
3. Select the desired SOA domain directory.
4. Select the *Extend my domain using an existing extension template* option.
5. Click **Browse**, and navigate to `$ORACLE_HOME/common/templates/applications`
6. Select oracle.ums.driver.worklist_template_11.1.1.jar
7. Complete the remaining steps of the Oracle Fusion Middleware Configuration Wizard, and restart the SOA servers.

> **Note:** *Special Considerations if the SOA managed server is on a remote computer*: The `oracle.ums.driver.worklist_template_11.1.1.jar` extension template includes a SOA composite application (`sca_sdpmessagingsca-worklist-composite_rev1.0.jar`) that is copied to `$DOMAIN_HOME/soa/autodeploy`, and is auto-deployed by the SOA Infra runtime upon server restart. However, if the SOA Infra runtime is on a remote computer, and the domain is packed with the *-managed=true* option (the correct option to use), this directory is not included in the archive. Thus, the composite is not deployed upon restarting the SOA managed server.
>
> To complete the installation, copy the contents of `$DOMAIN_HOME/soa/autodeploy` from the AdminServer computer to the corresponding location on the remote computer with the SOA managed server, and restart the SOA managed server. You may have to create the directory structure `soa/autodeploy` under `$DOMAIN_HOME` on the remote computer.

26.4.1.8.2 Install the Worklist Driver on IBM WebSphere Platform To enable the messaging worklist feature, the WebSphere SOA cell must be extended using the extension template available at `$ORACLE_HOME/common/templates/was/oracle.ums.driver.worklist_template_11.1.1.jar`.

To extend a SOA cell using the Oracle Fusion Middleware Configuration Wizard:

1. Set the environment variable `CONFIG_JVM_ARGS` with the value `-DTemplateCatalog.enable.selectable.all=true`. For example, in Linux bash shell: `exportCONFIG_JVM_ARGS="-DTemplateCatalog.enable.selectable.all=true"`
   At the Windows command prompt: `set CONFIG_JVM_ARGS=-DTemplateCatalog.enable.selectable.all=true`
2. Launch Oracle Fusion Middleware Configuration Wizard (`$ORACLE_HOME/common/bin/was_config.sh` or `%ORACLE_HOME%\common\bin\was_config.cmd`).
3. Click **Select and Configure Existing Cell**.
4. Select the desired SOA cell and click **Next**.
5. Select **Oracle User Messaging Service Worklist Driver** on the *Add Products to Cell* screen and click **Next**.
6. Complete the remaining steps of the Oracle Fusion Middleware Configuration Wizard with default selections, and restart the SOA cell.

26.4.1.8.3 Common Properties The common driver properties shown in Table 26–22 are indicative of the capabilities of this driver for use by the messaging engine when routing outbound messages. Some properties are set by the driver developer and do not normally require modification, while others can be modified by the administrator to change the routing behavior. Some properties such as `SendingQueuesInfo` are for advanced use and only require modification for advanced deployment topologies. For a complete description of these properties and available values, see the JavaDoc of `DriverConfigPropertyNames`.

Table 26–22 Common Worklist Properties

Name	Description	Mandatory?	Default Value
InstanceName	Instance name (for internal use only)	Yes	Worklist-Driver
Capability	Message sending and receiving capability	Yes	SEND
SupportedDeliveryTypes	Supported delivery types	Yes	WORKLIST
SupportedContentTypes	Supported content types	Yes	text/plain, text/html
SupportedStatusTypes	Supported status types	No	DELIVERY_TO_GATEWAY_SUCCESS, DELIVERY_TO_GATEWAY_FAILURE
Cost	Cost	No	N/A

Table 26–22 (Cont.) Common Worklist Properties

Name	Description	Mandatory?	Default Value
Speed	Speed	No	N/A
SupportedCarriers	SupportedCarriers	No	N/A
SupportedProtocols	SupportedProtocols	No	N/A
SupportsCancel	Supports cancel operation on the message	No	False
SupportsReplace	Supports replace operation on the message	No	False
SupportsTracking	Supports tracking operation on the message	No	False
SupportsStatusPolling	Supports status polling operation on the message	No	False
SenderAddresses	Sender addresses	No	N/A
DefaultSenderAddress	Default sender address	No	N/A
SendingQueuesInfo	Driver sending queue information	Yes	OraSDPM/QueueConnectionFactory:OraSDPM/Queues/OraSDPMDriverDefSndQ1

26.4.1.8.4 Custom Properties The custom property shown in Table 26–23 is available.

Table 26–23 Custom Worklist Property

Name	Description	Mandatory	Default Value
BPELConnectionURL	The URL of the BPEL server to connect to. The format is `'http://<bpel-host>:<bpel-port>'`. The default behavior, if this property is not set, is to use the local container's HTTP connection URL.		

26.4.1.8.5 Client API MessageInfo Support Table 26–24 shows if the protocol or driver implementation honor the following message delivery-related properties that are specified through the client API.

Table 26–24 Client API MessageInfo Support

Name	Description	Support
Expiration	Expiration means how long the message may exist until it expires.	False
Delay	Delay means the amount of time that must elapse before the message is sent.	False

26.4.1.9 Configuring the Proxy Driver

The Proxy Driver acts as a Messaging Web Service client to a Fusion Middleware Messaging server hosted elsewhere in the intranet or Internet. It uses SOAP over

HTTP (the Parlay X Multimedia Web Service protocol) to send messages and receive messages and return message delivery status. The ParlayX Web Service relays messages from one UMS instance to another. It can relay traffic from multiple instances in an Intranet to a terminating instance that has all of the protocol-specific drivers configured to an external gateway such as an SMSC, or to an SMTP or IMAP mail server.

26.4.1.9.1 Common Properties Table 26–25 shows common driver properties that are indicative of the capabilities of this driver for use by the messaging engine when routing outbound messages. Some properties are set by the driver developer and do not normally require modification, while others can be modified by the administrator to change the routing behavior. Some properties such as SendingQueuesInfo are for advanced use and only require modification for advanced deployment topologies. For a complete description of these properties and available values, see the JavaDoc of `DriverConfigPropertyNames`.

Table 26–25 Common Proxy Properties

Name	Description	Mandatory	Default Value
InstanceName	Instance name (for internal use only)	Yes	Proxy-Driver
Capability	Message sending and receiving capability	Yes	SEND
SupportedDeliveryTypes	Supported delivery types	Yes	EMAIL, SMS, VOICE, IM, WORKLIST
SupportedContentTypes	Supported content types	Yes	*
SupportedStatusTypes	Supported status types	No	DELIVERY_TO_GATEWAY_SUCCESS, DELIVERY_TO_GATEWAY_FAILURE
Cost	Cost	No	N/A
Speed	Speed	No	N/A
SupportedCarriers	Supported carriers	No	N/A
Supported Protocols	Supported protocols	No	N/A
SupportsCancel	Supports cancel operation on the message	No	False
SupportsReplace	Supports replace operation on the message	No	False
SupportsTracking	Supports tracking operation on the message	No	False
SupportsStatusPolling	Supports status polling operation on the message	No	False
SenderAddresses	Sender addresses	No	N/A
DefaultSenderAddress	Default sender address	No	N/A
SendingQueuesInfo	Driver sending queue information	Yes	OraSDPM/QueueConnectionFactory:OraSDPM/Queues/OraSDPMDriverDefSndQ1

26.4.1.9.2 Proxy Custom Properties The Proxy Driver includes the custom properties shown in Table 26–26.

Table 26–26 Custom Proxy Properties

Name	Description	Mandatory	Default Values
GatewayURL	The URL to the hosted 11*g* UMS Web Service gateway. The URL is in the following format: `http://<host>:<port>/sdpmessaging/parlayx/SendMessageService`	Yes	N/A
Username	Username of the messaging gateway.	No	N/A
Password	The password of the username	No	N/A
Policies	Comma-delimited list of Oracle Web Services Manager WS-Security policies to be attached to proxy driver requests	No	N/A

26.4.1.9.3 Client API MessageInfo Support These properties are message delivery related which are specified through client API. Table 26–27 describes if the protocol or driver implementation honors such properties.

Table 26–27 Client API MessageInfo Support

Name	Description	Support
Expiration	*Expiration* means how long the message may exist until it expires.	False
Delay	*Delay* means the amount of time that must elapse before the message is sent.	False

26.5 Configuring User Messaging Service Access to the LDAP User Profile

As part of the LDAP provider setup in a SOA deployment, you configure the "User Name Attribute" through the WebLogic Server Administration Console. If you configure that attribute with a value other than the default "cn" or if the user's email address is stored in an LDAP attribute which is different from "mail", you must make an additional configuration change in Oracle Platform Security Services (OPSS) for UMS to successfully access the user profile to obtain the list of communication channels provisioned in LDAP, such as business email.

For more information about Oracle Platform Security Services (OPSS), see *Oracle Fusion Middleware Security Guide*.

To configure access to the LDAP user profile:

1. Modify the `jps-config.xml` file in the `$DOMAIN_HOME/config/fmwconfig` directory by adding a `<property>` element in the `idstore.ldap serviceInstance` section.

   - To use the value of the User Name Attribute while searching the back-end LDAP server for user profile, add the following element:

     ```
     <property name="username.attr" value="username_attribute_value"/>
     ```

     where *`username_attribute_value`* is the value of the User Name Attribute property in the LDAP provider configuration. For instance, if the value of the User Name Attribute is `mail`, add the following line:

     ```
     <property name="username.attr" value="mail"/>
     ```

The following sample code shows the above line inserted in the `jps-config.xml` file:

```
<!-- JPS WLS LDAP Identity Store Service Instance -->

<serviceInstance name="idstore.ldap" provider="idstore.ldap.provider">

  <property name="idstore.config.provider"
value="oracle.security.jps.wls.internal.idstore.WlsLdapIdStoreConfigProvide
r"/>

  <property name="CONNECTION_POOL_CLASS"
value="oracle.security.idm.providers.stdldap.JNDIPool"/>

  <property name=”username.attr” value=”mail”/>

</serviceInstance>
```

- If the LDAP attribute containing the user's business email addresses is something other the `mail` attribute, add the following element:

  ```
  <property name="PROPERTY_ATTRIBUTE_MAPPING" value="BUSINESS_EMAIL=attr_
  containing_email"/>
  ```

  where `attr_containing_email` is the attribute name in the LDAP provider that contains the user's email address. For instance, if the user attribute containing the email address is `externalEmail`, add the following line:

  ```
  <property name="PROPERTY_ATTRIBUTE_MAPPING" value="BUSINESS_
  EMAIL=externalEmail"/>
  ```

  The following sample code shows the above line inserted in the `jps-config.xml` file:

  ```
  <!-- JPS WLS LDAP Identity Store Service Instance -->

  <serviceInstance name="idstore.ldap" provider="idstore.ldap.provider">

    <property name="idstore.config.provider"
  value="oracle.security.jps.wls.internal.idstore.WlsLdapIdStoreConfigProvide
  r"/>

    <property name="CONNECTION_POOL_CLASS"
   value="oracle.security.idm.providers.stdldap.JNDIPool"/>

    <property name="PROPERTY_ATTRIBUTE_MAPPING" value="BUSINESS_
  EMAIL=externalEmail"/>

  </serviceInstance>
  ```

  Note: You may have other properties defined in the same section.

2. Restart your domain.

26.6 Securing the Oracle User Messaging Service

The User Messaging Preferences User Interface and the Parlay X Web Services can be secured at the transport-level using Secure Sockets Layer (SSL). By default, all deployed web services are unsecured. Web Service Security should be enabled for any services that are deployed in a production environment.

- To enable SSL in the Oracle WebLogic Server, see "Configure SSL for Oracle WebLogic Server" in the *Oracle Fusion Middleware Administrator's Guide*. This step is sufficient to secure the User Messaging Preferences User Interface.
- To secure the Parlay X Web Services, see "Configuring Transport-Level Security" in the *Securing WebLogic Web Services*.

UMS supports the use of Oracle Web Services Manager WS-Security policies to protect UMS web services. For more information about Oracle Web Services Manager, see "Using Oracle Web Service Security Policies", in *Oracle Fusion Middleware Securing WebLogic Web Services for Oracle WebLogic Server*.

The recommended security configuration for web services uses Security Assertion Markup Language (SAML) tokens to pass identities between web service clients and UMS. With SAML tokens, instead of the web service client passing a username and password to UMS, a trust relationship is established between the client and UMS because of exchanging certificates. Once this keystore configuration is in place, the web service client passes only the user identity, and vouches for the fact that it has authenticated the user appropriately.

The recommended policies to use for UMS web services are:

- `oracle/wss11_saml_token_with_message_protection_service_policy (server-side)`
- `oracle/wss11_saml_token_with_message_protection_client_policy (client-side)`
- `oracle/wss11_saml_token_identity_switch_with_message_protection_client_policy (client-side)`

Note: The choice of client-side policy depends on the security context in which your application is executing.

- If the thread that is making the web service call has the intended Subject associated with it (for example, from a web application that performs user authentication, or a Java EE module with a *run-as* identity defined), then use the policy `oracle/wss11_saml_token_with_message_protection_client_policy`.

  The current thread Subject is passed through using the SAML Policy WS-Security headers. In this case you should not specify the parameter `javax.xml.ws.BindingProvider.USERNAME_PROPERTY` when creating your web service client instance.

- If the thread that is making the web service call has an undefined Subject associated with it, or if you must programmatically supply a different identity, then use the policy `oracle/wss11_saml_token_identity_switch_with_message_protection_client_policy`, and specify the parameter `javax.xml.ws.BindingProvider.USERNAME_PROPERTY` when creating your web service client instance. If you want to perform dynamic identity switching, you must grant additional code permissions to your application. For more information, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

26.6.1 Web Service Security on Notification

The different web services include corresponding notification web services (`MessageNotification`, `PresenceNotification`) that run on the client side and receive notifications (message delivery status, message receipt, presence status change) when the appropriate event occurs.

This implementation does not provide for the use of Web Service security (WS-Security) by default during notification of the clients. That is, the server assumes that the notification web services running on the client side do not use WS-Security, and makes no attempt to authenticate itself when sending notifications. If you enable WS-Security on the client side, the notification from the server fails because the notification SOAP request is missing the required headers.

26.6.2 Enabling UMS Service Security

To enable a policy for a UMS web service, follow the steps in "Configuring Oracle WSM Security Policies in Administration Console" in *Oracle Fusion Middleware Securing WebLogic Web Services for Oracle WebLogic Server*, selecting policy `oracle/wss11_saml_token_with_message_protection_service_policy`. This configuration must be repeated for each service you want to secure.

26.6.3 Enabling Client Security

Web service client security must be enabled programmatically. When using the client libraries described in *Parlay X Messaging Client API and Client Proxy Packages* (in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*), WS-Security policy configuration is provided when a client object is constructed. The client constructors take an argument of type `Map<String, Object>`. In general when using SAML

authentication, the key/value pairs (Table 26–28) should be added to the configuration map in addition to other required properties such as the endpoint address.

Table 26–28 Client Security Keys

Key	Type	Typical Value
`oracle.sdp.parlayx.ParlayXConstants.POLICIES`	String[]	`oracle/wss11_saml_token_with_message_protection_client_policy`
`javax.xml.ws.BindingProvider.USERNAME_PROPERTY`	String	*<valid username>*
`oracle.wsm.security.util.SecurityConstants.Config.KEYSTORE_RECIPIENT_ALIAS_PROPERTY`	String	(optional) keystore alias for target service. See Client Aliases.

Example 26–1 Web Service Client Security

```
import oracle.sdp.parlayx.multimedia_messaging.send.SendMessageClient

...

Map<String, Object> config = new HashMap<String, Object>();
config.put(javax.xml.ws.BindingProvider.ENDPOINT_ADDRESS_PROPERTY, ums_url);
config.put(oracle.sdp.parlayx.ParlayXConstants.POLICIES, new String[]
 {"oracle/wss11_saml_token_with_message_protection_client_policy"});
config.put(javax.xml.ws.BindingProvider.USERNAME_PROPERTY, "test.user1");

SendMessageClient sendClient = new SendMessageClient(config);
```

26.6.4 Keystore Configuration

To use the recommended WS-Security policy, you must configure a keystore containing the public and private key information required by OWSM. Refer to "Configuring the Credential Store Using WLST" in *Oracle Fusion Middleware Securing WebLogic Web Services for Oracle WebLogic Server* for information on how to configure the keystore and corresponding credential store entries.

- If both your web service client and UMS server are in the same domain, then they share a keystore and credential store.
- If your web service client and UMS server are in different domains, then you must import the UMS public key into your client domain's keystore, and must import your client domain's public key into the UMS keystore.

26.6.5 Client Aliases

When using certain WS-Security policies such as the SAML policy recommended here, the client must use the server's public key to encrypt the web service request. However, there is generally only one keystore configured per domain. Therefore, if you have a domain in which there are web service clients that communicate with web services in multiple other domains, then you may be required to override the default keystore entry used by OWSM.

For example, if you have a domain in which application "A" is a web service client to a UMS web service, and application "B" is a web service client to a web service in another domain, then A's requests must be encrypted using the public key of the UMS domain, and B's requests must be encrypted using the public key of the other domain. You can accomplish this goal by overriding the keystore alias used by OWSM for each request:

- Import (for example) the UMS public key with alias "ums_public_key", and the other public key with alias "other_public_key".
- When creating an UMS web service client, specify the recipient keystore alias parameter, setting the key to `oracle.wsm.security.util.SecurityConstants.Config.KEYSTORE_RECIPIENT_ALIAS_PROPERTY` and the value to "ums_public_key" as shown in Example 26–2.

Example 26–2 Client Aliases

```
import oracle.sdp.parlayx.multimedia_messaging.send.SendMessageClient

...

Map<String, Object> config = new HashMap<String, Object>();
config.put(javax.xml.ws.BindingProvider.ENDPOINT_ADDRESS_PROPERTY, ums_url);
config.put(oracle.sdp.parlayx.ParlayXConstants.POLICIES, new String[]
{"oracle/wss11_saml_token_with_message_protection_client_policy"});
config.put(javax.xml.ws.BindingProvider.USERNAME_PROPERTY, "test.user1");
config.put(oracle.wsm.security.util.SecurityConstants.Config.KEYSTORE_RECIPIENT_
ALIAS_PROPERTY, "ums_public_key")
SendMessageClient sendClient = new SendMessageClient(config);
```

- The other web service client similarly must override the keystore alias, but the exact mechanism may differ. For example if using a JAX-WS client stub directly, then you can add the override property to the JAX-WS request context. See "Policy Configuration Overrides for the Web Service Client" in *Oracle Fusion Middleware Securing WebLogic Web Services for Oracle WebLogic Server* for more details.

26.6.6 Securing JMS Resources

This (optional) procedure enables administrators to restrict access to the Oracle User Messaging Service's JMS resources (such as queues) for enhanced security.

> **Note:** This section details steps to follow to secure JMS Resources. If you are starting with a new installation (11*g* Release 1 11.1.1.3) of Oracle User Messaging Service, then follow these steps.
>
> If you are not upgrading to 11*g* Release 1 11.1.1.3, then these instructions are not to be used.
>
> If you previously created a domain in an earlier release and are upgrading, but have not already done so, then complete the steps in the latest Release Notes before proceeding.

To secure the JMS system resources, lock all JMS sub-deployments that start with the name *UMSJMSSystemResource* (there may be multiple automatically-created resources for UMS in a multi-server or cluster deployment) with the role *OracleSystemRole*. Do this using the Oracle WebLogic Server Administration Console, or you may run a WLST script (available at `$ORACLE_HOME/communications/bin/secure_jms_system_resource.py`) as follows:

```
$ORACLE_HOME/common/bin/wlst.sh
$ORACLE_HOME/communications/bin/secure_jms_system_resource.py
--username=<admin_username> --password=<password>
--url=t3://<admin-host>:<admin-port> --jmsSystemResource
<UMSJMSSystemResource> --role OracleSystemRole
```

For example:

```
$ORACLE_HOME/common/bin/wlst.sh
$ORACLE_HOME/communications/bin/secure_jms_system_resource.py
--username=weblogic --password=<password> --url=t3://localhost:7001
--jmsSystemResource UMSJMSSystemResource --role OracleSystemRole
```

By default, the UMS system runs as the user *OracleSystemUser* for accessing JMS resources. If the user *OracleSystemUser* does not exist, or you secure the UMS JMS resources with any other role that some other user has been granted, you must override the default user identity used by the UMS system by specifying an alternate username for the following JVM system property when you start the container:

```
oracle.ums.system.user=<username>
```

For example, if the user is *MySystemUser*, you can pass the JVM system property on command line as: `-Doracle.ums.system.user=MySystemUser`

26.7 Troubleshooting Oracle User Messaging Service

To debug User Messaging Service, first check the server diagnostic logs. The logs may contain exception, error, or warning messages that provide details about incorrect behavior along with actions to remedy the problem. Table 26–29 describes additional methods for debugging common User Messaging Service problems.

Table 26–29 Troubleshooting UMS

Symptom	Possible Causes	Solutions
Notifications are not being sent from BPEL or Human Workflow components in SOA.	Notification Mode is set to NONE in SOA Workflow Notification configuration.	Change the Notification Mode setting to *EMAIL* or *ALL* using Oracle Fusion Middleware Control.
Email notification is not being sent.	The Outgoing (SMTP) Mail Server settings in the UMS Email Driver are incorrect.	Check the following settings in the UMS Email Driver using Oracle Fusion Middleware Control: ■ `OutgoingMailServer` ■ `OutgoingMailServerPort` **Note**: Validate the values by using them in any email client for connecting to the SMTP server.
	The SMTP server requires authentication or a secure connection (TLS or SSL).	Check the following settings in the UMS Email Driver using Oracle Fusion Middleware Control: ■ `OutgoingUsername` ■ `OutgoingPassword` ■ `OutgoingMailServerSecurity`

Table 26–29 (Cont.) Troubleshooting UMS

Symptom	Possible Causes	Solutions

Notifications are not being sent because of error message: `No matching drivers found for sender address = <address>`	The UMS Driver for the appropriate channel is configured with a specific list of *SenderAddresses*, and the message sent by the application has set a non-matching Sender Address.
	
	Note: UMS Server matches the outbound message's sender address, if set, against the available drivers' *SenderAddresses* to find a matching driver to use for delivering the message. If a driver has set one or more SenderAddresses, then the UMS Server only sends messages with the matching sender address to it.
		▪ Check the following settings in the appropriate UMS Driver using Oracle Fusion Middleware Control:
		`SenderAddresses`
		Note: The format for SenderAddresses is a comma-delimited list of `<DeliveryType>:<Address>`.
		For example:
		`EMAIL:sender@example.com, EMAIL:sender@example2.com`
		▪ Leave this property blank, if you want this driver to service outbound messages for all sender addresses for this channel (delivery type).
		▪ If there are multiple driver instances deployed for the same channel (delivery type) with different configurations, use the *SenderAddresses* to differentiate the driver instances. For example, one instance can be set with a value in SenderAddresses to only service outbound messages with that matching sender address, while the other instance can keep the SenderAddresses blank to service all outbound messages that do not specify any sender address or one that does not match that of the first driver instance.
		▪ SenderAddresses that are configured with the incorrect syntax (such as missing `<DeliveryType>:`) are ignored by the UMS Server for driver selection.

The email client inconsistently receives notifications.	The Incoming Mail Server settings in the UMS Email Driver are configured with the same email account to which notifications are being sent. If the notification is sent to the same account, the UMS Email Driver may download and process the email before the email client can display it.	Use an exclusive email account for Incoming Mail Server settings. Check the following settings in the UMS Email Driver using Oracle Fusion Middleware Control: ▪ `IncomingMailIDs` ▪ `IncomingUserIDs`

Table 26–29 (Cont.) Troubleshooting UMS

Symptom	Possible Causes	Solutions
SOA Human Workflow notifications are sent, but are not actionable.	The Actionable Email Address is not configured in SOA Workflow Notification Properties.	Set the Actionable Email Address in SOA Workflow Notification Properties with the address of the email account configured in the UMS Email Driver.
	The Human Workflow task is not set to send actionable notifications.	Set the *actionable* attribute for the Human Workflow task in JDeveloper and redeploy the SOA composite application.
SOA Human Workflow actionable notifications are sent, but no action is taken after responding.	The Incoming Mail Server settings in the UMS Email Driver are incorrect.	Check the following settings in the UMS Email Driver using Oracle Fusion Middleware Control: ■ `MailAccessProtocol` (*IMAP* or *POP3*, in uppercase) ■ `ReceiveFolder` ■ `IncomingMailServer` ■ `IncomingMailServerPort` ■ `IncomingMailServerSSL` ■ `IncomingMailServerSSL` ■ `IncomingUserIDs` ■ `IncomingUserPasswords` ■ `ImapAuthPlainDisable` **Note**: Validate the values by using them in any email client for connecting to an IMAP or POP3 server.
	The mail access protocol is incorrect.	Check the following settings in the UMS Email Driver using Oracle Fusion Middleware Control: ■ `MailAccessProtocol` (*IMAP* or *POP3*, in uppercase)
	The email server is SSL-enabled.	Check the following settings in the UMS Email Driver using Oracle Fusion Middleware Control: ■ `IncomingMailServerSS`
	The receive folder name is incorrect.	Check the following settings in the UMS Email Driver using Oracle Fusion Middleware Control: ■ `ReceiveFolder` **Note**: Some email servers may expect the value INBOX to be inbox or Inbox (that is, case-sensitive). Based on your email server, use an appropriate value.
	A nondefault email client is configured for receiving notifications. When the user clicks the approval link, the default mail client page opens, which may send emails to a different email server.	Configure the default email client to receive actionable notifications.

Table 26–29 (Cont.) Troubleshooting UMS

Symptom	Possible Causes	Solutions
SOA BPEL User Notification or Human Workflow notifications are sent to the correct delivery type (email, sms, and so on) but to the wrong address.	A self-provisioned messaging channel was created by the user in User Messaging Preferences for use in BPEL User Notification or Human Workflow use cases. **Note**: The User Messaging Preferences UI allows the end user to create his or her own messaging channel for various use cases, but these are not to be used for BPEL User Notification and Human Workflow.	Do not use a self-provisioned messaging channel for BPEL User Notification or Human Workflow use cases (that is, do not set as Default channel, and do not use in a messaging filter for such use cases). BPEL User Notification and Human Workflow use User Messaging Preferences only for the delivery type preference, and the actual address is retrieved from the user profile in the identity management system. **Note**: Addresses from the user profile in the identity management system are available through User Messaging Preferences using predefined channel names, such as *Business Email*, *Business Mobile*, *Business Phone*, *Instant Messaging*. Use these predefined messaging channels instead for BPEL User Notification and Human Workflow use cases.

27

Monitoring Oracle User Messaging Service

This chapter describes how to monitor Oracle User Messaging Service by using Oracle Enterprise Manager Fusion Middleware Control.

This chapter includes the following topics:

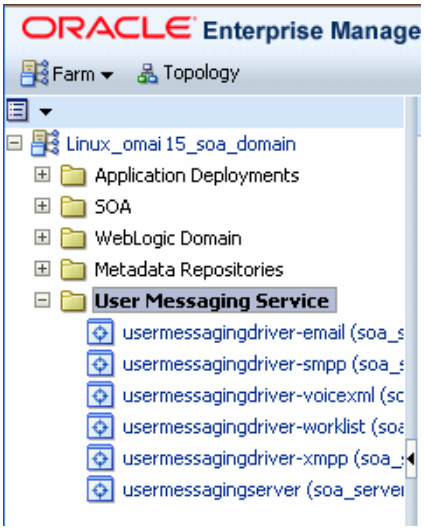
- Section 27.1, "Monitoring Oracle User Messaging Service"
- Section 27.2, "Viewing Log Files"
- Section 27.3, "Viewing Metrics and Statistics"

27.1 Monitoring Oracle User Messaging Service

You can monitor Oracle User Messaging Service logs and metrics using Oracle Enterprise Manager Fusion Middleware Control.

To monitor Oracle User Messaging Service:

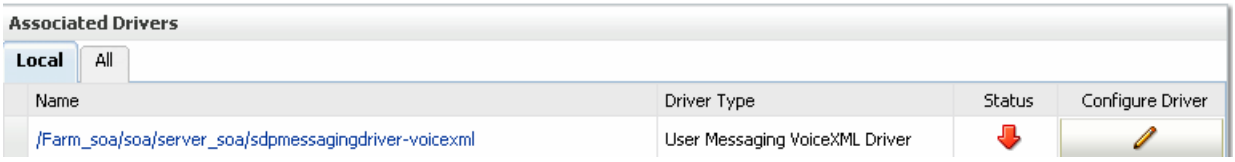
1. Go to the Oracle Enterprise Manager Fusion Middleware Control page for your SOA farm.



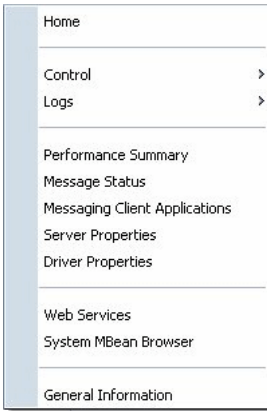
2. Select **Fusion Middleware** > **SOA** > **User Messaging Service**.
3. Select the server or driver of your choice.

   If you select a driver, quick statistics are displayed that indicate the state and performance of the driver.

If you select a server, you see a list of associated drivers, in addition to the quick statistics. You can select one of the drivers to view its statistics, or you can click the Configure Driver icon to configure it. For more information on configuring drivers, see Chapter 26, "Configuring Oracle User Messaging Service."

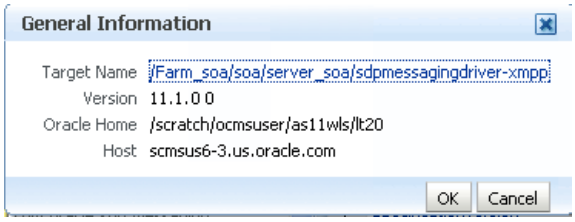


4. Right-click the server to select an action.



Selection	Action
Home	The home page lists the quick statistics for the selected driver.
Control	Start Up or Shut Down driver.
Logs	View and configure message logs for the selected driver.
Performance Summary	Displays performance statistics on a customizable metrics page. Use this page to view statistics for this driver. Customize this page using the Metric Palette. The Metric Palette enables you to choose from all of the available metrics so that you see only the information that is most valuable to you.
Message Status	Check the delivery status of messages sent and received, and resend selected messages. You can filter the search by adding more search fields and setting the desired operator and search value. Some fields can be added multiple times to use them with different and complementary operators, or with the *Contains* operator.
Messaging Client Applications	Messaging client applications registered with the User Messaging Service can be manually deregistered in cases where the applications have been undeployed and are holding onto access points that must be made available to other applications.
Server Properties	Configure message storage method and business terms for message filter creation. See Chapter 26, "Configuring Oracle User Messaging Service" for more information.
System MBean Browser	System MBean Browser and its configuration settings.

Selection	Action
General Information	General Information displays the name, version, Oracle home, and host for the selected driver.

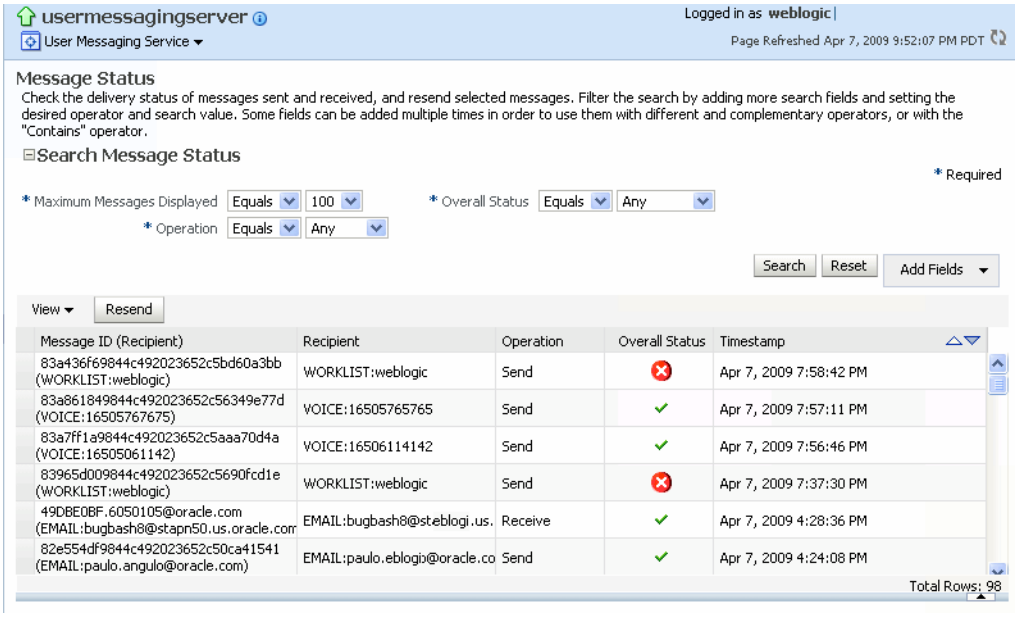


27.1.1 Using Message Status

You can check the delivery status of messages sent and received, and resend selected messages.

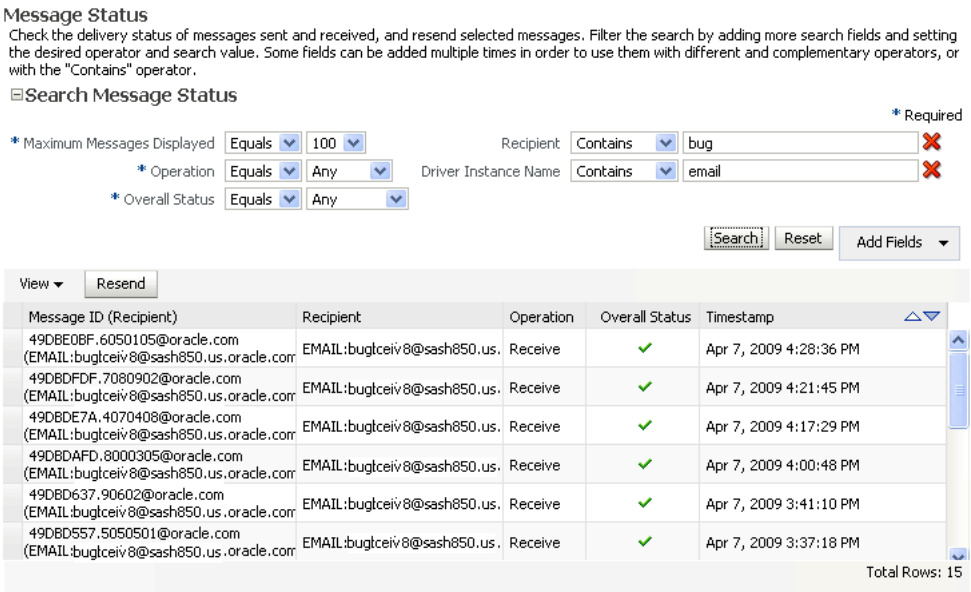
To check message status:

1. In the navigation tree, right-click the UMS target for which you want to view message status.
2. Select **Message Status**. The *Message Status* page appears.
3. Click **Search** to search the messages using the default criteria. The search returns a listing for the messages.



You can customize the search by adding more search fields and setting the desired operator and search value. Some fields can be added multiple times to use them with different and complementary operators, or with the Contains operator. To customize the search:

1. Click **Add Fields**.
2. Select the field(s) on which you want to search.
3. Choose operators and fill in variables as needed.
4. Click **Search**. The customized search is done and results returned.

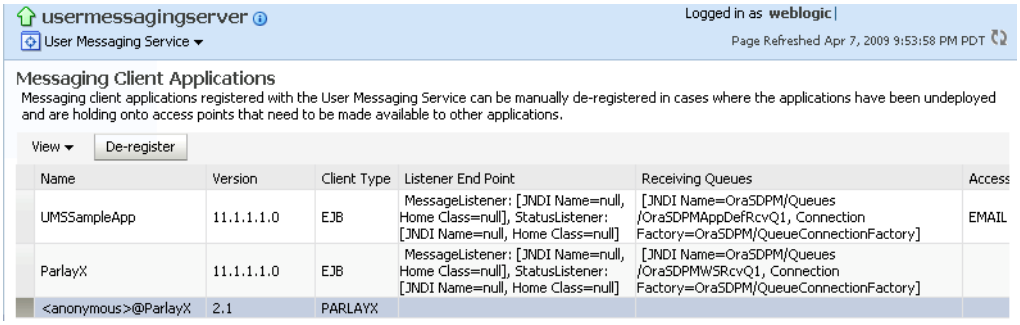


5. If you want to resend a message, select the message in the list and click **Resend**.

27.1.2 Deregistering Messaging Client Applications

You can manually deregister Messaging Client Applications after the applications have been undeployed and are holding onto access points that must be made available to other applications. To deregister Messaging Client Applications:

1. Right-click a target in the navigation tree, and select **Messaging Client**. The Messaging Client page appears.
2. Select the message to deregister.
3. Click **De-register**.



A confirmation box appears asking you to confirm your choice.

4. Confirm your choice.

27.1.3 Monitoring Drivers Using the All Tab

The **All** tab only lists successfully-registered drivers in the domain (not all drivers that exist).

Since the drivers are not configured out-of-the-box, they are not registered unless you configure them. To ensure that you see all of the drivers in the **All** tab, configure the

SMPP, VoiceXML and XMPP drivers (if you plan to use them). Once configured, they are registered with the engine and are displayed in the **All** tab.

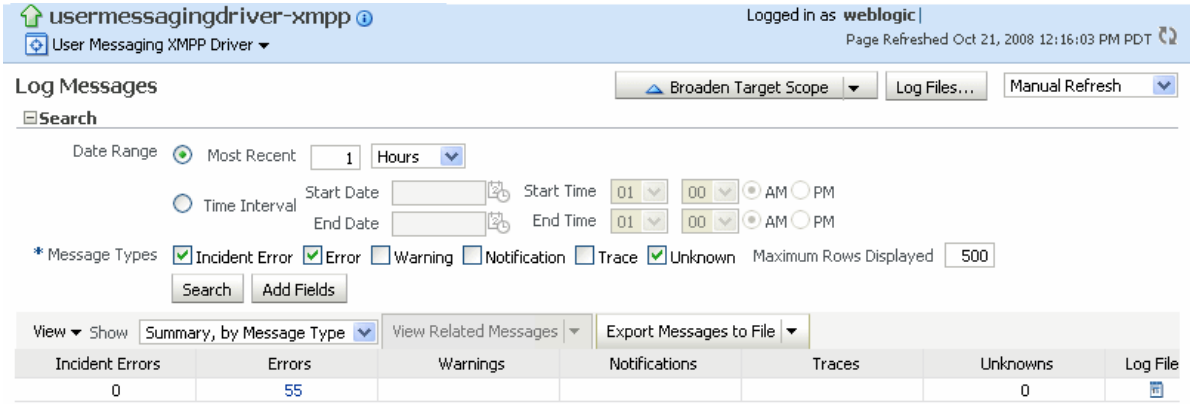
27.2 Viewing Log Files

You can view log files.

To view log files:

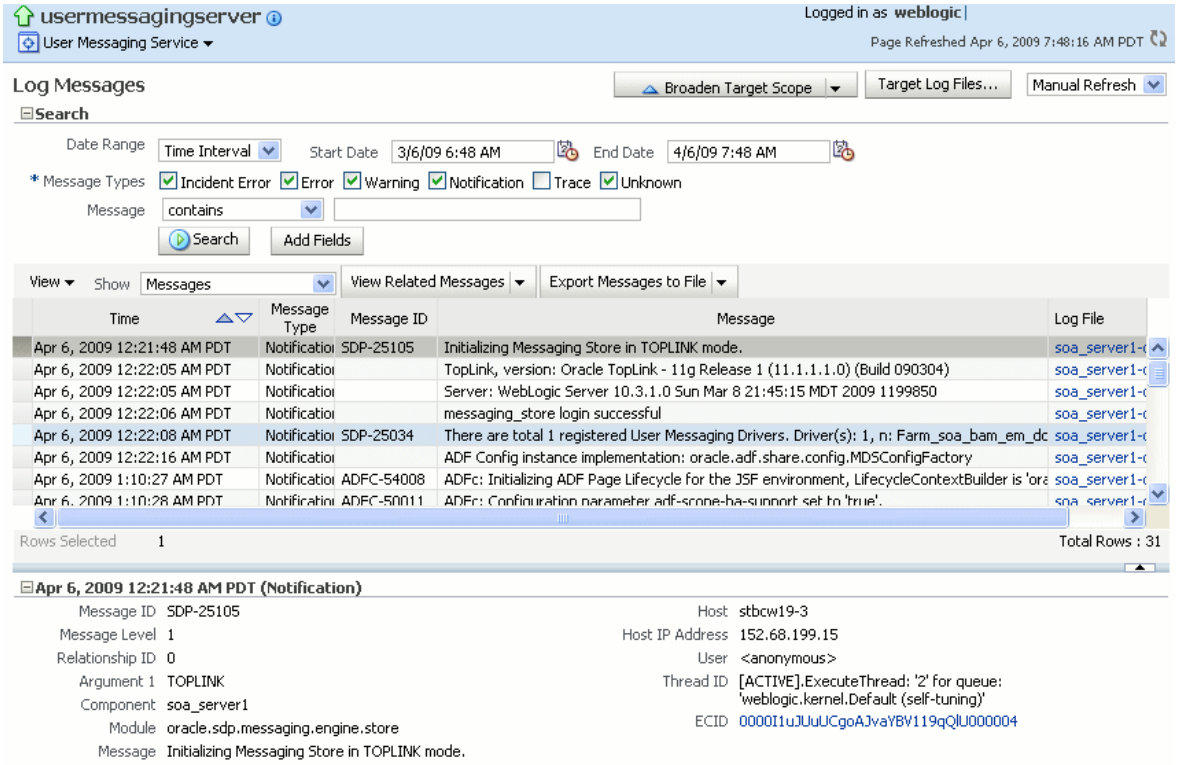
1. Right-click the driver (or server) for which you want to view log information, then choose **Logs** > **View Log Files**.

   The Log Messages page appears.



   Use this page to query for log information about a driver (or server). Fields and lists are used to customize the query.

2. After entering your search criteria, click **Log Files**. The Log Files page appears.

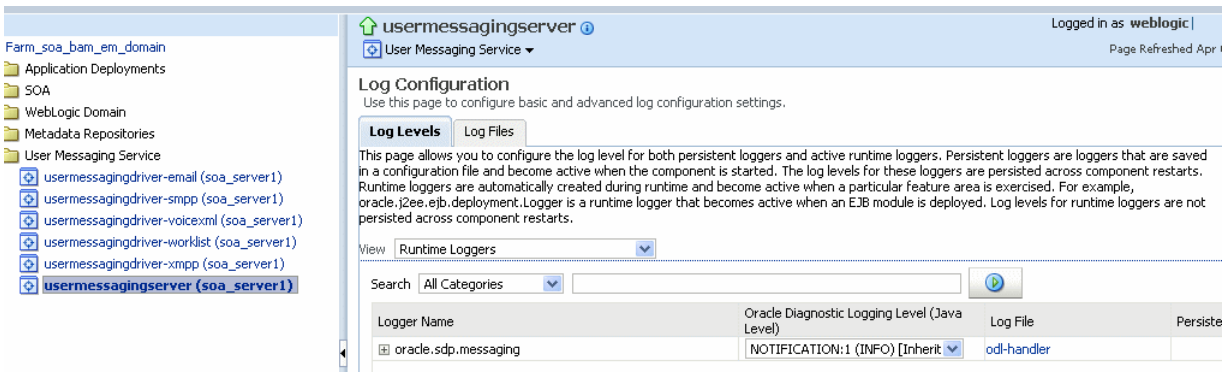


3. View log information or download the log.

27.2.1 Configuring Logging

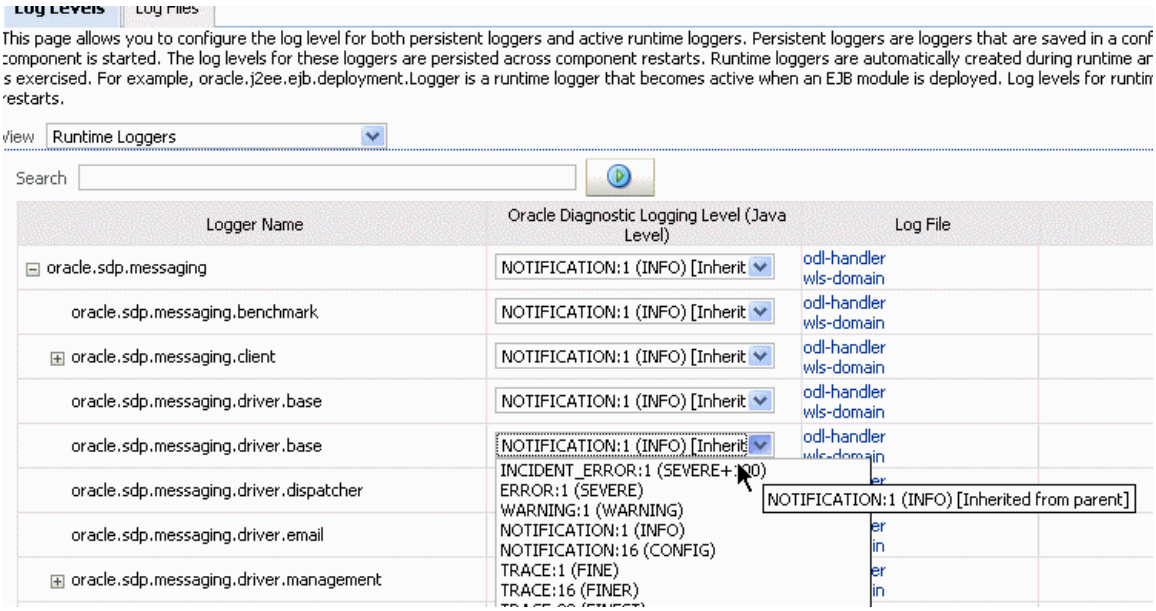
Use Oracle Enterprise Manager Fusion Middleware Control to configure log levels, as shown in Figure 27–1.

Figure 27–1 Configuring Log Levels



For each logger, set the notification level, as shown in Figure 27–2.

Figure 27–2 Select Notification Level



As a result of your configuration actions, notifications appear according to your specification. Figure 27–3 and Figure 27–4 provides details.

Figure 27–3 Viewing Log Files

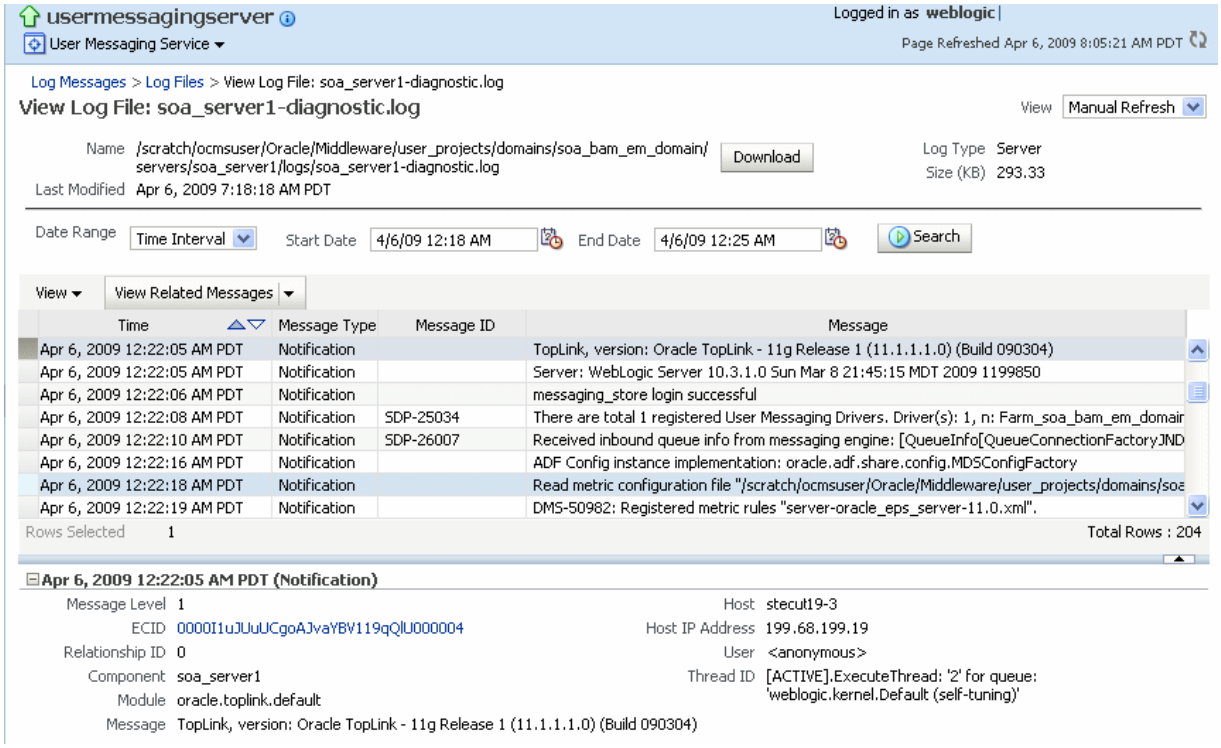
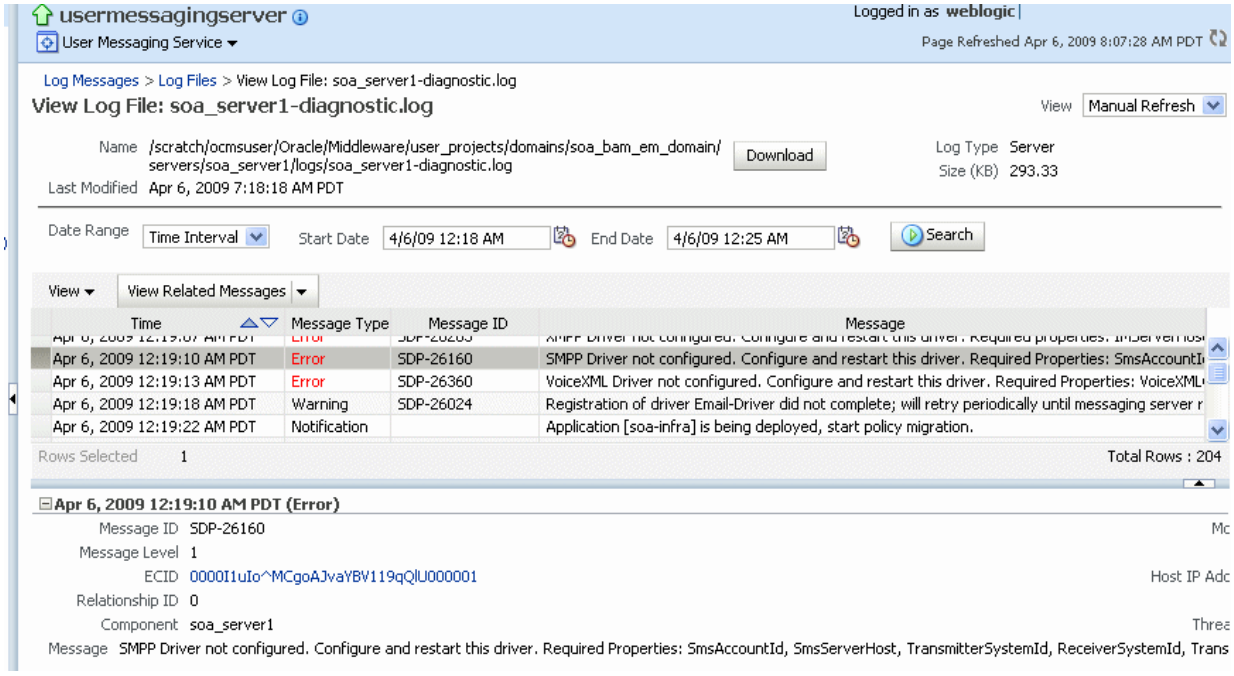


Figure 27–4 Error Messages



27.3 Viewing Metrics and Statistics

The performance of your applications is reflected in metrics and statistics.

To view metrics and statistics:

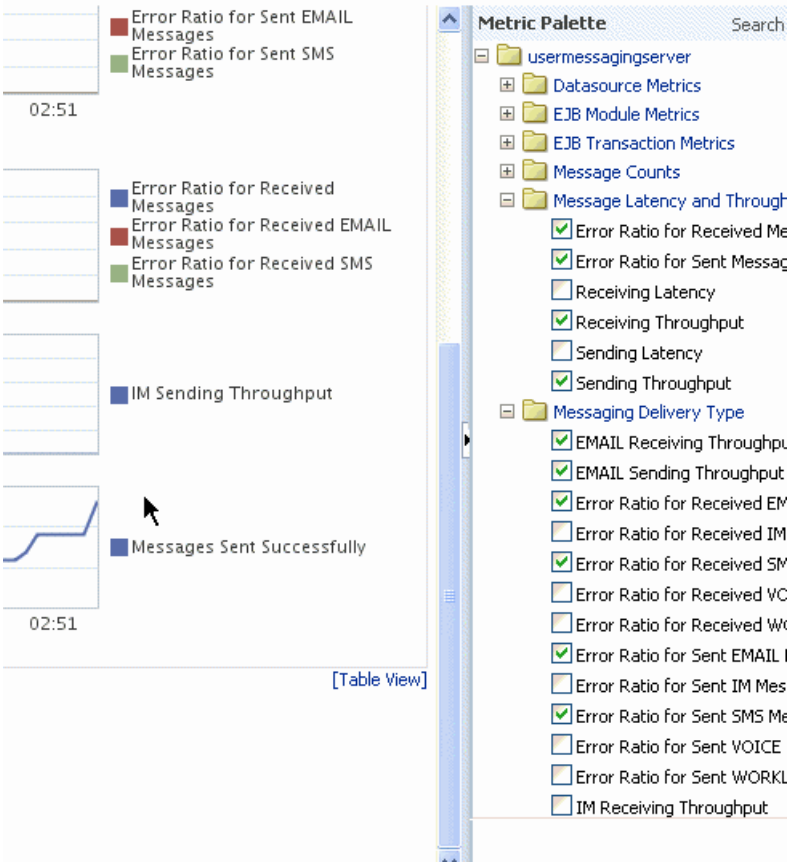
1. Select the Performance Summary for a driver (or server).

   The Performance Summary page appears.

   Many metrics are available for capture and display. To get the most valuable, focused information, use Metric Palette.

2. Click **Show Metric Palette** to display the Metric Palette.

3. Choose the metrics in which you are most interested. As you select or deselect metrics from the palette, the metrics display is automatically updated.



28

Managing Oracle User Messaging Service

This chapter describes how to deploy Oracle User Messaging Service (UMS) drivers by using Oracle WebLogic Scripting Tool (WLST), Oracle Enterprise Manager Fusion Middleware Control, Oracle Fusion Middleware Configuration Wizard, and the `wsadmin` utility of IBM WebSphere Application Server. It also describes how to undeploy and register Oracle UMS drivers.

This chapter includes the following topics:

- Section 28.1, "Deploying Drivers"
- Section 28.2, "Undeploying and Unregistering Drivers"

28.1 Deploying Drivers

When you install Oracle UMS, preinstalled drivers are included (Email, XMPP, SMPP, and VoiceXML). Among these drivers, only the Email driver is deployed to the WebLogic Server. To deploy other drivers, target that driver to the WebLogic Server (using Oracle WebLogic Server Administration Console, or you can target the drivers when creating or extending the domain using the Oracle Fusion Middleware Configuration Wizard).

The Worklist driver must be deployed to a SOA Server to make use of the UMS integration with Worklist. Because this integration involves multiple JEE applications and a SOA composite, there is a special extension template you must use to enable this feature in one step. See Install the Worklist Driver on the Oracle WebLogic Server Platform and Install the Worklist Driver on IBM WebSphere Platform for more information.

You can deploy additional drivers in a variety of ways using: WLST commands (recommended), wsadmin tool, Oracle Enterprise Manager Fusion Middleware Control, and through the Oracle Fusion Middleware Configuration Wizard.

The section includes the following topics:

- Section 28.1.1, "Deploying Drivers Using WLST Commands"
- Section 28.1.2, "Deploying Drivers Using Oracle Enterprise Manager Fusion Middleware Control"
- Section 28.1.3, "Deploying Drivers Using the Oracle Fusion Middleware Configuration Wizard"
- Section 28.1.4, "Deploying Drivers Using the wsadmin Tool"

Note: To deploy two or more driver instances of a particular driver EAR, you must use the custom deployment plan templates available at `$ORACLE_HOME/communications/plans`. Deploying drivers using WLST commands is recommended because these commands automatically modify your deployment plans for you; using other methods requires you to manually modify your deployment plans.

28.1.1 Deploying Drivers Using WLST Commands

You can deploy drivers using the WLST command `deployUserMessagingDriver`.

28.1.1.1 deployUserMessagingDriver

Command Category: UMS

Use with WLST: Online

28.1.1.1.1 Description `deployUserMessagingDriver` is used to deploy additional instances of user messaging drivers.

Specify a base driver type (for example: email, xmpp, voicexml, and others) and a short name for the new driver deployment. The string *usermessagingdriver-* is prepended to the specified application name. Any valid parameters for the *deploy* command can be specified, and is passed through when the driver is deployed.

28.1.1.1.2 Syntax `deployUserMessagingDriver(baseDriver, appName, [targets], [stageMode], [options])`

Argument	Definition
`baseDriver`	Specifies the base messaging driver type. Must be a known driver type, such as email, proxy, smpp, voicexml, or xmpp.
`appName`	A short descriptive name for the new deployment. The specified value is prepended with the string `usermessagingdriver-`.

`targets`
`stageMode`
`options`
	Optional. Additional arguments that are valid for the `deploy` command can be specified and passed through when the new driver is deployed.

28.1.1.1.3 Examples To deploy a second instance of an email driver with name `myEmail`.

```
wls:/base_domain/serverConfig> deployUserMessagingDriver(baseDriver='email',
appName='myEmail')
```

To deploy a second instance of an email driver, specifying deployment targets.

```
wls:/base_domain/serverConfig> deployUserMessagingDriver(baseDriver='email',
appName='email2', targets='server1,server2')
```

28.1.2 Deploying Drivers Using Oracle Enterprise Manager Fusion Middleware Control

To deploy drivers using Oracle Enterprise Manager Fusion Middleware Control:

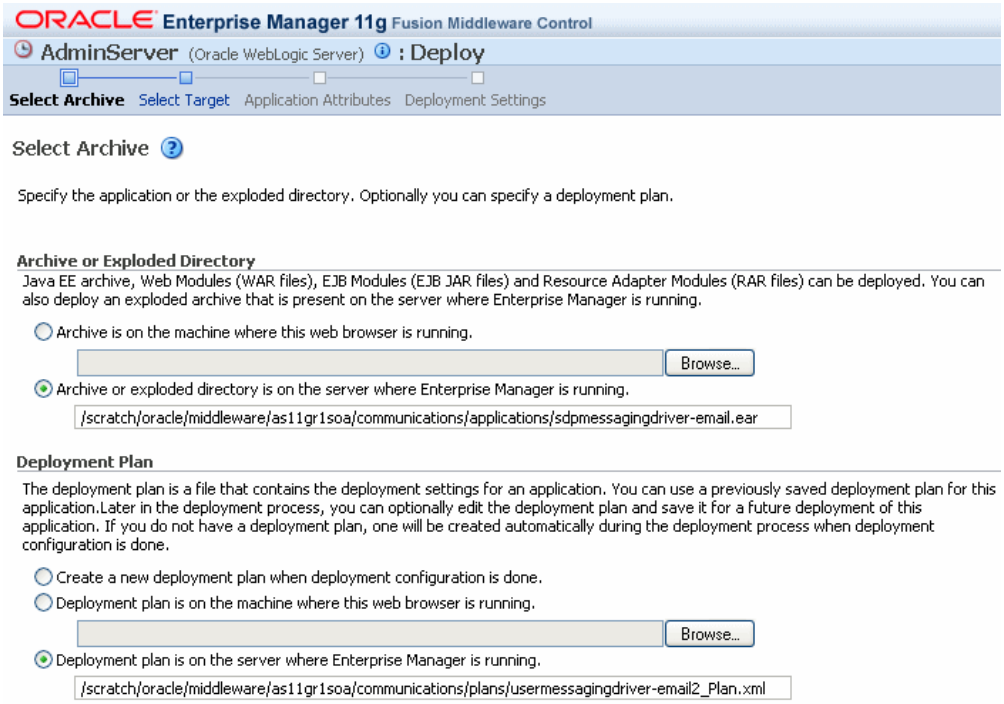
1. Retrieve a deployment template (for example: `ORACLE_HOME/communications/plans`)

2. Copy the plan to a location of your choice (to the same directory or any other directory).
3. Edit the plan:

   Replace `DriverDeploymentName` with whichever name you want to use (ensure you replace all instances of the name).

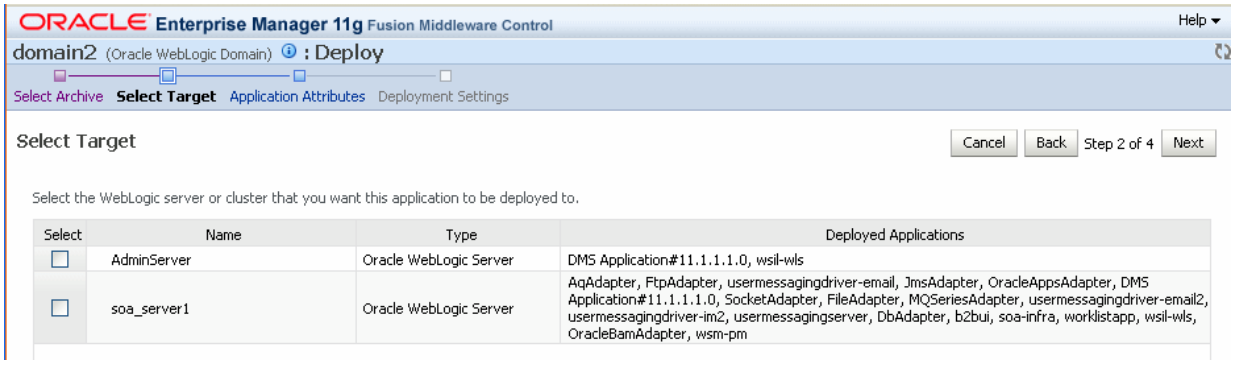
   Replace `DriverShortName` with any name you like.

   Replace the `@RunAsPrincipalName@` token with a valid principal for use by UMS. In a default deployment, the system principal `OracleSystemUser` is available for this purpose.
4. Start Oracle Enterprise Manager Fusion Middleware Control.
5. Enter the location of the `.ear` file.

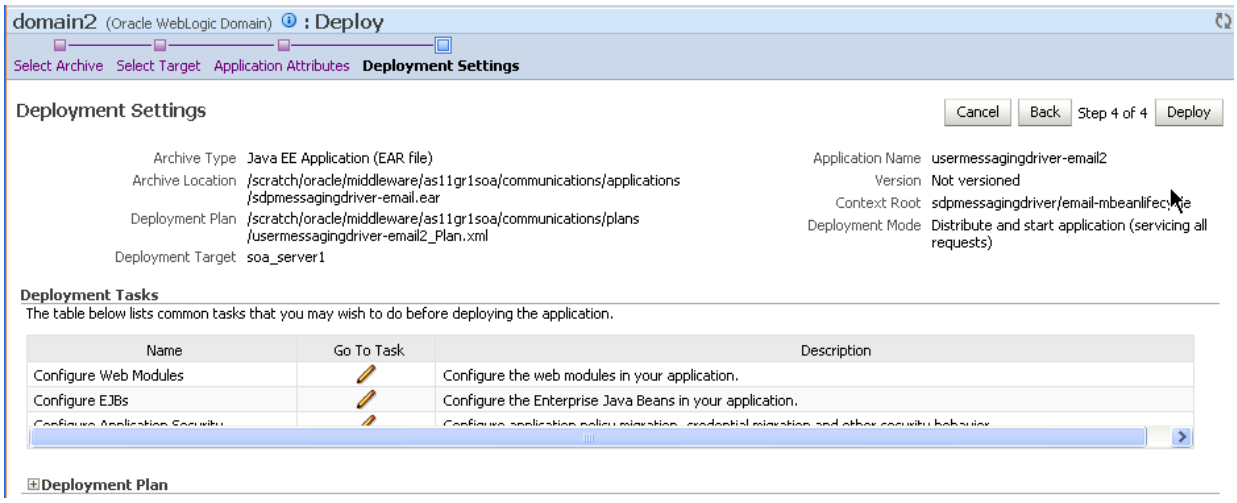


6. Click **Next**.

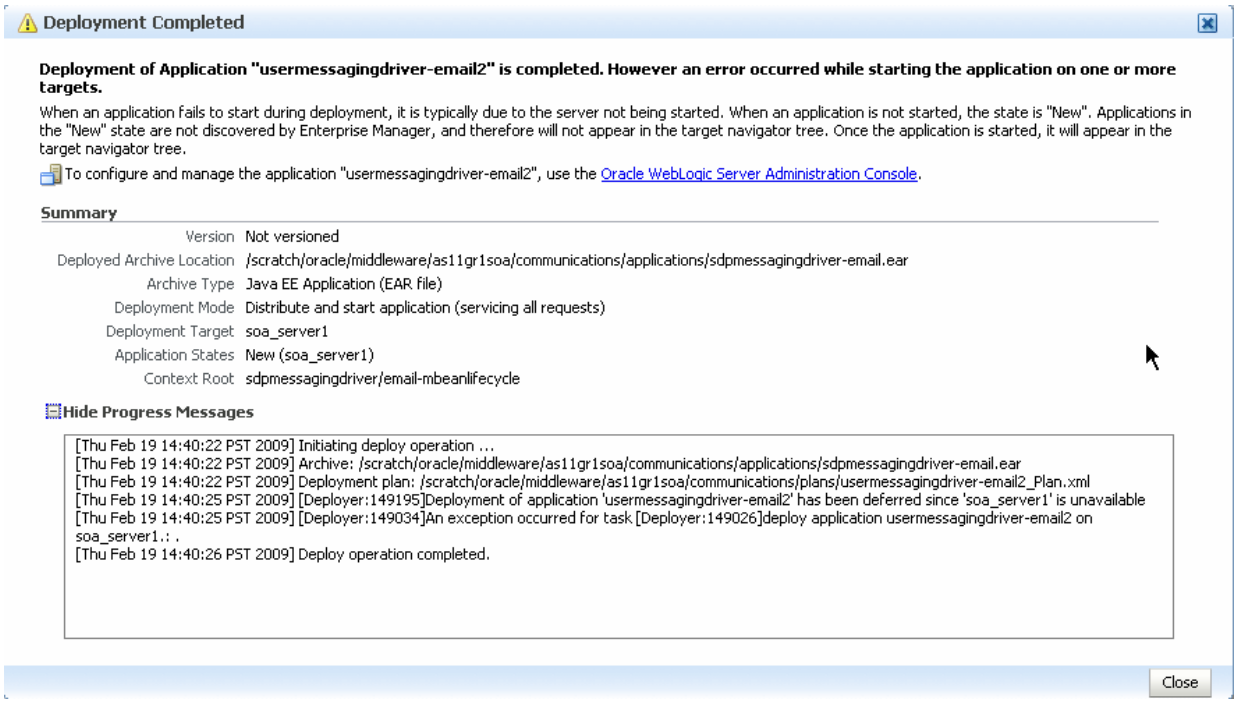
   The Select Target page appears.
7. Enter the location of the deployment plan.



8. Select the SOA target.
9. Enter an application name in the Application Attributes page. The application name must exactly match the string used for DriverDeploymentName (in Step 3 above) which is provided in the deployment plan. If it does not, the deployment and activation fails. The Deployment Setting page appears.



10. Click **Deploy**. The Deployment Completed page appears.



11. To see the result (driver deployed), start the SOA Server.

28.1.3 Deploying Drivers Using the Oracle Fusion Middleware Configuration Wizard

To install the SMPP, XMPP, and VoiceXML drivers, extend the domain using the extension template available at `$ORACLE_HOME/common/templates/applications/oracle.ums.drivers_template_11.1.1.jar`.

To extend a domain using Oracle Fusion Middleware Configuration Wizard:

1. Launch Oracle Fusion Middleware Configuration Wizard (`$ORACLE_HOME/common/bin/config.sh` or `%ORACLE_HOME%\common\bin\config.cmd`).
2. Select the **Extend an existing WebLogic domain** option.
3. Select the desired domain directory containing UMS.
4. Select the **Extend my domain using an existing extension template** option.
5. Click **Browse**, and navigate to `$ORACLE_HOME/common/templates/applications`
6. Select `oracle.ums.drivers_template_11.1.1.jar`.
7. Complete the remaining steps of the Oracle Fusion Middleware Configuration Wizard, and remember to target the required drivers to the desired Oracle WebLogic Servers, clusters, or both.
8. Restart the appropriate Oracle WebLogic Servers.

28.1.4 Deploying Drivers Using the wsadmin Tool

You can deploy additional instances of UMS drivers by using the Oracle Fusion Middleware-specific `wsadmin` command `OracleUMS.deployUserMessagingDriver`.

28.1.4.1 About the wsadmin OracleUMS.deployUserMessagingDriver Command

This section describes the syntax and arguments for the wsadmin `OracleUMS.deployUserMessagingDriver` command.

28.1.4.1.1 Description

`OracleUMS.deployUserMessagingDriver` is used to deploy additional instances of user messaging drivers.

Specify a base driver type (email, xmpp, smpp, voicexml, proxy, or worklist) and a short name for the new driver deployment. The string "usermessagingdriver-" is prepended to the specified application name.

28.1.4.1.2 Syntax

```
OracleUMS.deployUserMessagingDriver(baseDriver, appName, required_options)
```

Argument	Definition
`baseDriver`	Specifies the base messaging driver type. Must be a known driver type, such as email, proxy, smpp, voicexml, 'proxy', or 'worklist'.
`appName`	A short descriptive name for the new deployment. The specified value is prepended with the string `usermessagingdriver-`.
`required_options`	Additional arguments that are valid for the "AdminApp.install" command. These options are mandatory for this command: `-node`, `-cell`, `-server`, `-usedefaultbindings`, and `-defaultbinding.virtual.host`

Example:

```
OracleUMS.deployUserMessagingDriver('email', 'myEmail', '-node myNode -cell myCell
 -server myServer -usedefaultbindings -defaultbinding.virtual.host default_host')
```

where:

email and *myEmail* represent values for the `baseDriver` and `appName` arguments respectively.

-node, **-cell**, **-server**, **-usedefaultbindings**, **-defaultbinding.virtual.host** are the required options. *myNode*, *myCell*, *myServer* and *default_host* represent the values of respective options.

28.1.4.1.3 Examples

To deploy a second instance of an email driver with name "smpp2" on a SOA Server named "soa_server1":

```
wsadmin> OracleUMS.deployUserMessagingDriver('smpp','smpp2','-node DefaultNode01
 -cell DefaultCell01 -server soa_server1 -usedefaultbindings
 -defaultbinding.virtual.host default_host')
```

To deploy a second instance of an email driver with name "email2" on a BAM Server named "bam_server1":

```
wsadmin> OracleUMS.deployUserMessagingDriver('email','email2','-node DefaultNode01
 -cell DefaultCell01 -server bam_server1 -usedefaultbindings
 -defaultbinding.virtual.host default_host')
```

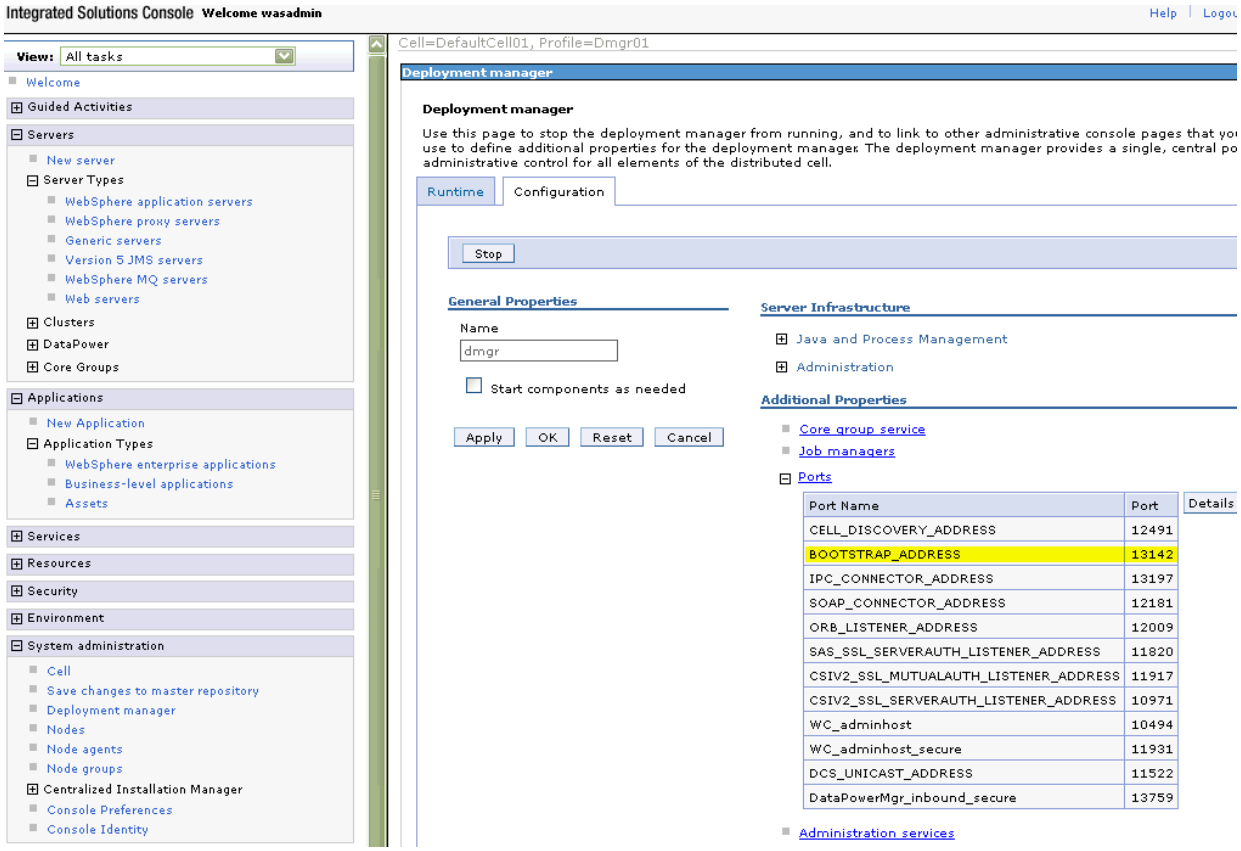
28.1.4.2 Deploying an Additional Driver

To deploy the UMS drivers by using the `wsadmin` tool, do the following:

1. Open a browser and enter the following URL to access the IBM WebSphere Administrative Console (also known as IBM WebSphere Integrated Solutions Console):

   ```
   https://hostname:SSL Port/ibm/console
   ```

2. Log in to the administration console by using the administrator name and password.
3. Navigate to **System administration** > **Deployment manager** > **Configuration** > **Ports** > **BOOTSTRAP_ADDRESS** and identify the corresponding port number.



4. On the system where Oracle SOA server is running, set the `ORACLE_HOME` environment variable to the location where Oracle Fusion Middleware SOA component is installed. You should set this variable before running the `wsadmin` command.

   For example,

   On UNIX platform, run the following command in the UNIX shell:

   ```
   $ export ORACLE_HOME=/fmwwas-nd/middleware/Oracle_SOA1
   ```

   On Windows platform, run the following command in the DOS shell:

   ```
   C:\> set ORACLE_HOME=C:\fmwwas-nd\middleware\Oracle_SOA1
   ```

5. From the *ORACLE_HOME*`\common\bin` folder, run the `wsadmin` command. Specify the RMI port number (see Step 2) as the value for the `-port` option.

   On UNIX:

   ```
   ORACLE_HOME/common/bin/wsadmin.sh
         -profileName profilename
         -connType RMI
         -host hostname
         -port RMI_port
         -user admin_user
         -password admin_password
   ```

   On Windows:

   ```
   ORACLE_HOME\common\bin\wsadmin.cmd
         -profileName profilename
   ```

```
-connType RMI
-host hostname
-port RMI_port
-user admin_user
-password admin_password
```

6. To view the list of enterprise applications and drivers that are already deployed, run the `wsadmin` command as follows:

```
wsadmin>print AdminApp.list()
    DMS Application_11.1.1.1.0
    DefaultToDoTaskFlow
    Dmgr DMS Application_11.1.1.1.0
    FMW Welcome Page Application_11.1.0.0.0
    User Messaging Parlay X Sample App
    composer
    em
    oracle-bam
    soa-infra
    usermessagingdriver-email
    usermessagingdriver-smpp
    usermessagingdriver-voicexml
    usermessagingdriver-xmpp
    usermessagingserver
    worklistapp
    wsil-nonwls
    wsm-pm
```

7. Run the `OracleUMS.deployUserMessagingDriver` command to deploy the UMS driver(s) as follows:

```
wsadmin> OracleUMS.deployUserMessagingDriver('email','email01','-node
DefaultNode01 -cell DefaultCell01 -server soa_server1 -usedefaultbindings
-defaultbinding.virtual.host default_host')
```

8. To verify that the newly deployed UMS driver(s) is listed along with the rest of enterprise applications and drivers, run the following command:

```
wsadmin>print AdminApp.list()
DMS Application_11.1.1.1.0
    DefaultToDoTaskFlow
    Dmgr DMS Application_11.1.1.1.0
    FMW Welcome Page Application_11.1.0.0.0
    User Messaging Parlay X Sample App
    composer
    em
    oracle-bam
    soa-infra
    usermessagingdriver-email
    usermessagingdriver-email-email01
    usermessagingdriver-smp
    usermessagingdriver-voicexml
    usermessagingdriver-xmpp
    usermessagingserver
    worklistapp
    wsil-nonwls
    wsm-pm
```

Make sure that the newly deployed UMS drivers(s) is listed. In the above example, `usermessagingdriver-email-email01` is the new driver.

For more information about using the wsadmin utility to configure Oracle Fusion Middleware products, see *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*. For more information about managing Oracle SOA Suite on IBM WebSphere, see *Oracle Fusion Middleware Third-Party Application Server Guide*.

28.2 Undeploying and Unregistering Drivers

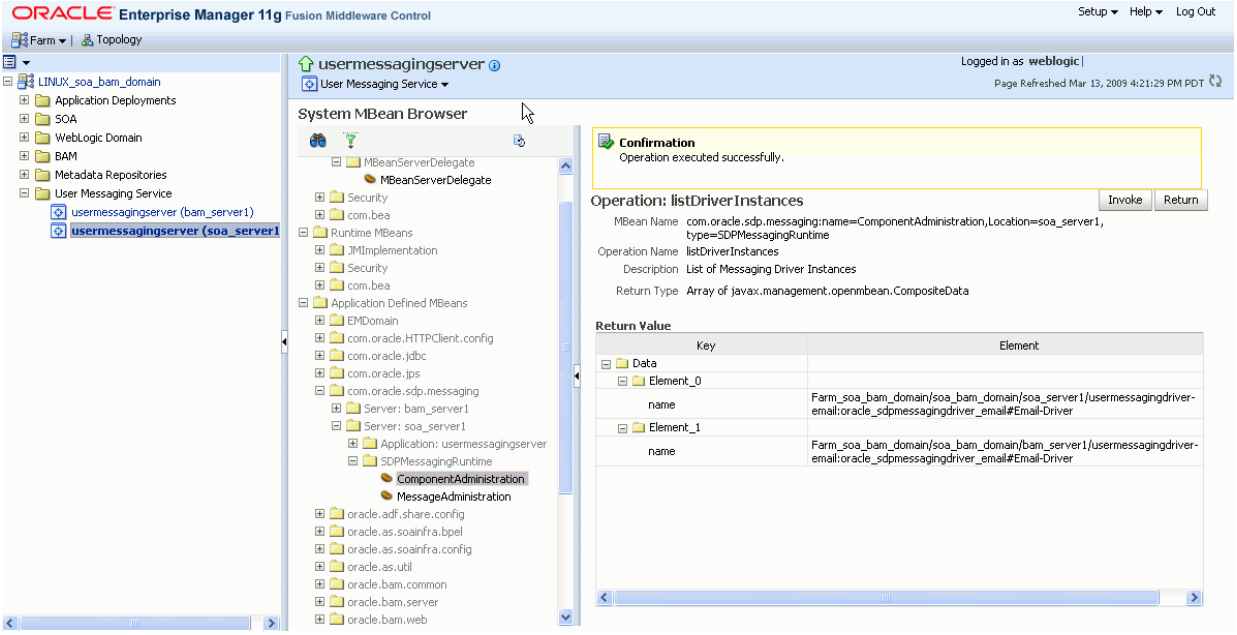
Since Messaging Drivers are standard JEE applications, they can be undeployed from the Oracle WebLogic Server using standard Oracle WebLogic tools such as the Administration Console or WLST.

However, since the UMS server keeps track of the messaging drivers that have been registered with it in a persistent store (database), this registration must be cleaned in a separate step using a runtime MBean exposed by the UMS server. The procedure to do this from Oracle Enterprise Manager Fusion Middleware Control is as follows.

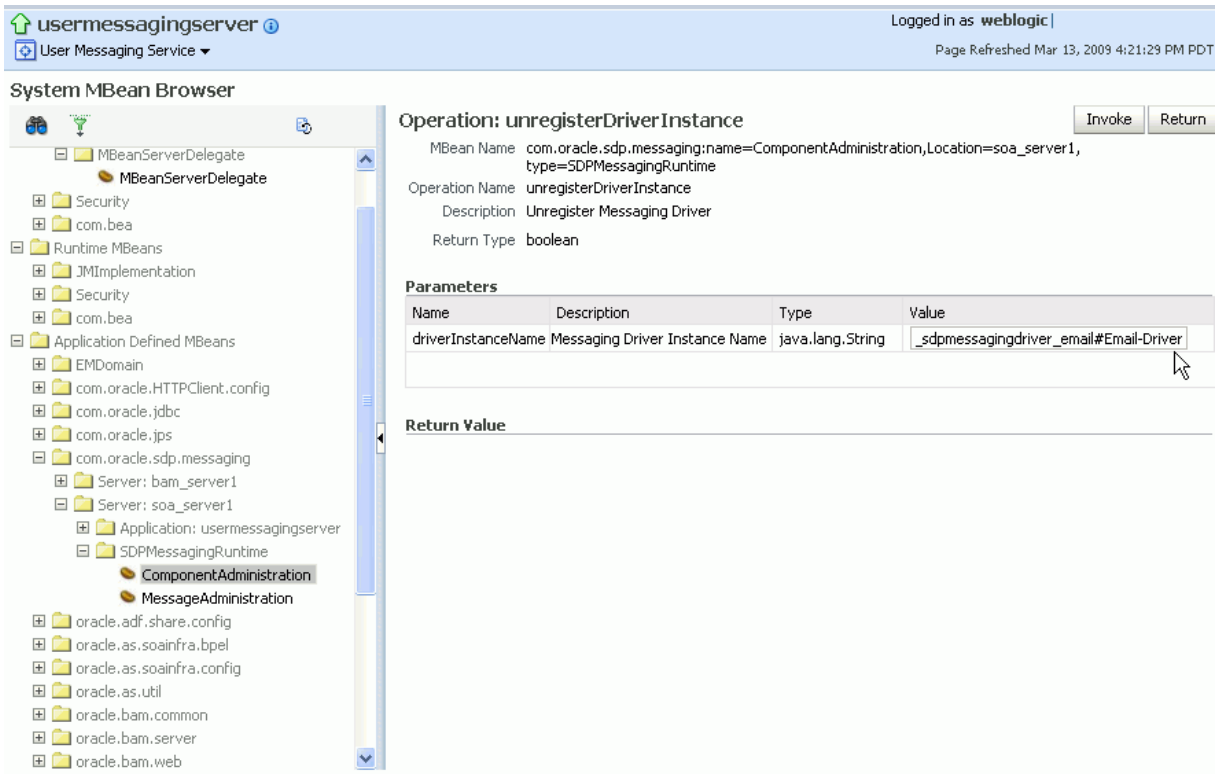
To undeploy and unregister drivers:

1. Ensure the UMS server is available.
2. In Oracle Enterprise Manager Fusion Middleware Control, select any `usermessagingserver` target in the domain.
3. From the target's menu, select **System MBean Browser**.
4. In System MBean Browser, locate the *ComponentAdministration* MBean of `usermessagingserver`:

   Expand the folder **com.oracle.sdp.messaging** > **Server** (such as **Server: soa_server1**) > **SDPMessagingrun time** > **ComponentAdministration**.
5. Invoke the operation **listDriverInstances**.
   a. Click the **Operations** tab.
   b. Click the operation **listDriverInstances**.
   c. Click **Invoke**.
   d. Identify and copy the name of the driver you want to unregister. (for example: `/Farm_soa_bam_domain/soa_bam_domain/soa_server1/usermessagingdriver-email:oracle_sdpmessagingdriver_email#Email-Driver`)



6. Click **Return**.
7. Invoke the operation **unregisterDriverInstance** with the desired driver name.
   a. Click the operation **unregisterDriverInstance**.
   b. Paste the driver name in the **Value** field (for example: `/Farm_soa_bam_domain/soa_bam_domain/soa_server1/usermessagingdriver-email:oracle_sdpmessagingdriver_email#Email-Driver`).
   c. Click **Invoke**.



8. Check the confirmation dialog for success.

This completes the unregistration of the specified driver from the UMS server and it is no longer used in future message delivery.

Part XI

Administering Oracle JCA Adapters

This part describes how to administer Oracle JCA Adapters.

This part includes the following chapters:

- Chapter 29, "Configuring Oracle JCA Adapters"
- Chapter 30, "Monitoring Oracle JCA Adapters"

29

Configuring Oracle JCA Adapters

This chapter describes how to configure inbound and outbound end point properties during runtime for Oracle JCA adapters.

This chapter includes the following topics:

- Section 29.1, "Configuring the End Point Properties for an Inbound Adapter"
- Section 29.2, "Configuring the End Point Properties for an Outbound Adapter"

29.1 Configuring the End Point Properties for an Inbound Adapter

This section describes how to configure the end point properties for an inbound adapter. It includes the following topics:

- Section 29.1.1, "Editing a Predefined Property for an Inbound Adapter"
- Section 29.1.2, "Adding Predefined Properties for an Inbound Adapter"
- Section 29.1.3, "Creating a New Property for an Inbound Adapter"
- Section 29.1.4, "Deleting a Property for an Inbound Adapter"
- Section 29.1.5, "Reverting a Property Value for an Inbound Adapter"

29.1.1 Editing a Predefined Property for an Inbound Adapter

The properties of an adapter are usually defined in Oracle JDeveloper during design time. However, it is possible edit the predefined properties at a later stage using Oracle Enterprise Manager Fusion Middleware Control.

To edit a predefined property for an inbound adapter:

1. Navigate to a SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composite applications is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA composite home page is displayed. 4. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Edit a predefined property:
   a. Click **Properties** to see a list of the currently defined binding properties.
   b. Select the property you want to edit.
   c. Edit the value in the **Value** text box, and then click **Save**.

      You have edited a predefined property for an inbound adapter.

For more information about configuring adapters, see Section 34.1, "Configuring Service and Reference Binding Component Properties" and *Oracle Fusion Middleware User's Guide for Technology Adapters*.

29.1.2 Adding Predefined Properties for an Inbound Adapter

The properties of an adapter are usually defined in Oracle JDeveloper during design time. However, it is possible to add properties at a later stage using Oracle Enterprise Manager Fusion Middleware Control.

> **Note:** Though the Properties page list both endpoint and the binding properties, you can *only* add endpoint properties.

To add a predefined property for an inbound adapter:

1. Navigate to a SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composite applications is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA composite home page is displayed. 4. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA composite home page is displayed. 2. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Add a predefined property:
   a. Click the **Properties** tab to see a list of the currently defined binding properties.
   b. Click the **Add** button.

      A new empty row is appended to the existing list of properties.
   c. Click the **Select Values** icon in the **Name** field of the new row.

      The Properties dialog is displayed.
   d. Select a property that is valid for the particular adapter from the list of properties, and then click **OK**.
   e. Click **Save**.

      You have added a predefined property for an inbound adapter.

For more information about configuring adapters, see Section 34.1, "Configuring Service and Reference Binding Component Properties" and *Oracle Fusion Middleware User's Guide for Technology Adapters*.

29.1.3 Creating a New Property for an Inbound Adapter

The properties of an adapter are usually defined in Oracle JDeveloper during design time. However, it is possible to add new properties at a later stage using Oracle Enterprise Manager Fusion Middleware Control.

To create a new property for an inbound adapter:

1. Navigate to a SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composite applications is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA composite home page is displayed. 4. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Create a new property:
   a. Click the **Properties** tab to see the list of the currently defined binding properties.
   b. Click the **Add** button.

      A new empty row is appended to the existing list of properties.
   c. Specify the property name and value in the **Name** and the **Value** fields of the new row.
   d. Click **Save**.

      You have created a new property for an inbound adapter.

For more information about configuring adapters, see Section 34.1, "Configuring Service and Reference Binding Component Properties" and *Oracle Fusion Middleware User's Guide for Technology Adapters*.

29.1.4 Deleting a Property for an Inbound Adapter

You can delete only properties that you added from the predefined list of properties or the ones that you newly created.

To delete a property for an inbound adapter:

1. Navigate to a SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composite applications is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Delete a property:
   a. Click the **Properties** tab to see a list of the currently defined binding properties.
   b. Select the property you want to delete, and then click **Delete**.

      A message asking you to confirm your action is displayed.
   c. Click **OK** to confirm.
   d. Click **Save**.

      You have deleted a property for an inbound adapter.

For more information about configuring adapters, see Section 34.1, "Configuring Service and Reference Binding Component Properties" and *Oracle Fusion Middleware User's Guide for Technology Adapters*.

29.1.5 Reverting a Property Value for an Inbound Adapter

You can only revert the properties that you have changed. Also note that you can perform the revert operation only on the existing property values and not on those that you added from the predefined list of properties or the ones that you created.

To revert a property value for an inbound adapter:

1. Navigate to a SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composite applications is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Revert a property value for an inbound adapter:
   a. Click the **Properties** tab to see a list of the currently defined binding properties.
   b. Select the property you want to revert, and then click **Revert**.

      A message asking you to confirm your action is displayed.
   c. Click **OK** to confirm.
   d. Click **Save**.

      You have reverted a property value for an inbound adapter.

For more information about configuring adapters, see Section 34.1, "Configuring Service and Reference Binding Component Properties" and *Oracle Fusion Middleware User's Guide for Technology Adapters*.

29.2 Configuring the End Point Properties for an Outbound Adapter

This section describes how to configure the end point properties for an outbound adapter. It includes the following topics:

- Section 29.2.1, "Editing a Predefined Property for an Outbound Adapter"
- Section 29.2.2, "Adding a Predefined Property for an Outbound Adapter"
- Section 29.2.3, "Creating a New Property for an Outbound Adapter"
- Section 29.2.4, "Deleting a Property for an Outbound Adapter"
- Section 29.2.5, "Reverting a Property Value for an Outbound Adapter"

29.2.1 Editing a Predefined Property for an Outbound Adapter

The properties of an adapter are usually defined in Oracle JDeveloper during design time. However, it is possible to edit the predefined properties at a later stage using Oracle Enterprise Manager Fusion Middleware Control.

To edit a predefined property for an outbound adapter:

1. Navigate to a SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composite applications is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Reference Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Reference Home page is displayed.

2. Edit a predefined property:
   a. Click the **Properties** tab to see a list of the currently defined binding properties.
   b. Select the property you want to edit.
   c. Edit the value in the **Value** text box, and then click **Save**.

   You have edited a predefined property for an outbound adapter.

   Note: For Oracle MQ Series Adapter in an asynchronous outbound request/reply scenario, properties are differentiated by an `(Enqueue)` or `(Dequeue)` label. For example, `QueueName(Enqueue)` is used for putting a message and `QueueName(Dequeue)` is used for dequeuing the reply.

   When editing Oracle MQ Series Adapter properties in this scenario, note the following:

   - If you change the `ReplyToQueueName(Enqueue)` property, you must also change the `QueueName(Dequeue)` property to the same value.
   - If you change the `MessageId(Dequeue)` property, you must also change the `MessageId(Enqueue)` property to the same value.
   - If you change the `CorrelationId(Dequeue)` property, you must also change the `CorrelationId(Enqueue)` property to the same value.

For more information about configuring adapters, see Section 34.1, "Configuring Service and Reference Binding Component Properties" and *Oracle Fusion Middleware User's Guide for Technology Adapters*.

29.2.2 Adding a Predefined Property for an Outbound Adapter

The properties of an adapter are usually defined in Oracle JDeveloper during design time. However, it is possible to add predefined properties at a later stage using Oracle Enterprise Manager Fusion Middleware Control.

To add a predefined property for an outbound adapter:

1. Navigate to a SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composite applications is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Reference Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Reference Home page is displayed.

2. Add a predefined property:
   a. Click the **Properties** tab to see a list of the currently defined binding properties.
   b. Click the **Add** button.

      A new empty row is appended to the existing list of properties.
   c. Click the **Select Value** icon in the **Name** field of the new row.

      The Properties dialog is displayed.
   d. Select a property that is valid for the particular adapter from the list of properties, and then click **OK**.
   e. Click **Save**.

      You have added a predefined property for an outbound adapter.

For more information about configuring adapters, see Section 34.1, "Configuring Service and Reference Binding Component Properties" and *Oracle Fusion Middleware User's Guide for Technology Adapters*.

29.2.3 Creating a New Property for an Outbound Adapter

The properties of an adapter are usually defined in Oracle JDeveloper during design time. However, it is possible to add new properties at a later stage using Oracle Enterprise Manager Fusion Middleware Control.

To create a new property for an outbound adapter:

1. Navigate to a SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composite applications is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Reference Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Reference Home page is displayed.

2. Create a new property:
   a. Click the **Properties** tab to see a list of the currently defined binding properties.
   b. Click the **Add** button.

      A new empty row is appended to the existing list of properties.
   c. Specify the property name and value in the **Name** and the **Value** fields of the new row.
   d. Click **Save**.

      You have created a new property for an outbound adapter.

For more information about configuring adapters, seeSection 34.1, "Configuring Service and Reference Binding Component Properties" and *Oracle Fusion Middleware User's Guide for Technology Adapters*.

29.2.4 Deleting a Property for an Outbound Adapter

You can delete only properties that you added from the predefined list of properties or the ones that you newly created.

To delete a property for an outbound adapter:

1. Navigate to a SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composite applications is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Reference Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Reference Home page is displayed.

2. Delete a property for an outbound adapter:
   a. Click the **Properties** tab to see a list of the currently defined binding properties.
   b. Select the property you want to delete, and then click **Delete**.

      A message asking you to confirm your action is displayed.
   c. Click **OK** to confirm.
   d. Click **Save**.

      You have deleted a property for an outbound adapter.

For more information about configuring adapters, see Section 34.1, "Configuring Service and Reference Binding Component Properties" and *Oracle Fusion Middleware User's Guide for Technology Adapters*.

29.2.5 Reverting a Property Value for an Outbound Adapter

You can revert changes made, if any, only for the already existing property values and not those that you added from the predefined list of properties or the ones that you newly created.

To revert a property value for an inbound adapter:

1. Navigate to a SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composite applications is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the outbound adapter (reference) from the S**ervices and References** section in the right panel. The Reference Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Reference Home page is displayed.

2. Revert a property value for an outbound adapter:
   a. Click the **Properties** tab to see a list of the currently defined binding properties.
   b. Select the property you want to revert, and then click **Revert**.

      A message asking you to confirm your action is displayed.
   c. Click **OK** to confirm.
   d. Click **Save**.

      You have reverted a property value for an outbound adapter.

For more information about configuring adapters, see Section 34.1, "Configuring Service and Reference Binding Component Properties" and *Oracle Fusion Middleware User's Guide for Technology Adapters*.

30

Monitoring Oracle JCA Adapters

This chapter describes how to monitor inbound and outbound Oracle JCA adapters, including monitoring instances and faults, recent faults and rejected messages, adapter properties, and adapter logs.

This chapter includes the following topics:

- Section 30.1, "Monitoring Instances and Faults for an Inbound Adapter"
- Section 30.2, "Monitoring Recent Faults and Rejected Messages for an Inbound Adapter"
- Section 30.3, "Monitoring Faults and Rejected Messages for an Inbound Adapter"
- Section 30.4, "Monitoring Properties for an Inbound Adapter"
- Section 30.5, "Monitoring Instances and Faults for an Outbound Adapter"
- Section 30.6, "Monitoring Recent Faults for an Outbound Adapter"
- Section 30.7, "Monitoring Faults for an Outbound Adapter"
- Section 30.8, "Monitoring Properties for an Outbound Adapter"
- Section 30.9, "Monitoring Adapter Logs"

30.1 Monitoring Instances and Faults for an Inbound Adapter

An invocation to a service from a composite may result in an error. This error is captured as a fault in the service. You can view the details of the instances and faults of the inbound adapter in the **Instances and Faults** section of the Dashboard page.

To monitor instances and faults for an inbound adapter:

1. Navigate to the SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composites is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Monitor the instances and faults for an inbound adapter:
   a. Click **Dashboard**.

      The **Dashboard** page is displayed.
   b. View the instances and faults listed in the **Instances and Faults** section.

      The details of the fault is displayed in a line chart in the **Instances and Faults** section. This line chart shows the total number of outgoing messages since the start of the server, and the total number of faults since the start of the server.

For more information about monitoring adapters, see *Oracle Fusion Middleware User's Guide for Technology Adapters*.

30.2 Monitoring Recent Faults and Rejected Messages for an Inbound Adapter

You can view the details of the recent faults and rejected messages of the inbound adapter in the **Recent Faults and Rejected Messages** section of the Dashboard page.

To monitor the recent rejected messages for an inbound adapter:

1. Navigate to the SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composites is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Monitor the recent faults and rejected messages for an inbound adapter:
   a. Click **Dashboard**.

      The Dashboard page is displayed.
   b. View the recent faults and rejected messages listed in the **Recent Faults and Rejected Messages** section.

      A list of recently rejected faults and messages with details such as error message, fault time, and the composite instance ID is displayed.

      For more information about monitoring adapters, see *Oracle Fusion Middleware User's Guide for Technology Adapters*.

30.3 Monitoring Faults and Rejected Messages for an Inbound Adapter

You can view the details of the faults and rejected messages of an inbound adapter in the Faults and Rejected Messages page.

To monitor the rejected messages for an inbound adapter:

1. Navigate to the SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composites is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Monitor the faults and rejected messages for an inbound adapter:
   a. Click **Faults and Rejected Messages**.

      The Faults and Rejected Messages page is displayed.

      A list of faults and rejected messages with details such as error message, fault time, and composite instance ID is displayed.

For more information about monitoring adapters, see *Oracle Fusion Middleware User's Guide for Technology Adapters*.

30.3.1 Searching for Rejected Messages for an Inbound Adapter

Use the *Search* feature to search for faults and rejected messages for an inbound adapter.

To search for faults and rejected messages for an inbound adapter:

1. Navigate to the SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composite applications is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Click the **Faults and Rejected Messages** tab.
3. Enter any or all of the following search criteria:
   - **Error Message Contains**

     Enter any part of the error message text.
   - **Fault ID**

     Enter the ID of the fault.
   - **Fault Time From**

     Enter the lower limit of the time when the fault could have occurred.
   - **Fault Time To**

     Enter the upper limit of the time when the fault could have occurred.
   - **Composite Instance ID**

     Enter the ID of the composite.
   - **Fault Type**

     Select a type from the list of faults available.
4. Click **Search** to start the search operation.

   The fault or rejected message matching the criteria you specified is displayed.
5. Click **Reset** to reset the search criteria.

For more information about configuring adapters, see Oracle Fusion Middleware User's Guide for Technology Adapters.

30.3.2 Deleting Rejected Messages for an Inbound Adapter

You can directly delete rejected messages from the database by specifying a search criteria.

To delete rejected messages for an inbound adapter:

1. Navigate to the SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composite applications is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Click the **Faults and Rejected Messages** tab.
3. Click **Delete Rejected Messages...**.

   The Delete: Rejected Messages dialog is displayed.
4. Specify a selection criterion for deleting rejected messages directly from the database, and then click **Delete**.

To delete a fault, you must delete the associated composite instance from the Instances page.

For more information about configuring adapters, see Oracle Fusion Middleware User's Guide for Technology Adapters.

30.4 Monitoring Properties for an Inbound Adapter

You can view the details of the properties of an inbound adapter in the Properties page.

To monitor the properties for an inbound adapter:

1. Navigate to the SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composites is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the inbound adapter (service) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Monitor the properties for an inbound adapter:
   a. Click **Properties**.

      The Properties page is displayed.

      A list of properties with details such as name and value is displayed.

      Note: In any adapter that has an inbound asynchronous request-reply scenario (the Get Message operation preceding the Send Reply operation) only details about the activation specification are displayed, and details about the interaction specification are not displayed.

For more information about monitoring adapters, see *Oracle Fusion Middleware User's Guide for Technology Adapters*.

30.5 Monitoring Instances and Faults for an Outbound Adapter

An invocation to a reference from a composite may result in an error. This error is captured as a fault in the reference. The details of the instances and faults of the outbound adapter can be viewed in the **Instances and Faults** section of the Dashboard page.

To monitor instances and faults for an outbound adapter:

1. Navigate to the SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composites is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Monitor the instances and faults for an outbound adapter:
   a. Click **Dashboard**.

      The Dashboard page is displayed.
   b. View the instances and faults listed in the **Instances and Faults** section.

      The details of the fault is displayed in a line chart in the **Instances and Faults** section. This line chart shows the total number of outgoing messages since the start of the server, and the total number of faults since the start of the server.

For more information about monitoring adapters, see *Oracle Fusion Middleware User's Guide for Technology Adapters*.

30.6 Monitoring Recent Faults for an Outbound Adapter

The details of the recent faults of the outbound adapter can be viewed in the **Recent Faults** section of the Dashboard page.

To monitor recent faults for an outbound adapter:

1. Navigate to the SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composites is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Monitor the recent faults for an outbound adapter:
   a. Click **Dashboard**.

      The Dashboard page is displayed.
   b. View the recent faults listed in the **Recent Faults** section.

      A list of recent faults with details such as name, time, and type is displayed.

For more information about monitoring adapters, see *Oracle Fusion Middleware User's Guide for Technology Adapters*.

30.7 Monitoring Faults for an Outbound Adapter

The details of the instances and faults of the outbound adapter can be viewed in the Faults page.

To monitor faults for an outbound adapter:

1. Navigate to the SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composites is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Monitor the faults for an outbound adapter:
   a. Click the **Faults** tab.

      The Faults page is displayed.
   b. Click **View**.

      A list of faults with details such as name, time, and type is displayed.

For more information about monitoring adapters, see *Oracle Fusion Middleware User's Guide for Technology Adapters*.

30.7.1 Searching for Faults for an Outbound Adapter

Use the *Search* feature to search for faults for an outbound adapter.

To search for faults for an outbound adapter:

1. Navigate to the SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composite applications is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Reference Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Reference Home page is displayed.

2. Click the **Faults** tab.
3. In the **Search** section, enter any or all of the following search criteria:
   - **Error Message Contains**

     Enter any part of the error message text.
   - **Fault ID**

     Enter the ID of the fault.
   - **Fault Time From**

     Enter the lower limit of the time when the fault could have occurred.
   - **Fault Time To**

     Enter the upper limit of the time when the fault could have occurred.
   - **Composite Instance ID**

     Enter the ID of the composite.
   - **Fault Type**

Select the type of the fault from the list of types available.

4. Click **Search** to start the search operation.

   The fault matching the criteria you specified is displayed.

5. Click **Reset** to reset the search criteria.

For more information about configuring adapters, see *Oracle Fusion Middleware User's Guide for Technology Adapters*.

30.8 Monitoring Properties for an Outbound Adapter

The details of the properties of the outbound adapter can be viewed on the Properties page.

To monitor properties for an outbound adapter:

1. Navigate to the SOA composite application by using either of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Click **Home**. The SOA Infrastructure page is displayed. 2. Click the **Deployed Composites** tab. The list of deployed composites is displayed. 3. In the **Composite** section, click a specific SOA composite application. The SOA Composite home page is displayed. 4. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Service Home page is displayed.	1. Under **soa-infra**, click a specific SOA composite application. The SOA Composite home page is displayed. 2. Click the outbound adapter (reference) from the **Services and References** section in the right panel. The Service Home page is displayed.

2. Monitor the properties for an outbound adapter:

   a. Click the **Properties** tab.

      The Properties page is displayed.

   b. Click **View**.

      A list of properties with details such as name and value is displayed.

      Note: In the case of an adapter that has an outbound asynchronous request-reply scenario (the Send Message operation preceding the Get Response operation), only details about the interaction specification are displayed, and details about the activation specification are not displayed.

For more information about monitoring adapters, see *Oracle Fusion Middleware User's Guide for Technology Adapters*.

30.9 Monitoring Adapter Logs

Oracle Fusion Middleware 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. There is only one log for all Oracle JCA Adapters, and the log is called oracle.soa.adapter.

To monitor the File adapter logs:

1. Navigate to **Composite Home** > **J2EE Application** (menu) > **Monitoring** > **Logs**.
2. Find the **oracle.soa.adapter** logger for adapters and increase verbosity.
3. Find resulting log files.
4. Cause an adapter to fail. For example, delete a directory that the file adapter writes to, or delete a table that the Database adapter reads from.
5. Ensure that the resulting log files give a good indication of the cause of failure.

For information about configuring logs, see Section 3.4, "Configuring Log Files."

Part XII

Administering Oracle B2B

This part describes how to administer Oracle B2B.

This part includes the following chapters:

- Chapter 31, "Configuring Oracle B2B"
- Chapter 32, "Monitoring Oracle B2B"

31

Configuring Oracle B2B

This chapter describes how to configure Oracle B2B to view data on B2B Bindings and configure Oracle B2B operations and attributes.

This chapter includes the following topics:

- Section 31.1, "Configuring Oracle B2B Server Properties"
- Section 31.2, "Configuring Oracle B2B Operations"
- Section 31.3, "Configuring Oracle B2B Attributes"

31.1 Configuring Oracle B2B Server Properties

You can configure the **Enable Metrics** property on the B2B Server Properties page of Oracle Enterprise Manager Fusion Middleware Control.

The metrics are enabled by default, and include data on the top five recently active document types, top five recently active trading partners, and inbound and outbound endpoints.

> **Note:** You do not need to restart the server after changing this property.

To configure Oracle B2B server properties:

1. From the **SOA Infrastructure** menu, select **SOA Administration** > **B2B Server Properties**.
2. Select **Enable Metrics** to view data on the B2B Bindings and SOA composite pages.

For information about using Oracle B2B, which enables the secure and reliable exchange of business documents between an enterprise and its trading partners, see *Oracle Fusion Middleware User's Guide for Oracle B2B*.

31.2 Configuring Oracle B2B Operations

You can configure the properties of Oracle B2B operations by setting values using the System MBean Browser.

> **Note:** Restarting the SOA server is required after updating the following MBean properties:
>
> - **b2b.inboundThreadCount**
> - **b2b.outboundThreadCount**

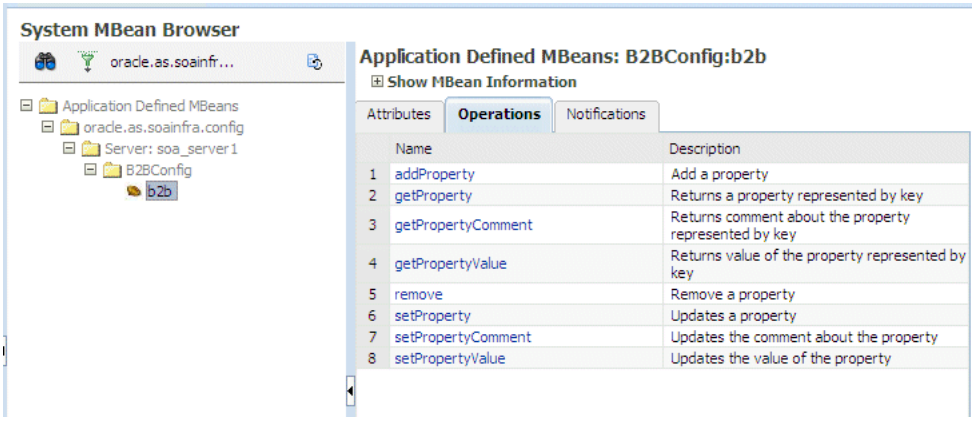
To specify Oracle B2B operation properties:

> **Note (a shortcut):** To access the Application Defined MBeans for Oracle B2B, you can also click the **More B2B Configuration Properties** link on the B2B Server Properties page.

1. Expand the **SOA** node and select the **soa-infra** node.
2. From the **SOA Infrastructure** menu, choose **Administration** > **System MBean Browser**.

   The System MBean Browser page is displayed.
3. Under **Application Defined MBeans**, expand the **oracle.as.soainfra.config** node.
4. Expand the **Server:soa_server1** node.
5. Expand the **B2BConfig** node.
6. Click the **b2b** MBean.

   The properties of the MBean are displayed in the right pane.
7. Click the **Operations** tab.
8. Click an operation in the list.



9. Provide the needed information (key, value, and an optional comment) and click **Invoke**.

   See Appendix "Setting Oracle B2B Configuration Properties in Fusion Middleware Control" in *Oracle Fusion Middleware User's Guide for Oracle B2B* for a list of property names and valid values, including properties for turning off validation during deployment, setting the MDS cache size, setting thread count and thread sleep time, specifying how functional acknowledgments are handled, setting payload obfuscation, and more.

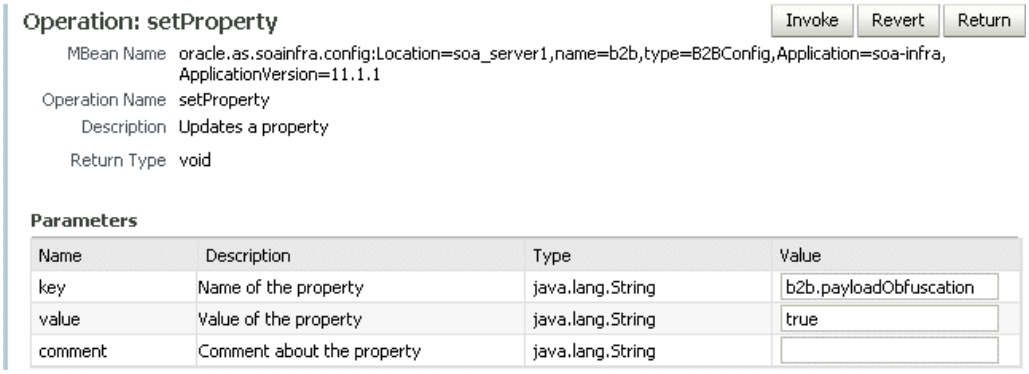
31.3 Configuring Oracle B2B Attributes

You can configure Oracle B2B attributes by setting values using the System MBean Browser.

To specify Oracle B2B attribute properties:

1. Perform Steps 1 through 6 in Section 31.2, "Configuring Oracle B2B Operations."
2. Click the **Attributes** tab.
3. Click **Properties**.
4. Expand **Element_0**, **Element_1**, and so on to find the property you want to change.
5. Note the property name and value, and click **Return**.

   For example, under **Element_3**, you see the **b2b.payloadObfuscation** property with the default value **false**.
6. Click the **Operations** tab.
7. Click **setProperty**.
8. Enter values for the key, value, and optional comments.



   For example, to enable payload obfuscation, set **b2b.payloadObfuscatio**n to **true**.
9. Click **Invoke**.
10. Click **Return**.

32

Monitoring Oracle B2B

This chapter describes how to monitor Oracle B2B, including monitoring the Oracle B2B infrastructure, viewing Oracle B2B binding component message flow, viewing services and references, and accessing Oracle B2B reports from the Oracle B2B composite flow trace page.

This chapter includes the following topics:

- Section 32.1, "Monitoring the Oracle B2B Infrastructure"
- Section 32.2, "Accessing Oracle B2B from the B2B Infrastructure Page"
- Section 32.3, "Viewing the Message Flow of an Oracle B2B Binding Component"
- Section 32.4, "Viewing Services and References"
- Section 32.5, "Accessing Oracle B2B Reports from the Oracle B2B Composite Flow Trace Page"

32.1 Monitoring the Oracle B2B Infrastructure

To monitor the Oracle B2B infrastructure, metrics must be enabled (the default) on the B2B Server Properties page.

Use the **B2B Server Properties** link under **Related Links** on the B2B Infrastructure (SOA Binding) page to change the setting. See Section 31.1, "Configuring Oracle B2B Server Properties," for more information on enabling metrics.

Table 32–1 describes the information displayed on the B2B Infrastructure (SOA Binding) page.

Table 32–1 B2B Infrastructure (SOA Binding)

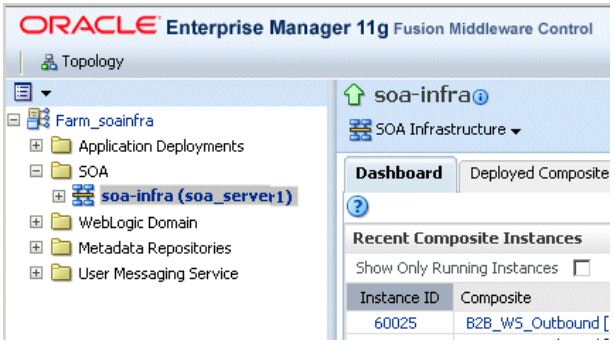
Section/Column	Description
Top 5 Recently Active Document Types	Shows the active document types with the maximum number of messages exchanged (inbound and outbound combined) during the current session of the server. The document types listed in this section are from Oracle DMS metrics, triggered by runtime sensors. This data is not persisted. Therefore, if Oracle B2B is restarted, then new data based on Oracle B2B activity appears here.
Number of Messages Processed	Shows the number of document messages exchanged between the host and trading partners. **Outbound** indicates messages sent from the host to the trading partner and **Inbound** indicates messages sent from the trading partner to the host.
Average Message Processing Time (sec)	Shows the average document processing time, in seconds, for both outbound and inbound messages.

Table 32–1 (Cont.) B2B Infrastructure (SOA Binding)

Section/Column	Description
Average Message Size (kb)	Shows the average document size, in kilobytes, for both outbound and inbound messages.
Errors	Shows the document error count.
Top 5 Recently Active Trading Partners	Shows the active trading partners with the maximum number of messages exchanged (from and to combined) during the current session of the server. The trading partners listed here are from Oracle DMS metrics, triggered by runtime sensors. This data is not persisted. Therefore, if Oracle B2B is restarted, then new data based on Oracle B2B activity appears here.
Number of Messages Processed	Shows the number of messages sent and received between the host and trading partners. **From** indicates messages sent from this partner to its trading partner. **To** indicates messages received by this partner from the trading partner.
Average Message Processing Time (sec)	Shows the average document processing time, in seconds, for exchanged messages.
Average Message Size (kb)	Shows the average document size, in kilobytes, for exchanged messages.
Errors	Shows the document error count.
Inbound Endpoints	Shows the status of the listening endpoints at the time the connection was attempted (not for a later point in time). For example, in an EDI transaction using Generic File transport, when Oracle B2B reads from a directory, that directory is the inbound endpoint.
Protocol	Shows the type of transport protocol used in the exchange, for example, File, AQ, and FTP, among others.
Endpoint	Shows the location from which messages are received. The endpoint can be a URL, folders, or path, among others.
Status	Shows the status (up or down) of the endpoint (protocol) the last time a connection was attempted.
Outbound Endpoints	Shows the status of the delivery endpoints at the time the delivery was attempted (not for a later point in time). For example, in an EDI transaction using Generic File transport, when Oracle B2B writes to a directory, that directory is the outbound endpoint.
Protocol	Shows the type of transport protocol used in the exchange, for example, File, AQ, and FTP, among others.
Endpoint	Shows the location to which messages are sent. The endpoint can be a URL, folders, or path, among others.
Status	Shows the status (up or down) of the endpoint (protocol) the last time a delivery was attempted.

To monitor the Oracle B2B infrastructure using Oracle Enterprise Manager Fusion Middleware Control:

1. Expand the **SOA** node.
2. Select the SOA Infrastructure, for example, **soa-infra (soa_server1)**.



3. From the **SOA Infrastructure** menu, select **Bindings** > **B2B**.

   See Table 32–1 for a description of the information displayed.

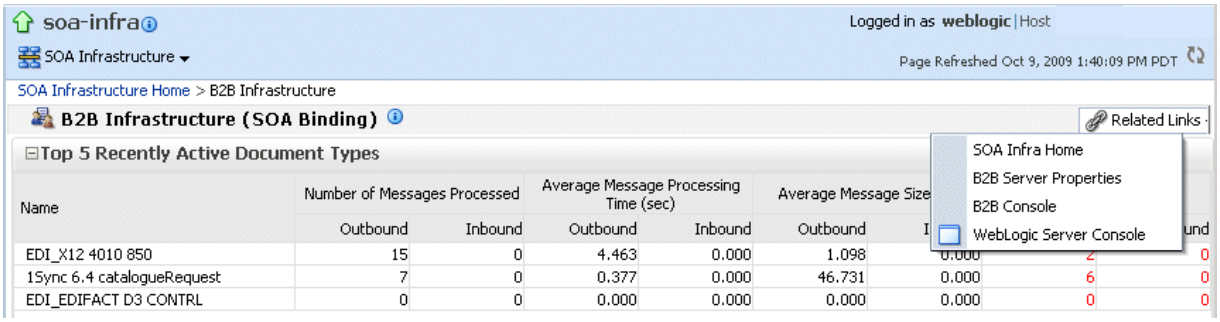
Information displayed in Fusion Middleware Control is based on DMS metrics, including inbound and outbound endpoints. Therefore, changes to Oracle B2B runtime data are not necessarily or immediately reflected in Fusion Middleware Control.

For example, if you purge runtime data or make Oracle B2B configuration changes in the Oracle B2B interface, the changes do not affect DMS metrics. To view current runtime data, use the Oracle B2B interface. Fusion Middleware Control data shows Oracle B2B message traffic from the time that the Oracle B2B (SOA) server starts.

32.2 Accessing Oracle B2B from the B2B Infrastructure Page

Use the **B2B Console** link under **Related Links**, as shown in Figure 32–1, to log in to Oracle B2B.

Figure 32–1 Accessing Oracle B2B from Oracle Enterprise Manager Fusion Middleware Control



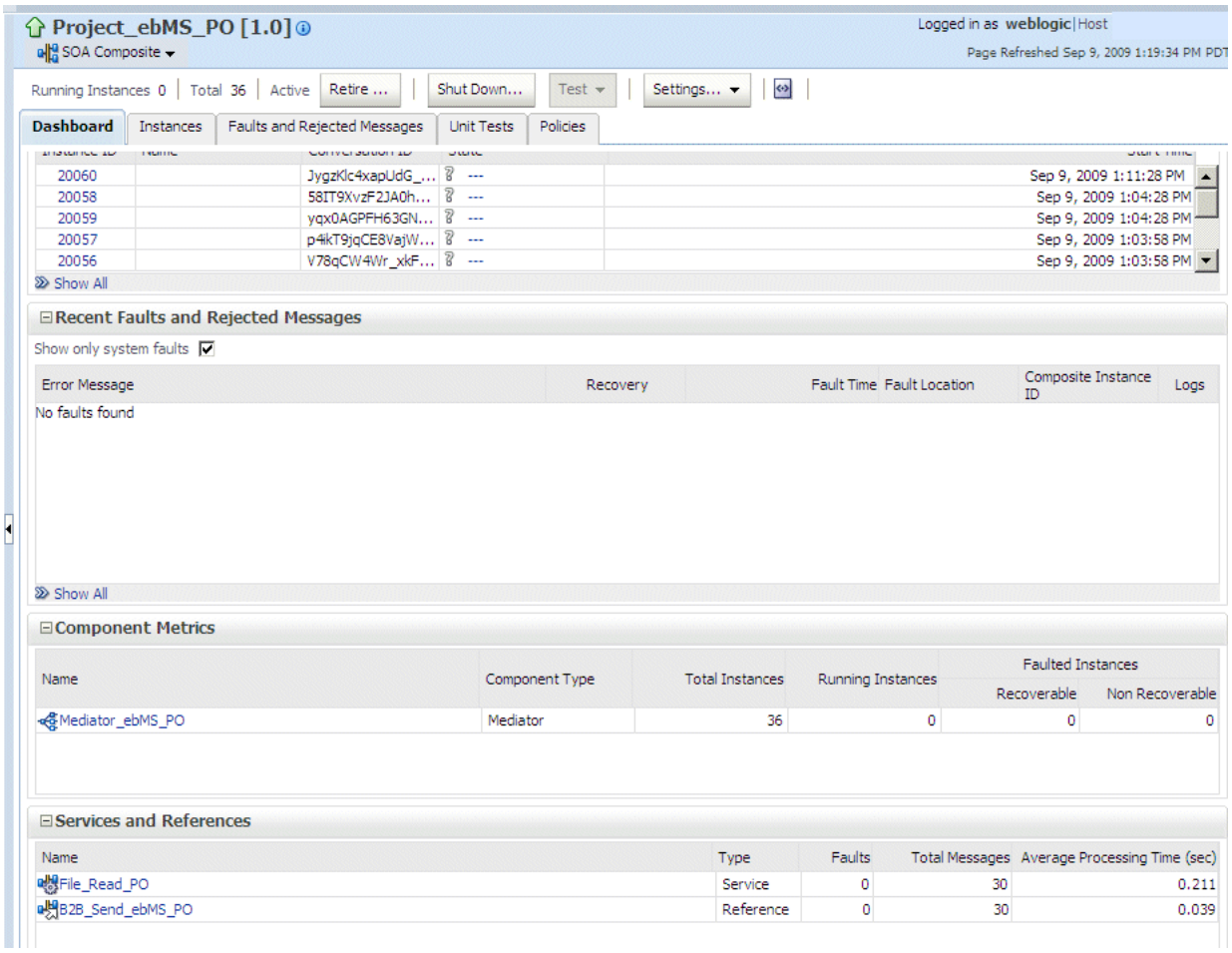
See *Oracle Fusion Middleware User's Guide for Oracle B2B* for information on using the Oracle B2B interface.

32.3 Viewing the Message Flow of an Oracle B2B Binding Component

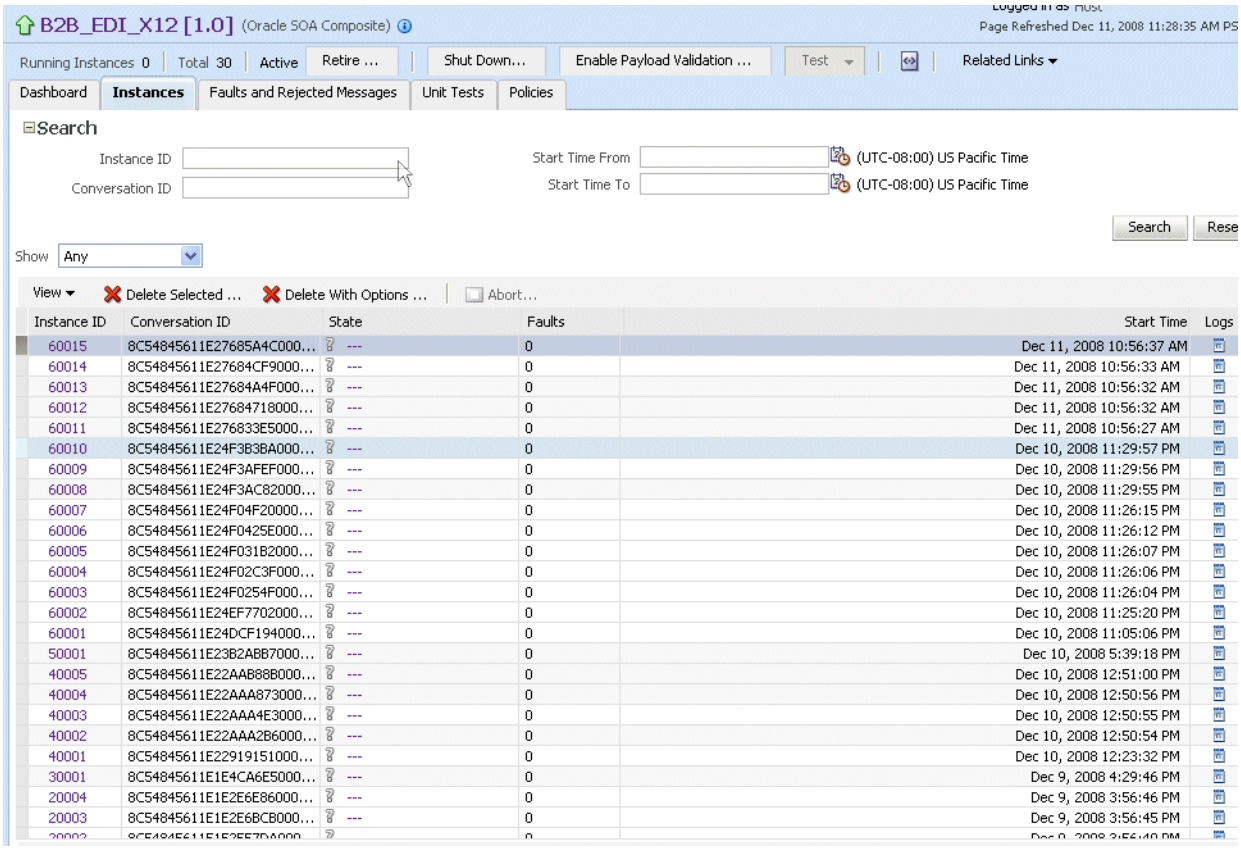
To view the message flow of an Oracle B2B binding component in a SOA composite application instance:

1. Select a SOA composite application with an Oracle B2B binding component.

   A list of the recent instances is displayed in the Dashboard page.

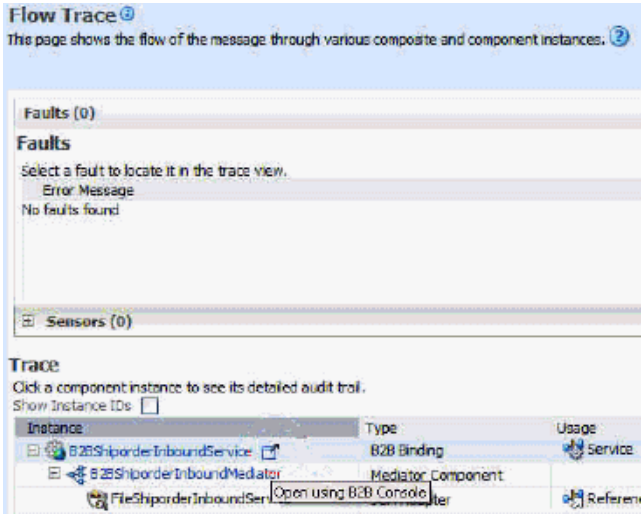


2. To see all instances of this SOA composite application, click the **Instances** tab.



Note: You can see details of a specific message by searching on the instance ID on the **Reports** page of the Oracle B2B interface.

3. Click a specific instance in the **Instance ID** list to see faults and the flow trace.



The Flow Trace page displays the following details:

- The **Faults** section shows the faults occurring and sensor information collected in the services, service components, and references that comprise the SOA composite application.

- The **Trace** section shows the sequence of the message flow through the services, service components, and references that comprise the SOA composite application.

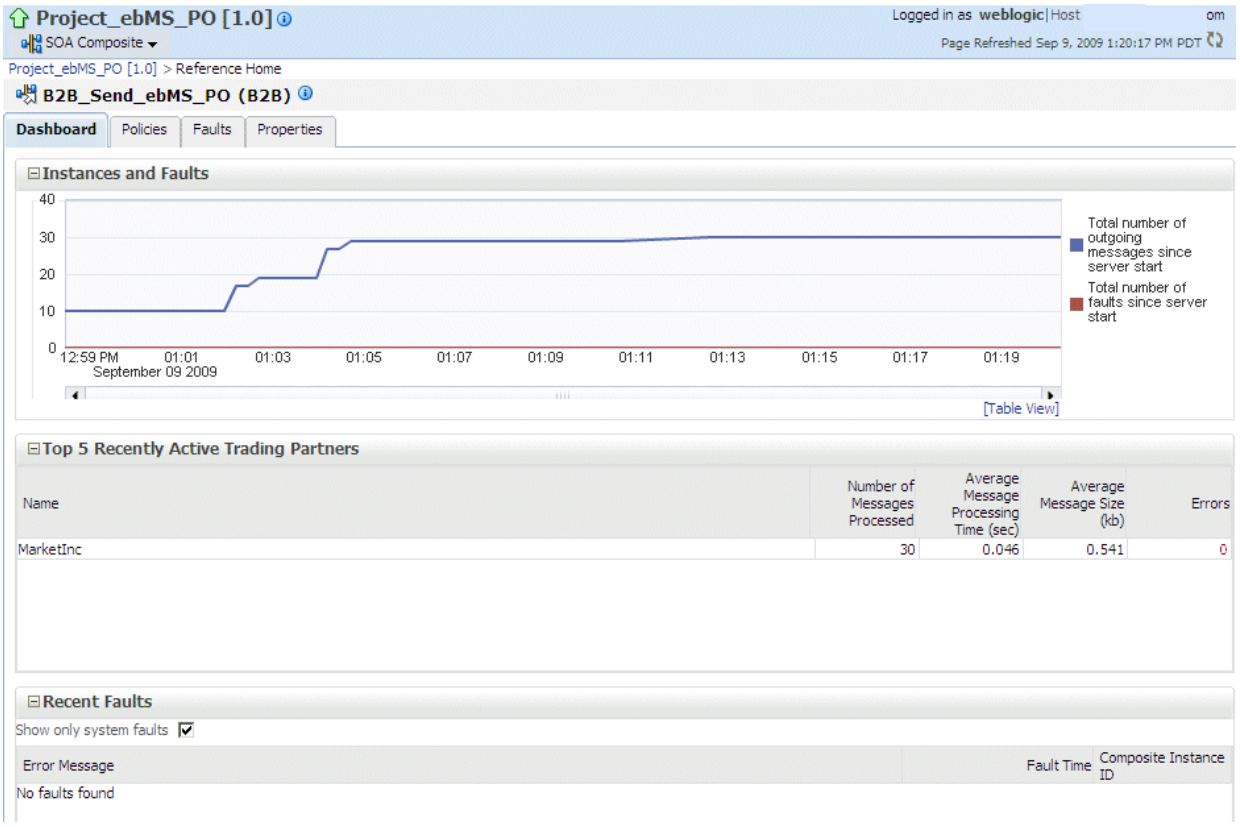
32.4 Viewing Services and References

The Dashboard page for a composite application, lists the services and references used in the composite.

When Oracle B2B is used as a *service* (inbound), it receives messages from trading partners and delivers them to SOA composite applications. When Oracle B2B is used as a *reference* (outbound), it sends messages from the SOA composite application to partners.

Details of the Oracle B2B reference shown in Figure 32–2, **B2B_Send_ebMS_PO**, are displayed on the Reference home page.

Figure 32–2 The Oracle B2B Reference, B2B_Send_ebMS_PO



The **Instances and Faults** section shows the number of outgoing messages and the number of faults since the server was started. See Table 32–1 for a description of the **Top 5 Recently Active Trading Partners** section.

The **Recent Faults** section lists the faults, including details such as the error message, time of the fault, and the associated composite instance ID. Faults can be on the incoming messages processed by a service binding component or on the outgoing messages processed by a reference binding component.

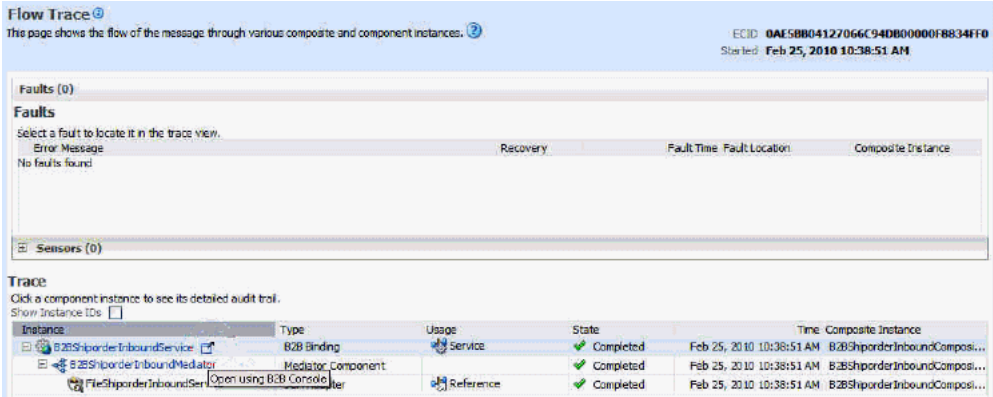
While all errors appear in the Oracle B2B console, only a subset of faults appears in Oracle Enterprise Manager Fusion Middleware Control. This subset includes all

inbound messages containing an error after trading partner identification in Oracle B2B.

See Chapter 4, "Monitoring the SOA Infrastructure,"and Chapter 35, "Monitoring Service and Reference Binding Components," for more information.

32.5 Accessing Oracle B2B Reports from the Oracle B2B Composite Flow Trace Page

The Oracle B2B composite Flow Trace page displays Oracle B2B faults and traces. Click an instance to open the Oracle B2B console and view a report.



Part XIII

Administering Business Events

This part describes how to administer business events.

This part includes the following chapter:

- Chapter 33, "Managing Business Events"

33

Managing Business Events

This chapter describes how to manage business events, including subscribing to business events, managing subscribers, and recovering from business event faults. Business events consist of message data sent as the result of an occurrence in a business environment. When a business event is published, other service components or database agents can subscribe to it.

This chapter includes the following topics:

- Section 33.1, "Introduction to the Event Delivery Network"
- Section 33.2, "Subscribing to Business Events"
- Section 33.3, "Managing Business Event Subscribers"
- Section 33.4, "Recovering from Business Event Faults"

For information about troubleshooting business event issues, see Section B.5, "Event Delivery Network Troubleshooting."

For more information about business events, see Chapter "Using Business Events and the Event Delivery Network" of *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

33.1 Introduction to the Event Delivery Network

The subsystem of Oracle SOA Suite that accepts published business events and delivers them to the subscribers is called the Event Delivery Network (EDN). EDN has two different implementations:

- EDN-DB

  EDN-DB uses an Oracle database as a back-end store and depends on Oracle-specific features.

- EDN-JMS

  EDN-JMS uses a JMS queue as a back-end store.

The correct implementation is configured during installation by Oracle Universal Installer. You can also make postinstallation changes. From an application development point of view, the two implementations are identical and the behavior is the same.

33.2 Subscribing to Business Events

You can subscribe to business events in Oracle Enterprise Manager Fusion Middleware Control. Business events consist of message data sent as the result of an

occurrence in a business environment. You create business events in Oracle JDeveloper and include them in SOA composite applications that you deploy to Oracle Enterprise Manager Fusion Middleware Control. Service components and database agents can subscribe to business events. Only database agents can be subscribed to from Oracle Enterprise Manager Fusion Middleware Control. Service components are only subscribed to from Oracle JDeveloper. When a business event is published, the entity subscribed to that event receives it.

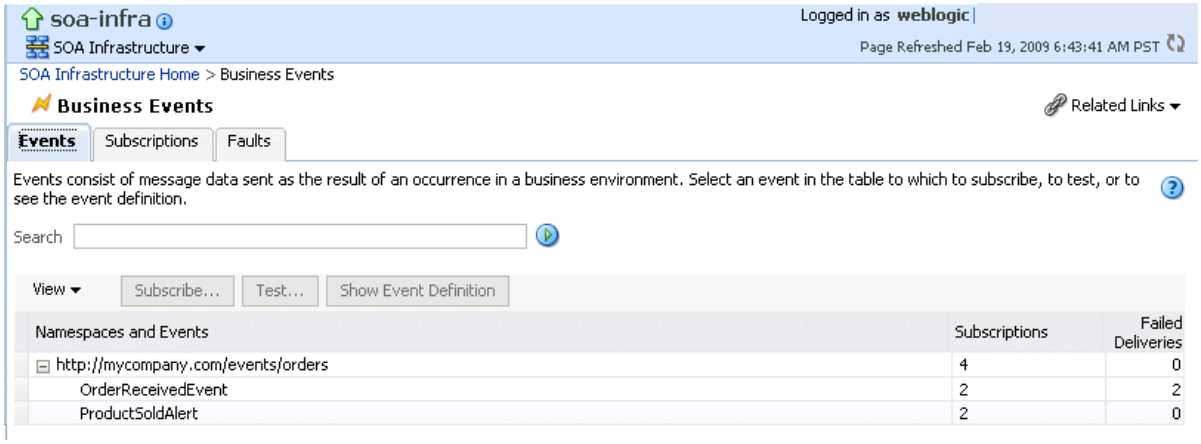
To subscribe to business events:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
Select **Business Events**.	Right-click **soa-infra** and select **Business Events**.

The Events page displays the following details:

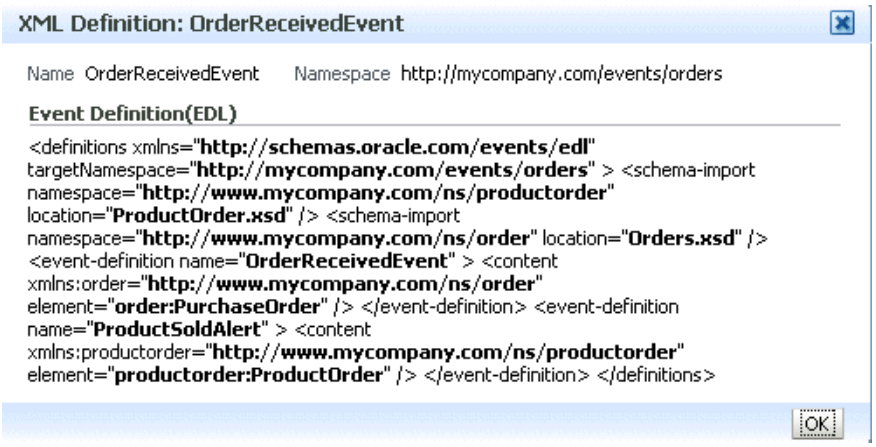
- A utility for searching for a specific business event by specifying a full or partial name and clicking the **Search** icon. Click the **Help** icon for details.
- Business events, including the namespace used, event name, number of subscriptions to each event, and number of failed event deliveries. Business events are contained within their namespace.



2. Select a specific event in the **Namespaces and Events** section.
3. Click **Show Event Definition**.

   The event definition language (EDL) file for the selected event appears. The business event is defined using EDL. EDL is a schema used to build business event definitions.

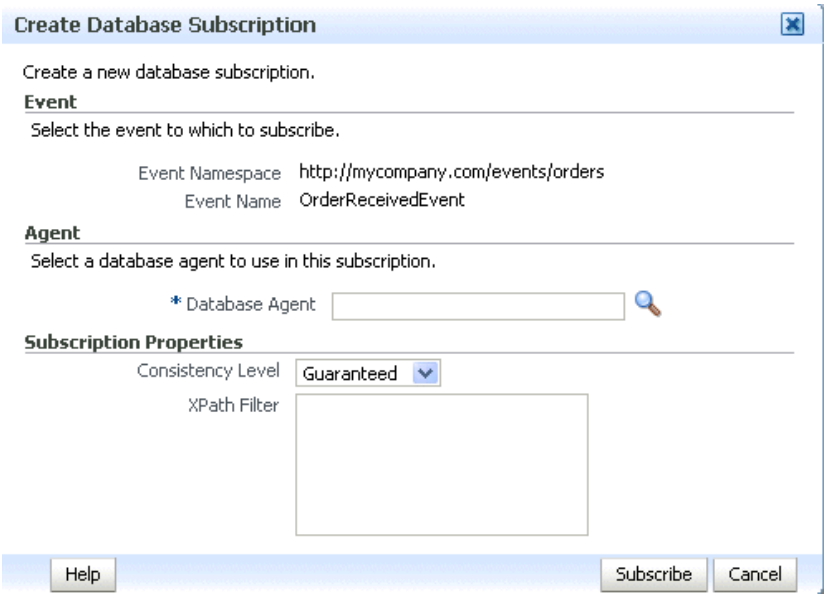
   In this example, two business events named **OrderReceivedEvent** and **ProductSoldAlert** appear in the event definition. The namespace (**orders**) and associated schema file (**ProductOrder.xsd**) are referenced.



4. Click **OK** to close the dialog.
5. Click **Subscribe** to subscribe a database agent to the event selected in the **Namespaces and Events** table.

> **Note:** The Events page for EDN-JMS implementations does not display the **Subscribe** button for subscribing a database agent to the selected event.

The Create Database Subscription to an Event dialog appears. You create an agent to listen for and subscribe to appropriate events when they are published.



6. Enter the following values. An asterisk indicates a required field.

Field	Description
Event Namespace	Displays the event namespace.
Event Name	Displays the event name.

Field	Description
Database Agent	Specify a database agent name or click **Create Agent** to create an agent. The agent listens for and subscribes to appropriate events when they are published.
Consistency Level	Select one of the following options: ▪ **Guaranteed**: Delivers events to the subscriber asynchronously without a global transaction. The subscriber can choose to create its own local transaction for processing, but it is committed independently of the rest of the event processing. By default, guaranteed delivery attempts to deliver the event to the subscriber for three times. If it cannot deliver the event in these three attempts, then the delivery fails. Also, there is a possibility that a system failure can cause an event to be delivered multiple times because there is no global transaction. If the subscriber throws an exception (or fails in any way), the exception is logged, but the event is not resent. ▪ **Immediate**: Delivers events to the subscriber on the same transaction and same thread as the publisher. The publish call does not return until all immediate subscribers have completed processing. If any subscribers throw an exception, no additional subscribers are invoked and an exception is thrown to the publisher.
XPath Filter	If you want to filter the event, specify an XPath expression. When the subscription is delivered and the expression logic is satisfied, the event is accepted for delivery. If the logic is not satisfied, the event is not delivered.

7. Click **Subscribe**.
8. Click **Test** to test a selected event. This action enables you to publish a test event that subscribers can act upon.

   The Test Event dialog appears.
9. Specify the XML payload to use in the test.
10. Click **Publish**.

> **Note:** Enforcement of policies for event subscriptions is not supported in this release. You can attach or detach a policy to or from a service component that subscribes to a business event (such as Oracle Mediator) without being warned. This action does not result in any errors; policy enforcement simply does not occur.

For more information, see the following documentation:

- Section 1.2.8, "Introduction to the Contents of SOA Composite Applications"
- Chapter "Using Business Events and the Event Delivery Network" of *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite* for details about business events

33.3 Managing Business Event Subscribers

You can also create, edit, and delete existing database subscriptions. When a business event is published, service components and database agents can subscribe to it. Use this page to view all subscriptions to business events. This page also enables you to manage event subscriptions by database agents. Service component subscriptions are

created in Oracle JDeveloper during design time and cannot be modified in Oracle Enterprise Manager Fusion Middleware Control.

Notes:

- If your SOA composite application includes a business event subscription, and you deploy different revisions of the composite, all event subscriptions from all revisions of the composite are active and receive messages. To receive the event with the latest revision of the composite only, it is recommended that you retire all previous revisions of the composite.
- A BPEL component in a retired SOA composite application cannot receive a business event, even if the event is associated with a noninitiating receive activity of the BPEL process.

To manage business event subscribers:

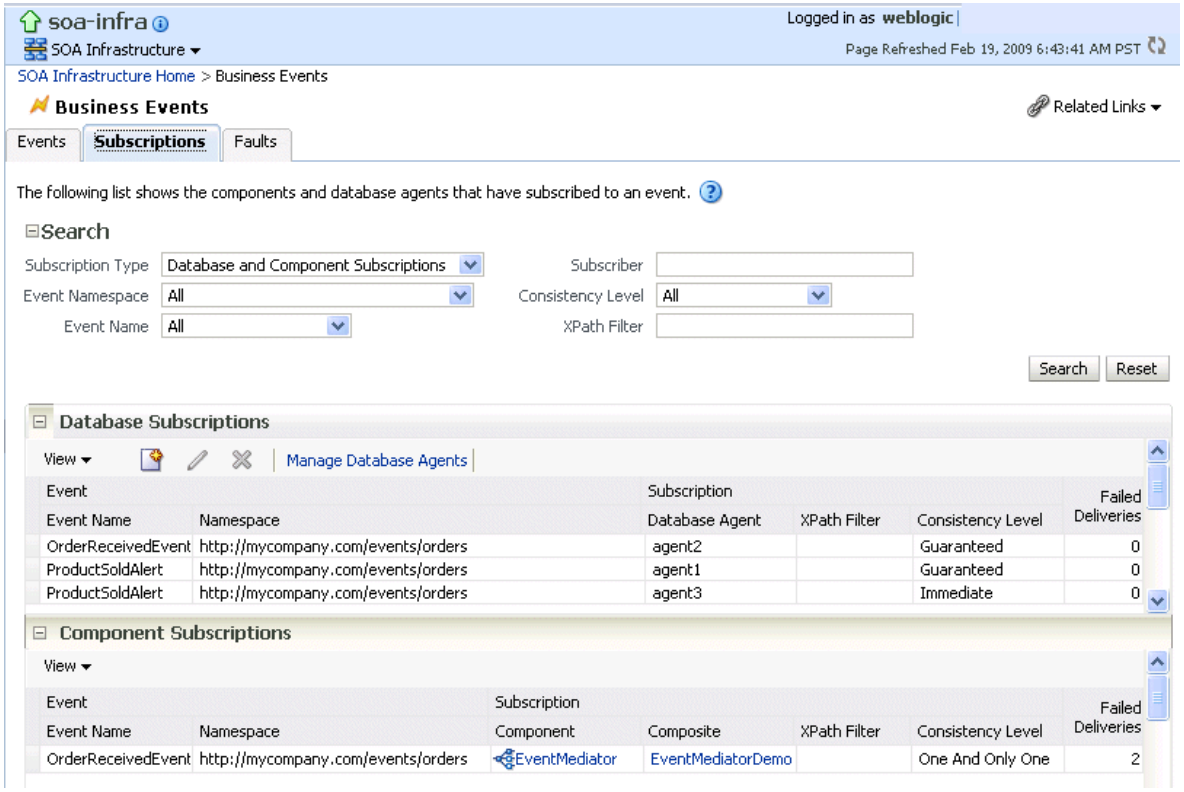
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Business Events**.	1. Right-click **soa-infra**. 2. Select **Business Events**.

2. Click **Subscriptions**.

   The Subscriptions page displays the following details:

   - A utility for searching for a specific subscription by specifying criteria and clicking **Search**. Click the **Help** icon for details.
   - The database subscriptions, including the event name, namespace used, subscription agent name, optionally defined XPath filters, consistency level, and number of failed deliveries. You create database subscriptions to business events on this page during runtime.
   - The component subscriptions, including the event name, namespace used, the service component subscribing to the event, the SOA composite application, any optionally defined XPath filters, and the consistency level.



Note: For business events implementing EDN - JMS, the **Create Database Agent** icon, **Edit Database Agent** icon, **Delete Database Agent** icon, and **Manage Database Agents** button are not displayed.

3. Click the appropriate icon below the **Database Subscriptions** title to create, edit, or delete a database subscription.

4. Click **Manage Database Agents** to edit the PL/SQL procedure or delete agents not currently subscribing to events.
5. In the **Subscription Component** column, click a service component to access its home page.
6. In the **Composite** column, click a SOA composite application to access its home page.

For more information about business events, see Chapter "Using Business Events and the Event Delivery Network" of *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

33.4 Recovering from Business Event Faults

You can recover from business event faults that are identified as recoverable.

To recover from business event faults:

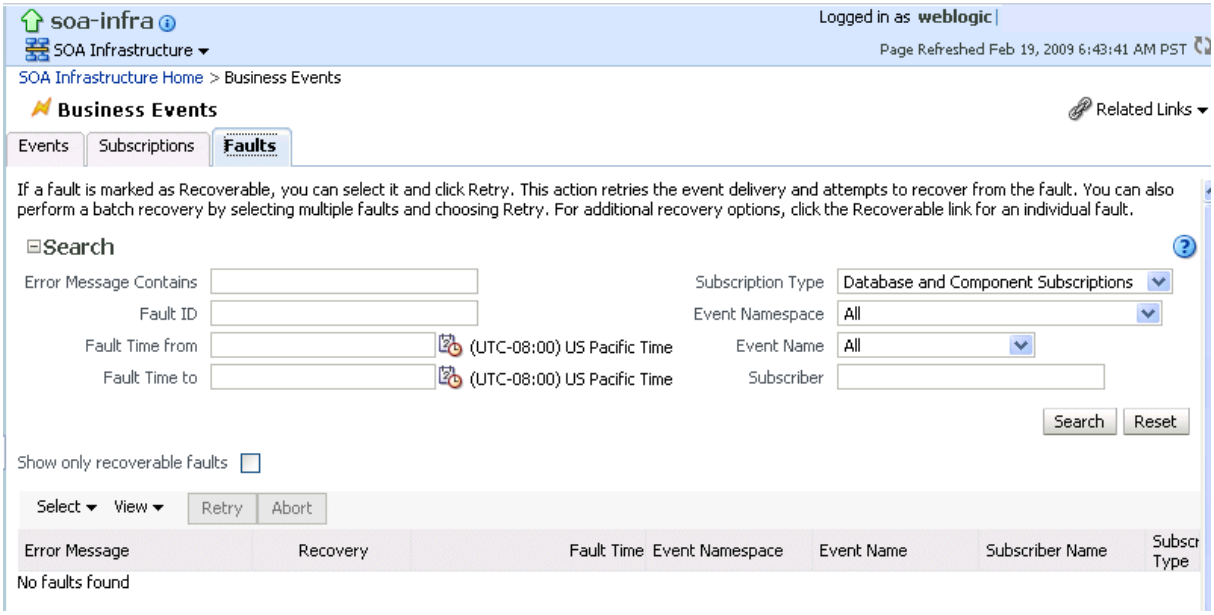
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Business Events**.	1. Right-click **soa-infra**. 2. Select **Business Events**.

2. Click **Faults**.

   The Faults page displays the following details:

   - A utility for searching for a specific business event fault by specifying criteria and clicking **Search**.
   - Faults occurring in a business event, including the error message, whether you can recover from the fault, the time at which the fault occurred, the event namespace, event name, the subscriber, and the subscription type (database or component). The **Recoverable?** column identifies faults for which recovery actions can be performed.



3. From the **View** list, select **Columns** > **Fault ID** to display the fault IDs for each error message.

4. Click a specific fault that has been identified as recoverable and select one of the following options:

Action	Description
Retry	Retries the instance in which the fault occurred
Abort	Terminates the entire instance in which the fault occurred

5. In the **Logs** column, click a specific log to access the Log Messages page with filtered messages specific to that instance.

For more information about business events, see Chapter "Using Business Events and the Event Delivery Network" of *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

Part XIV

Administering Binding Components

This part describes how to administer binding components.

This part includes the following chapters:

- Chapter 34, "Configuring Service and Reference Binding Components"
- Chapter 35, "Monitoring Service and Reference Binding Components"
- Chapter 36, "Managing Service and Reference Binding Components"

34

Configuring Service and Reference Binding Components

This chapter describes how to configure runtime properties for web service and JCA adapter service and reference binding components in SOA composite applications and change the endpoint reference and service key for Oracle Service Registry integration. Binding components are network protocols and services that connect the SOA platform with the outside world.

This chapter includes the following topic:

- Section 34.1, "Configuring Service and Reference Binding Component Properties"

For more information about binding components, see the following documentation:

- Section 1.2.5, "Introduction to Binding Components"
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

Notes:

- You cannot configure message header properties for direct binding components that invoke a SOA composite application through a remote RMI call. For this reason, the **Properties** tab does not display for direct binding components.
- You cannot change the **httpBinding** property for the HTTP binding component.
- Support is provided for adding MTOM attachments to web services. For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

34.1 Configuring Service and Reference Binding Component Properties

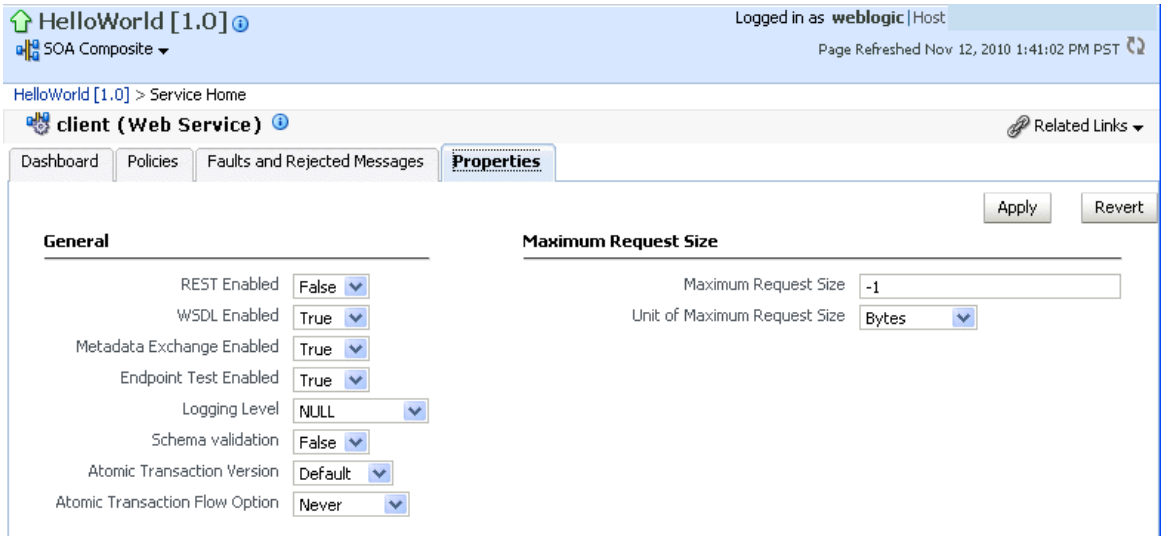
You can configure message header properties for the service and reference binding components included in a deployed SOA composite application.

To configure service and reference binding component properties:

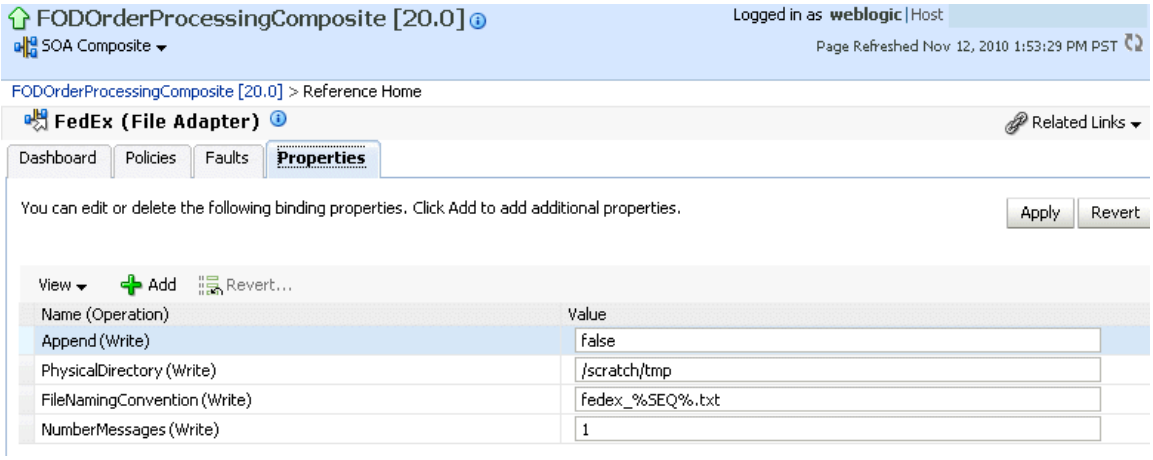
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...	From the SOA Composite Menu...
1. Select **Services and References**. 2. Select a specific service or reference. 3. Click the **Properties** tab.	1. Right-click **soa-infra**. 2. Select **Services and References**. 3. Select a specific service or reference. 4. Click the **Properties** tab.	1. Select **Services/Reference Properties**. 2. Select a specific service or reference. 3. Click the **Properties** tab.

The following binding component properties appear for a web service.



The following binding component properties appear for a file adapter. Depending upon your selection of JCA adapter, different parameters display for configuring.



2. Change properties based on your selection of binding component. See sections Section 34.1.1, "Configuring Properties for Web Services" and Section 34.1.2, "Configuring Properties for Oracle JCA Adapters" for available properties.

Note: To see adapter header properties and their values on this page, ensure that you change the value of the **Audit Level** property from **Production** (the default) to **Development** in the SOA Infrastructure Common Properties page. If this property is set to **Production**, the properties and their values do not display.

34.1.1 Configuring Properties for Web Services

Table 34–1 describes the properties available for a web service binding component.

Table 34–1 Web Service Properties

Service, Reference, or Endpoint	Property Name	Description
Service	**REST Enabled**	Enable or disable the web services port to accept messages in Representational State Transfer (REST) format.
Service	**WSDL Enabled**	Enable or disable the WSDL of the web service.
Service	**Metadata Exchange Enabled**	Enable or disable a metadata exchange of the web service.
Service	**Endpoint Test Enabled**	Enable or disable an endpoint test of the web service.
Service	**Logging Level**	Select the level of logging to perform on the web service.
Service	**Schema Validation**	Enable or disable schema validation.
Service	**Atomic Transaction Version**	Select the WS-AtomicTransaction (WS-AT) supported version (1.0, 1,1, 1,2, or default).
Service	**Atomic Transaction Flow Option**	Select the transaction participation value: ■ **Never** No transaction context is imported (for services) or exported (for references). ■ **Supports** If a transaction exists, a transaction context is imported (for services) or exported (for references). This information is added to the `composite.xml` file. ■ **Mandatory** A transaction context is imported (for services) or exported (for references). For exports, a web service exception message is thrown if there is no active transaction. For imports, a fault is returned to the client if there is no transaction context in the request. For more information, see Section "WS-AtomicTransaction Support" of *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.
Service	**Maximum Request Size**	Enter the maximum request size of the web service. A value of `-1` indicates the size is unlimited.
Service	**Unit of Maximum Request Size**	Select the request unit of the web service (either bytes, kilobytes, megabytes, or gigabytes).

34.1.2 Configuring Properties for Oracle JCA Adapters

If you manually add a nonregistered JCA binding level property in the `composite.xml` file, then you cannot subsequently edit that property or any other registered properties for that service or reference through Oracle Enterprise Manager Fusion Middleware Control. Table 34–2 describes the JCA adapter property types.

Table 34–2 Types of JCA Adapters

Property Type	Description	Restrictions
Activation specification and interaction specification	Activation specification properties operate as services and interaction specification properties operate as references in a SOA composite application.	Do *not* add or remove these properties. You can only change their values. These properties require the adapter endpoint to be recycled. These types of properties are also dependent upon other properties. If you attempt to add one of these properties, you have no way of knowing which dependent properties must also be added.
Endpoint	These are tuning-related properties that are not exposed through the activation or interaction specification properties, such as specifying time outs, thresholds, maximum intervals, and so on.	There are no restrictions on adding, removing, or changing endpoint properties. The adapter is notified when these properties are added, removed, or changed, but it does not require redeployment. You cannot add or remove `jca.retry.*` endpoint properties without redeploying the composite. However, you can change these properties by using Oracle Enterprise Manager Fusion Middleware Control without redeploying the composite.

This section includes the following topics:

- Section 34.1.2.1, "Oracle AQ Adapter"
- Section 34.1.2.2, "Oracle Database Adapter"
- Section 34.1.2.3, "Oracle File Adapter"
- Section 34.1.2.4, "Oracle FTP Adapter"
- Section 34.1.2.5, "Oracle JMS Adapter"
- Section 34.1.2.6, "Oracle MQ Series Adapter"
- Section 34.1.2.7, "Oracle Socket Adapter"
- Section 34.1.2.8, "Oracle JCA Adapters Endpoint Properties"

34.1.2.1 Oracle AQ Adapter

Table 34–3 describes the properties available for the Oracle AQ Adapter.

Table 34–3 Oracle AQ Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`DequeueTimeOut`	Specifies the dequeue timeout interval.
Service	`ConnectionRetryDelay`	Specifies the connection retry interval.
Service	`adapter.aq.dequeue.threads`	Specifies the number of dequeue threads.
Service	`jca.retry.count`	Specifies the number of retries to post the message.

Table 34–3 (Cont.) Oracle AQ Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`jca.retry.interval`	Specifies the time interval between message retries.
Service	`QueueName`	Specifies the AQ queue name.
Service	`DatabaseSchema`	Specifies the database schema name that hosts the queue.
Service	`SchemaValidation`	Validates the message payload.
Service	`ObjectFieldName`	Specifies the ADT entry used as a payload. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Service	`PayloadHeaderRequired`	Specifies if the ADT entries, except for the payload, are accessible.
Service	`DequeueCondition`	Specifies the expression to dequeue messages.
Service	`Consumer`	Specifies the consumer of a queue.
Service	`MessageSelectorRule`	Specifies the message selector to dequeue messages.
Service	`StreamPayload`	Specifies if the payload is streamed or not.
Service	`Correlation`	Specifies a correlation criterion or search criterion for the dequeue operation. For the enqueue operation, the value becomes the correlation of the message sent to AQ.
Reference	`Username`	Identifies a user name requesting access to a secure environment or program, such as an Oracle database or Oracle applications system.
Reference	`Responsibility`	Specifies a collection of functions within Oracle E-Business Suite. Each user is assigned one or more responsibilities to allow access to the appropriate functions and data.
Reference	`QueueName`	Indicates an AQ queue name.
Reference	`DatabaseSchema`	Indicates the database schema name that hosts the queue.
Reference	`ObjectFieldName`	Specifies the ADT entry to use as a payload. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Reference	`PayloadHeaderRequired`	Specifies if ADT entries, except for the payload, are accessible.
Reference	`RecipientList`	Specifies the recipient of the messages.
Reference	`Correlation`	Specifies a correlation criterion or search criterion for the dequeue operation. For the enqueue operation, the value becomes the correlation of the message sent to AQ.

34.1.2.2 Oracle Database Adapter

Table 34–4 describes the properties available for the Oracle Database adapter.

Table 34–4 Oracle Database Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`DescriptorName`	The key into the TopLink metadata indicating the root relational table. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Service	`QueryName`	The key into the TopLink metadata indicating the named query to execute, with a predefined `where` clause and parameters. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Service	`MappingsMetaDataUR`	In conjunction with `MappingsMetaDataURL`, the path to the TopLink metadata file that describes how a database schema has been mapped to a particular predefined XML schema. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Service	`MappingsMetaDataUR`	Indicates the path to the TopLink metadata file that describes how a database schema has been mapped to an XML schema. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Service	`OXMappingsMetaDataURL`	In conjunction with `MappingsMetaDataURL`, the path to the TopLink metadata file that describes how a database schema has been mapped to a particular predefined XML schema. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Service	`PollingInterval`	Indicates the number of seconds between queries to the database for new events.
Service	`MaxRaiseSize`	Indicates the maximum number of XML records that can be raised at a time to Oracle BPEL Process Manager or the activation listener.
Service	`PollingStrategy`	Indicates the method in which events are raised from the database. This is also known as `AfterReadStrategy`. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Service	`MarkReadColumn`	Indicates the status column used by `LogicalDeletePollingStrategy` for indicating which rows have been processed.
Service	`MarkUnreadValue`	The status of unprocessed rows.
Service	`MarkReservedValue`	The status of rows reserved by this polling instance.
Service	`MarkReadValue`	The status of processed rows.
Service	`SequencingTableName`	For example, `DB_ADAPTER_SEQUENCING`.
Service	`SequencingTableKeyColumn`	For example, `TABLE_NAME`. This column holds the primary keys of the various sequences stored.

Table 34–4 (Cont.) Oracle Database Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`SequencingTableValueColumn`	For example, `LAST_READ_ID`. This is the column that holds the value of the last processed sequence value.
Service	`SequencingColumn`	This is the monotonically increasing column on the polled table. For example, `SCOTT.EMP.EMP_ID`.
Service	`SequencingColumnType`	This is the type of sequence counter used. For example, the `java.math.BigDecimal`, `java.sql.Timestamp`.
Service	`SequencingTableKey`	For example, the `SCOTT.EMP`. This is the key into the sequencing helper table, usually the name of the table being polled.
Service	`ShouldCacheSequenceValue`	For `SequencingPollingStrategy`, the `lastReadId` can be kept in-memory so it does not have to be read at the start of each polling interval.
Service	`ShouldWriteSequenceValue`	For `SequencingPollingStrategy`, the `lastReadId` can be kept in-memory and only written out to disk or a database when the process, application server, or both are shut down.
Service	`DeleteDetailRows`	For the delete polling strategy, this indicates whether to delete detail rows and master rows after XML records have been read.
Service	`SequencingDataSourceName`	For `ExternalSequencingPollingStrategy`.
Service	`SequencingIsXADataSource`	For `ExternalSequencingPollingStrategy`.
Service	`UseBatchDestroy`	Deletes or updates multiple processed rows with a single Data Manipulation Language (DML) statement.
Service	`DelayCommit`	Indicates whether to delay the commit/destroy action until all rows picked up (across multiple `maxTransactionSize` units) have been processed.
Service	`PollForChildUpdates`	Indicates whether an update to a detail row must result in the master row and all its details being picked up for processing.
Service	`ShouldOrderRows`	For `SequencingPollingStrategy`, there may be cases in which you do not want to order the rows by the sequence value.
Service	`NumberOfThreads`	Indicates the number of transactional threads to be used by the database adapter.
Service	`FetchSize`	Indicates the JDBC level cursor fetch size.
Service	`MaxBatchWritingSize`	Indicates the JDBC level statements per batch statement.
Service	`DeleteFromCursor`	Indicates the JDBC delete level while iterating over the result set.
Service	`UseDirectSQL`	Enables performance optimization in certain cases.
Service	`ReturnSingleResultSet`	Reads from multiple joined tables in a single SQL `select` statement rather than many, and returns a single giant result set.

Table 34–4 (Cont.) Oracle Database Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`MaxTransactionSize`	The maximum number of rows to raise as part of one database transaction.
Service	`SequencingFileName`	For `FileSequencingPollingStrategy` only. The file contains a single value, the last read sequence value/updated time. Timestamps appear in ISO format.
Service	`SchemaValidation`	Validates the XML against the XML schema before raising it. This can only happen if the automatically generated file has been hand-edited (that is, to add restrictions). Invalid records are rejected.
Service	`EnableStreaming`	Allows LOBs and multirecord XML files to be streamed through SAX events, allowing only a small part of the payload to be kept in memory at a time.
Reference	`DescriptorName`	The key into the TopLink metadata indicating the root relational table. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Reference	`OutputCompletedXml`	Indicates if this interaction execution have an output message that is the input message with primary keys set.
Reference	`MappingsMetaDataURL`	The path to the `toplink.xml` file that describes how a database schema has been mapped to an XML schema. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Reference	`OXMappingsMetaDataURL`	In conjunction with `MappingsMetaDataURL`, the path to the TopLink metadata file that describes how a database schema has been mapped to a particular predefined XML schema. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Reference	`GetActiveUnitOfWork`	Operations within the same JTA transaction use the same TopLink session and connection, and writes occur together on a JTA commit.
Reference	`DmlType`	Indicates the type of DML operation (merge, insert, update, delete, or write).
Reference	`OutputCompletedXml`	Does this interaction execute have an output message that is the input message with primary keys set? This is useful when assigning primary keys on an insert.
Reference	`OptimizeMerge`	Sets some properties to improve performance of the merge. For one, if multiple objects are merged, the read queries to check for the existence and loads database versions into memory. Comparisons are performed as a single query, using an `in()` clause on the primary key.
Reference	`DetectOmissions`	Indicates whether to differentiate between null and "not there" in the input XML. The difference between `<director />` (omission) and `<director xsi:nil=\"true\"/>` (null).

Table 34–4 (Cont.) Oracle Database Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Reference	`MaxBatchWritingSize`	The JDBC level statements per batch statement.
Reference	`UseDirectSQL`	Enables performance optimization in certain cases.
Reference	`QueryName`	The key into the TopLink metadata indicating the named query to execute, with predefined `where` clause and parameters.
Reference	`IsQueryByExample`	The query is generated dynamically on each invoke, based on matching the example input XML record.
Reference	`ReturnSingleResultSet`	An advanced feature that influences how many total select statements TopLink uses when querying against multiple related tables (that is, master-detail). The safest setting is the default (one per table); setting it to `true` than attempts one total, by outer joining all related selects into a single result set.
Reference	`CursorWindowSize`	The number of records to return at a time when selecting a very large number of rows in a single query.
Reference	`EnableStreaming`	Allows LOBs and multirecord XMLs to be streamed through SAX events, allowing only a small part of the payload to be kept in memory at a time.
Reference	`SqlString`	The SQL to execute in the pure SQL interaction. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Reference	`IsTransactional`	Indicates whether the pure SQL being executed is a write statement that must occur within transaction boundaries.
Reference	`QueryTimeout`	Indicates the JDBC level `queryTimeout` setting. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Reference	`MaxRows`	Indicates the JDBC level `maxRows` setting. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Reference	`SchemaName`	The schema of the stored procedure to execute.
Reference	`PackageName`	The package of the stored procedure to execute.
Reference	`ProcedureName`	The procedure of the stored procedure to execute.
Reference	`Overload`	Further identifies the stored procedure to execute (in case of overloading) by the expected number of parameters.
Reference	`QueryTimeout`	Indicates the JDBC level `queryTimeout` setting.

34.1.2.3 Oracle File Adapter

Table 34–5 describes the properties available for the Oracle File adapter.

Table 34–5 Oracle File Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`PhysicalDirectory`	Specifies the physical directory for the file/FTP adapter.
Service	`UseHeaders`	Set to `true` if the file/FTP adapter must publish metadata (for example, the file name, directory name, last modified data, file size, and to exclude the payload). This is typically used in large payload scenarios where you want to use the inbound adapter as a notifier.
Service	`Recursive`	If set to `true`, the file/FTP adapter processes files recursively in subdirectories.
Service	`PhysicalArchiveDirectory`	Directory in which the inbound file/FTP adapter archives files after they have been processed successfully.
Service	`PhysicalErrorArchiveDirectory`	Indicates the directory where the inbound file/FTP adapter archives files in case of failures, such as translation errors, and errors during publishing.
Service	`DeleteFile`	If set to `true`, the file/FTP adapter deletes the file after it has been processed.
Service	`IncludeFiles`	Indicates the `regex` pattern against which the file names are matched. If the name of a certain file matches this pattern, it is processed by the file/FTP adapter.
Service	`ExcludeFiles`	Indicates the `regex` pattern against which the file names are matched. If the name of a certain file matches this pattern, it is excluded and not processed by the file/FTP adapter.
Service	`PollingFrequency`	This parameter specifies how often does the file/FTP adapter looks for files in the inbound directory. It is specified in seconds.
Service	`MinimumAge`	This parameter specifies the time interval after which to pick up a file for processing. For example, this enables a large file to be completely copied into the directory before it is retrieved for processing. The age is determined by the last modified time stamp. For example, if you know that it takes three to four minutes for a file to be written, set the minimum age of pollable files to five minutes. If a file is detected in the input directory and its modification time is less than `5` minutes older than the current time, the file is not retrieved because it is still potentially being written to.
Service	`PublishSize`	This parameter indicates that a file contains multiple messages and specifies how many messages to process simultaneously. For example, if a certain file has `11` records and this parameter is set to `2`, then the file is processed `2` records at a time and the final record is processed in the sixth iteration.

Table 34–5 (Cont.) Oracle File Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`Lenient`	If set to `true`, then the file adapter does not complain if it does not have enough permission to read/write to the inbound directory. By default, this is set to `false`.
Service	`TriggerFilePhysicalDirectory`	The directory path where the file/FTP adapter looks for the trigger files.
Service	`TriggerFile`	The name of the trigger file that causes the inbound file/FTP adapter to be activated.
Service	`TriggerFileStrategy`	This parameter defines the strategy that the file/FTP adapter uses to look for the specified trigger file in the trigger file directory. The acceptable values are `EndpointActivation`, `EveryTime`, or `OnceOnly`.
Service	`MaxRaiseSize`	This parameter specifies the maximum number of files that the file/FTP adapter submits for processing in each polling cycle. For example, if the inbound directory has `1000` files and this parameter is set to 100 and the polling frequency is one minute, then the file/FTP adapter submits `100` files every minute.
Service	`DirectorySeparator`	When you choose multiple directories, the generated JCA files use `\";\"` as the separator for these directories. If you want, you can change the separator. If you do change it, manually add `DirectorySeparator=\"<<chosen separator>>\"` in the generated JCA file. For example, to use a comma (`,`) as the separator, you must first change the separator to `\",\"` in the physical directory and then add `DirectorySeparator=\",\"` in the JCA file.
Service	`AsAttachment`	If set to `true`, it causes the inbound file to be published as an attachment.
Service	`CharacterSet`	Set it to the character set for the attachment. This parameter is not used internally by the file/FTP adapter and is meant for third party applications that process the attachments published by the file/FTP adapter.
Service	`Encoding`	Set it to the encoding for the attachment. This parameter is not used internally by the file/FTP adapter and is meant for third party applications that process the attachments published by the file/FTP adapter.
Service	`ContentType`	Set it to the `Mime-Type` for the attachment. This parameter is not used internally by the file/FTP adapter and is meant for third party applications that process the attachments published by the file/FTP adapter.

Table 34–5 (Cont.) Oracle File Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`ListSorter`	Specifies the sorter that the file/FTP adapter uses to sort files in the inbound direction. You can set this parameter to `\"oracle.tip.adapter.file.inbound.listing.TimestampSorterAscending\"` or `\"oracle.tip.adapter.file.inbound.listing.TimestampSorterDescending\"`. You can also plug in your own sorter by writing a class that implements `\"java.util.Comparator\"`.
Service	`SingleThreadModel`	If the value is `true`, the file/FTP adapter poller processes files in the same thread; it does not use the global in-memory queue for processing.
Service	`ThreadCount`	If this parameter is available, the adapter creates its own processor threads rather than depending on the global thread pool processor threads (by default, four of them). This parameter partitions the in-memory queue and each composite application gets its own in-memory queue. If the `ThreadCount` is set to `\"0\"`, then it behaves in the same manner as the single thread model. If `ThreadCount` is set to `\"-1\"`, then it starts using the global thread pool. The maximum value for this property is `40`.
Service	`recoveryInterval`	Used by the inbound adapter to configure the recovery interval in case of errors. For example, if the physical directory is nonexistent, then the adapter uses this value to perform periodic sleep/wake ups to check if the physical directory has been created and is accessible.
Reference	`PhysicalDirectory`	Specifies the physical directory for the file/FTP adapter.
Reference	`NumberMessages`	Specifies the outbound file naming convention used.
Reference	`ElapsedTime`	This parameter is used for outbound batching. When the time specified elapses, the outgoing file is created. The parameter is of type `String` and is not mandatory. The default value is `1`.
Reference	`FileSize`	Indicates if an opaque schema is being used.
Reference	`FileNamingConvention`	This parameter is for the naming convention for the outbound write operation file.
Reference	`FileName`	Use this parameter to specify a static single file name during the write operation.
Reference	`Append`	Setting this parameter to `\"true\"` causes the file/FTP adapter to append to a file in the outbound direction. If the file does not exist, a new file is created.
Reference	`UseStaging`	If `\"true\"`, then the outbound file/FTP adapter writes translated data to a staging file and afterward streams the staging file to the target file. If `\"false\"`, then the outbound file/FTP adapter does not use an intermediate staging file.

Table 34–5 (Cont.) Oracle File Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Reference	`ConcurrentThreshold`	The maximum number of translation activities that can be allowed to execute in parallel for a particular outbound scenario. The translation step during the outbound operation is CPU intensive and hence must to be guarded as it may cause starvation in other applications/threads. The maximum is `100` (same as the maximum for `dspMaxThreads` in BPEL).
Reference	`SequenceName`	Specifies the Oracle database sequence name to be used if you have already configured the outbound file/FTP adapter for high availability.
Reference	`oracle.tip.adapter.file.mutex`	Set it to the class name that specifies the mutex you want to use for the outbound write operation. This class must extend the "`oracle.tip.adapter.file.Mutex`" abstraction.
Reference	`serializeTranslation`	If set to `\"true\"`, then the translation step is serialized using a semaphore. The number of permits for semaphore (guarding the translation step) comes from the `ConcurrentThreshold` parameter. If `\"false\"`, then the translation step occurs outside the semaphore.
Reference	`inMemoryTranslation`	This parameter is applicable only if `UseStaging` is `\"false\"`. If set to `\"true\"`, then the translation step occurs in-memory (that is, an in-memory byte array is created). If set to `\"false\"`, then the adapter creates an output stream to the target file (FTP, FTPS, and SFTP included) and allows the translator to translate and write directly to the stream.
Reference	`DeleteFile`	If set to `\"true\"`, the file/FTP adapter deletes the file after it has been processed.
Reference	`IgnoreZeroByteFile`	Set it to `true` if you do not want the file/FTP adapter to throw an exception during the outbound read operation if the file was not found. This parameter is ignored if the schema for the inbound file is anything other than opaque.
Reference	`IncludeFiles`	Indicates the `regex` pattern against which the file names are matched. If the name of a certain file matches this pattern, it is processed by the file/FTP adapter.
Reference	`ExcludeFiles`	Includes the `regex` pattern against which the file names are matched. If the name of a certain file matches this pattern, it is excluded and not processed by the file/FTP adapter.
Reference	`Recursive`	If set to `true`, the file/FTP adapter processes files recursively in subdirectories.
Reference	`MaxRaiseSize`	This parameter specifies the maximum number of files that the file/FTP adapter submits for processing in each polling cycle. For example, if the inbound directory has `1000` files and this parameter is set to `100` and the polling frequency is one minute, then the file/FTP adapter submits `100` files every minute.

Table 34–5 (Cont.) Oracle File Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Reference	`DirectorySeparator`	When you choose multiple directories, the generated JCA files use `\";\"` as the separator for these directories. If you want, you can change the separator. If you do so, manually add `DirectorySeparator=\"<<chosen separator>>\"` in the generated JCA file. For example, to use a comma (`,`) as the separator, you must first change the separator to `\",\"` in the physical directory and then add `DirectorySeparator=\",\"` in the JCA file.
Reference	`SourceFileName`	The source file for the file I/O operation.
Reference	`SourcePhysicalDirectory`	The source directory for the file I/O operation.
Reference	`SourceType`	Set this to `\"native\"` if the source file is native and to `\"xml\"` if the source file is XML.
Reference	`SourceSchema`	Set it to the schema for the source file.
Reference	`SourceSchemaRoot`	Set it to the root element name for the source file.
Reference	`TargetFileName`	Indicates the target file for the file I/O operation.
Reference	`TargetPhysicalDirectory`	Indicates the target directory for the file I/O operation.
Reference	`TargetType`	Set this to `\"native\"` if the target file is native and to `\"xml\"` if the source file is XML.
Reference	`TargetSchema`	Set it to the schema for the target file.
Reference	`TargetSchemaRoot`	Set it to the root element name for the target file.
Reference	`Xsl`	Set it to the XSL transformer between the source and target.
Reference	`Type`	Set it to `\"COPY\"`,`\"MOVE\"`, or `\"DELETE\"`.
Reference	`BatchSize`	Set it to the batch size for the batching transformation.
Reference	`ChunkSize`	Specifies the number of outbound messages.

34.1.2.4 Oracle FTP Adapter

Table 34–6 describes the properties available for the Oracle FTP adapter.

Table 34–6 Oracle FTP Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`timestampOffset`	This parameter is used by the FTP adapter to tackle time zone issues (typically, to convert the time difference between the FTP server and the system on which the FTP adapter is running to milliseconds).
Service	`PhysicalDirectory`	This parameter specifies the physical directory for the file/FTP adapter.
Service	`UseHeaders`	Set to `\"true\"` if the file/FTP adapter must publish metadata (for example, the file name, directory name, last modified data, file size, and to exclude the payload. This is typically used in large payload scenarios where you want to use the inbound adapter as a notifier.
Service	`FileType`	Set it to either `\"ascii\"` or `\"binary\"`, depending on the requirement.
Service	`Recursive`	If set to `\"true\"`, the file/FTP adapter processes files recursively in subdirectories.
Service	`PhysicalArchiveDirectory`	Directory in which the inbound file/FTP adapter archives files after they have been processed successfully.
Service	`PhysicalErrorArchiveDirectory`	Indicates the directory where the inbound file/FTP adapter archives files in case of failures, such as translation errors, and errors during publishing.
Service	`UseRemoteArchive`	Set this parameter to `\"true\"` to notify the FTP adapter that the archival directory is on the same FTP server. If set to `\"false\"`, the FTP adapter uses a local file system folder for archival.
Service	`UseNlst`	Set this parameter to `\"true\"` if you need the FTP adapter to use the `\"NLST\"` FTP command instead of the `\"LIST\"` that the adapter uses by default.
Service	`DeleteFile`	If set to `\"true\"`, the file/FTP adapter deletes the file after it has been processed.
Service	`IncludeFiles`	Indicates the `regex` pattern against which the file names are matched. If the name of a certain file matches this pattern, it is processed by the file/FTP adapter.
Service	`ExcludeFiles`	Indicates the `regex` pattern against which the file names are matched. If the name of a certain file matches this pattern, it is excluded and not processed by the file/FTP adapter.
Service	`PollingFrequency`	This parameter specifies how often the file/FTP adapter checks for files in the inbound directory. It is specified in seconds.

Table 34–6 (Cont.) Oracle FTP Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`MinimumAge`	This parameter specifies the time interval after which to pick up a file for processing. For example, this enables a large file to be completely copied into the directory before it is retrieved for processing. The age is determined by the last modified time stamp. For example, if you know that it takes three to four minutes for a file to be written, set the minimum age of pollable files to five minutes. If a file is detected in the input directory and its modification time is less than `5` minutes older than the current time, the file is not retrieved because it is still potentially being written to.
Service	`PublishSize`	This parameter indicates that a file contains multiple messages and specifies how many messages should be processed simultaneously. For example, if a certain file has `11` records and this parameter is set to `2`, then the file is processed `2` records at a time and the final record is processed in the sixth iteration.
Service	`TriggerFilePhysicalDirectory`	The directory path in which the file/FTP adapter looks for the trigger files.
Service	`TriggerFile`	The name of the trigger file that causes the inbound file/FTP adapter to activate.
Service	`TriggerFileStrategy`	This parameter defines the strategy that the file/FTP adapter uses to look for the specified trigger file in the trigger file directory. The acceptable values are `EndpointActivation`, `EveryTime`, or `OnceOnly`.
Service	`MaxRaiseSize`	This parameter specifies the maximum number of files that the file/FTP adapter submits for processing in each polling cycle. For example, if the inbound directory has `1000` files and this parameter is set to `100` and the polling frequency is one minute, then the file/FTP adapter submits `100` files every minute.
Service	`DirectorySeparator`	When you choose multiple directories, the generated JCA files use `\";\"` as the separator for these directories. If you want, you can change the separator. If you do, manually add `DirectorySeparator=\"<<chosen separator>>\"` in the generated JCA file. For example, to use a comma (`,`) as the separator, you must first change the separator to `\",\"` in the physical directory and then add `DirectorySeparator=\",\"` in the JCA file.
Service	`AsAttachment`	If set to `\"true\"`, it causes the inbound file to be published as an attachment.
Service	`CharacterSet`	Set it to the character set for the attachment. This parameter is not used internally by the file/FTP adapter and is meant for third party applications that process the attachments published by the file/FTP adapter.

Table 34–6 (Cont.) Oracle FTP Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`Encoding`	Set it to the encoding for the attachment. This parameter is not used internally by the file/FTP adapter and is meant for third party applications that process the attachments published by the file/FTP adapter.
Service	`ContentType`	Set it to the `Mime-Type` for the attachment. This parameter is not used internally by the file/FTP adapter and is meant for third party applications that process the attachments published by the file/FTP adapter.
Service	`ListSorter`	Specifies the sorter that the file/FTP adapter uses to sort files in the inbound direction. You can set this parameter to `\"oracle.tip.adapter.file.inbound.listing.TimestampSorterAscending\"`, or `\"oracle.tip.adapter.file.inbound.listing.TimestampSorterDescending\"`. You can also plug in your own sorter by writing a class that implements `\"java.util.Comparator\"`.
Service	`SingleThreadModel`	If the value is `true`, the file/FTP adapter poller processes files in the same thread; it does not use the global in-memory queue for processing.
Service	`ThreadCount`	If this parameter is available, the adapter creates its own processor threads rather than depending on the global thread pool processor threads (by default, `4` of them). In other words, this parameter partitions the in-memory queue and each composite application gets its own in-memory queue. If the `ThreadCount` is set to `\"0\"`, then it behaves in the same manner as the `SingleThreadModel`. If the `ThreadCount` is set to `\"-1\"`, then it starts using the global thread pool. The maximum value for this property is `40`.
Service	`recoveryInterval`	Used by the inbound adapter to configure the `recoveryInterval` in case of errors. For example, if the `PhysicalDirectory` is nonexistent, then the adapter uses this value to perform periodic sleep/wake ups to check if the `PhysicalDirectory` has been created and is accessible.
Service	`jca.message.encoding`	This parameter is used to override the encoding specified in the NXSD schema for the inbound file/FTP adapter.
Service	`oracle.tip.adapter.file.`debatching.`rejection.quantum`	This property lets you control the size of a rejected message for the inbound file/FTP adapter partner link. For example, if you set it to `\"100\"`, it causes the file/FTP adapter to reject `100` lines from the file since the actual file is too large.
Service	`useFileSystem`	This parameter is used by the inbound file/FTP adapter during read-only polling in a clustered environment. Setting it to `\"true\"` causes the adapter to use the file system to store metadata about files that have been already processed. Setting it to `\"false\"` causes the adapter to use a database table.

Table 34–6 (Cont.) Oracle FTP Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`oracle.tip.adapter.file.timeout.recover picked.minutes`	Used by the inbound high-availability adapter when using `\"FILEADAPTER_IN\"` as the coordinator. When a file is first claimed (enqueued) by a node for processing, the `FILE_PROCESSED` column in `FILEADAPTER_IN` is set to `\"0\"`. At a later point in time, when one of the decoupled processor threads picks up the file for processing, the value of `FILE_PROCESSED` column is updated from `\"0\"` to `\"1\"`. When the file is processed completely, the `FILE_PROCESSED` column is updated from `\"1\"` to `\"2\"`. However, if the processor thread picks up a file, but the node crashes before it can process the file, then the file is never processed. This parameter is used to `\"undo\"` the pick operation. The adapter does this by deleting the entries in the `FILEADAPTER_IN` table that have been picked up, but not processed within the value specified here.
Service	`oracle.tip.adapter.file.timeout.recoverunpicked.minutes`	Used by the inbound high-availability adapter when using `\"FILEADAPTER_IN\"` as the coordinator. When a file is first claimed by a node for processing, the `FILE_PROCESSED` column in `FILEADAPTER_IN` is set to `\"0\"`. At a later point in time, when the decoupled-processor thread picks up the file for processing, the value of the `FILE_PROCESSED` column is updated from `\"0\"` to `\"1\"`. When the file is processed completely, the `FILE_PROCESSED` column is updated from `\"1\"` to `\"2\"`. If the node crashes when the `FILE_PROCESSED` is still `\"0\"`, the file is enqueued by a node (this means no other nodes can pick this one up). However, it also means that the decoupled processor threads have still not picked this one for processing. This parameter is used to `\"undo\"` the `claim(enqueue_` operation. The adapter does this by deleting the entries in the `FILEADAPTER_IN` table that have been claimed (for example, `FILE_PROCESSED ==\"0\"`), but not picked up until now.
Service	`purgeIntervalMillis`	Defines how often the poller thread purges control files for read-only polling scenarios. This defaults to `5` days.
Service	`oracle.tip.adapter.file.highavailability.maxRetryInterval`	The number of milliseconds after which the inbound file/FTP adapter retries to establish a database connection in distributed polling scenarios.
Service	`oracle.tip.adapter.file.highavailability.maxRetry`	The number of times that the inbound file/FTP adapter retries to establish a database connection in distributed polling scenarios.
Service	`oracle.tip.adapter.file.rejectOriginalContent`	Setting to `\"true\"` causes the file/FTP adapter to reject the original actual content. If set to `\"false\"`, the adapter rejects the XML data created because of the translation step.

Table 34–6 (Cont.) Oracle FTP Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`notifyEachBatchFailure`	Setting to `\"true\"` causes the file/FTP adapter to call the notification agent's `onBatchFailure` every time an error occurs in a debatching scenario. If set to `\"false\"`, the file/FTP adapter calls `onBatchFailure` only once after all messages in the debatching scenario.
Reference	`PhysicalDirectory`	The directory path for the file/FTP adapter.
Reference	`FileType`	Set it to either `\"ascii\"` or `\"binary\"`, depending on the requirement.
Reference	`NumberMessages`	This parameter is used for outbound batching. The outgoing file is created when the number of messages condition is met. The parameter is of type `String` and is not mandatory. The default value is `1`.
Reference	`ElapsedTime`	This parameter is used for outbound batching. When the time specified elapses, the outgoing file is created. The parameter is of type `String` and is not mandatory. The default value is `1`.
Reference	`FileSize`	This parameter is used for outbound batching. The outgoing file is created when the file size condition is met. The parameter is of type String and is not mandatory. The default value is `1000` KB.
Reference	`FileNamingConvention`	This parameter is for the naming convention of the outbound write operation file.
Reference	`FileName`	Use this parameter to specify a static single file name during the write operation.
Reference	`Append`	Setting this parameter to `\"true\"` causes the file/FTP adapter to append to a file in the outbound direction. If the file does not exist, a new file is created.
Reference	`UseStaging`	If `\"true\"`, then the outbound file/FTP adapter writes translated data to a staging file and afterward streams the staging file to the target file. If `\"false\"`, then the outbound file/FTP adapter does not use an intermediate staging file.
Reference	`ConcurrentThreshold`	The maximum number of translation activities allowed to execute in parallel for a particular outbound scenario. The translation step during the outbound operation is CPU intensive and must to be guarded as it may cause starvation in other applications/threads. The maximum value is `100` (same as the maximum value for `dspMaxThreads` in BPEL).
Reference	`SequenceName`	Specifies the Oracle database sequence name to use if you have already configured the outbound file/FTP adapter for high availability.
Reference	`oracle.tip.adapter.file.mutex`	Set it to the class name that specifies the mutex you want to use for the outbound write operation. This class must extend the `\"oracle.tip.adapter.file.Mutex\"` abstraction.

Table 34–6 (Cont.) Oracle FTP Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Reference	serializeTranslation	If set to \"true\", then the translation step is serialized using a semaphore. The number of permits for semaphore (guarding the translation step) comes from the ConcurrentThreshold parameter. If \"false\", then the translation step occurs outside the semaphore.
Reference	inMemoryTranslation	This parameter is applicable only if UseStaging is \"false\". If set to \"true\", then the translation step occurs in-memory (that is, an in-memory byte array is created). If set to \"false\", then the adapter creates an output stream to the target file (FTP, FTPS, and SFTP included) and allows the translator to translate and write directly to the stream.
Reference	FileType	Set it to either \"ascii\" or \"binary\", depending on the requirement.
Reference	UseRemoteArchive	Set this parameter to \"true\" to notify the FTP adapter that the archival directory is on the same FTP server. If set to \"false\", the FTP adapter uses a local file system folder for archival.
Reference	DeleteFile	If set to \"true\", the file/FTP adapter deletes the file after it has been processed.
Reference	IgnoreZeroByteFile	Set it to \"true\" if you do not want the file/FTP adapter to throw an exception during the outbound read operation if the file was not found. This parameter is ignored if the schema for the inbound file is anything other than \"Opaque\".
Reference	IncludeFiles	Indicates the regex pattern against which the file names are matched. If the name of a certain file matches this pattern, it is processed by the file/FTP adapter.
Reference	ExcludeFiles	Indicates the regex pattern against which the file names are matched. If the name of a certain file matches this pattern, it is excluded and not processed by the file/FTP adapter.
Reference	Recursive	If set to \"true\", the file/FTP adapter processes files recursively in subdirectories.
Reference	MaxRaiseSize	This parameter specifies the maximum number of files that the file/FTP adapter submits for processing in each polling cycle. For example, if the inbound directory has 1000 files and this parameter is set to 100 and the polling frequency is one minute, the file/FTP adapter submits 100 files every minute.
Reference	DirectorySeparator	When you choose multiple directories, the generated JCA files use \";\" as the separator for these directories. If you want, you can change the separator. If you do, manually add DirectorySeparator=\"<<chosen separator>>\" in the generated JCA file. For example, to use a comma (,) as the separator, you must first change the separator to \",\" in the physical directory and then add DirectorySeparator=\",\" in the JCA file.

Table 34–6 (Cont.) Oracle FTP Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Reference	`UseNlst`	Set this parameter to `\"true\"` if you need the FTP adapter to use the `\"NLST\"` FTP command instead of `\"LIST\"` that the adapter uses by default.
Reference	`timestampOffset`	This parameter is used by the FTP adapter to tackle time zone issues (typically, to convert the time difference between the FTP server and the system on which the FTP adapter is running to milliseconds).
Reference	`SourceFileName`	Indicates the source file for the file I/O operation.
Reference	`SourcePhysicalDirectory`	Indicates the source directory for the file I/O operation.
Reference	`SourceType`	Set this to `\"native\"` if the source file is native and to `\"xml\"` if the source file is XML.
Reference	`SourceSchema`	Set it to the schema for the source file.
Reference	`SourceSchemaRoot`	Set it to the root element name for the source file.
Reference	`TargetFileName`	Indicates the target file for the file I/O operation.
Reference	`TargetPhysicalDirectory`	The target directory for the file I/O operation.
Reference	`TargetType`	Set this to `\"native\"` if the target file is native and to `\"xml\"` if the source file is XML.
Reference	`TargetSchema`	Set it to the schema for the target file.
Reference	`TargetSchemaRoot`	Set it to the root element name for the target file.
Reference	`Xsl`	Set it to the XSL transformer between the source and target.
Reference	`Type`	Set it to either `\"ascii\"` or `\"binary\"`, depending on the requirement.
Reference	`BatchSize`	Set it to the batch size for the batching transformation.
Reference	`SourceIsRemote`	Set it to `\"false\"` to notify the FTP adapter that the source for the I/O operation is a local file system as opposed to a remote FTP server.
Reference	`TargetIsRemote`	Set it to `\"false\"` to I/O notify the FTP adapter that the target for the I/O operation is a local file system as opposed to a remote FTP server.
Reference	`ChunkSize`	Set it to the `ChunkSize` for the chunked interaction operation.

34.1.2.5 Oracle JMS Adapter

Table 34–7 describes the properties available for the Oracle JMS adapter.

Table 34–7 Oracle JMS Adapter

Service, Reference, or Endpoint	Property Name	Description
Service	`adapter.jms.encoding`	Set the encoding to be used by JMS inbound adapter.
Service	`adapter.jms.receive.timeout`	Set the receive timeout interval.
Service	`adapter.jms.retry.interval`	Set the interval the JMS adapter uses to retry.
Service	`adapter.jms.registration. interval`	Set the registration interval.
Service	`adapter.jms.receive.threads`	Set the number of receive threads.
Service	`JMSReplyTo DestinationProperties`	Set the reply-to destination.
Service	`JMSReplyUse MessageIdForCorrelation`	Specifies whether the message ID is necessary for correlation. Valid values are `true` or `false`.
Service	`JMSReplyUseCorrelation IdForCorrelation`	Specifies whether the correlation ID is necessary for correlation. Valid values are `true` or `false`.
Service	`suppressHeaders`	Specifies whether to bypass headers. Valid values are `true` or `false`.
Service	`JMSReply PropagateJMSExpiration`	Specifies whether a reply message is created with a TTL value of other than `0`, which is the default value in a request-reply scenario. Valid values are `true` or `false`.
Service	`minimumDelayBetweenMessages`	This is a throttling parameter (in milliseconds) that adds a brief sleep period between each inbound posted message, thus slowing down the message inflow.
Reference	`requestReply.useCorrelation`	Specifies whether correlation is necessary in a request-reply scenario. Valid values are `true` or `false`.
Reference	`requestReply.cacheReceivers`	Specifies whether correlation is necessary in a request-reply scenario. Valid values are `true` or `false`.
Reference	`DestinationName`	The name of the JMS physical destination.
Reference	`PayloadType`	Indicates the JMS message type.
Reference	`DeliveryMode`	Sets the JMS delivery mode. Delivery modes supported by JMS API are `PERSISTENT` and `NON_PERSISTENT`.
Reference	`TimeToLive`	Indicates the message lifetime in milliseconds.
Reference	`Priority`	Indicates the JMS message priority.
Reference	`PayloadEntry`	Specifies the `MapMessage` entry to use as the payload.
Reference	`AttachmentList`	Specifies the `MapMessage` entry to use as the payload and publish as an attachment.
Reference	`RequestDestinationName`	Indicates the name of the JMS physical destination.
Reference	`ReplyDestinationName`	Indicates the name of the JMS physical destination.
Reference	`AllowTemporaryReplyDestination`	Allows creation of a temporary destination for receiving messages.

Table 34–7 (Cont.) Oracle JMS Adapter

Service, Reference, or Endpoint	Property Name	Description
Reference	`DurableSubscriber`	Indicates the unique name of the durable subscription within a client identifier.
Reference	`MessageSelector`	The message selector expression used for message selection when delivering a message to the destination.
Reference	`ReplyTimeout`	Indicates the `Timeout` to receive a reply message.

34.1.2.6 Oracle MQ Series Adapter

Table 34–8 describes the properties available for the Oracle MQ Series adapter.

Table 34–8 Oracle MQ Series Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`MessageType`	Indicates the message type of the outbound/dequeue message.
Service	`QueueName`	Indicates the name of the inbound/outbound queue.
Service	`UseMessageEncodingForTranslation`	Set to `true` if the character set `from` header is used during translation.
Service	`InboundThreadCount`	Indicates the number of threads used in the inbound direction.
Service	`BackoutQueueName`	Indicates the name of a backout queue in which the rejected message goes.
Service	`BackoutQueueManagerName`	Indicates the name of the backout queue manager in which the rejected message goes.
Service	`MaximumBackoutCount`	Indicates the number of times the adapter retries before sending the message to a backout queue.
Service	`BackoutInterval`	Indicates the interval between a retry for a backout message.
Service	`jca.message.encoding`	This encoding value overwrites any encoding specified, either in MQMD or in a schema file.
Service	`UseMessageEncodingForTranslation`	Set to `\"true\"` if the `characterSet` field of MQMD is use for translation in the inbound direction. If set to `\"false\"`, the translator uses the encoding from the schema file. You can use the `jca.message.encoding` property to overwrite any encoding specified, either in MQMD or in the schema file.
Service	`FallbackReplyToQueueName`	Indicates the `Fallback` Reply To queue name.
Service	`FallbackReplyToQueueManagerName`	Indicates the `Fallback` Reply To queue manager name.
Service	`Priority`	Indicates the priority of the reply/outbound message.
Service	`Expiry`	Indicates the expiry time for the reply/outbound message.

Table 34–8 (Cont.) Oracle MQ Series Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`Persistence`	Indicates the persistence of the reply/outbound message.
Service	`Feedback`	Indicates the feedback code of the reply/outbound message. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Service	`MessageFormat`	Indicates the format of the reply/outbound message.
Service	`OnDeliveryFailure`	Indicates the behavior of the adapter if reply/outbound message delivery fails.
Service	`SegmentIfRequired`	Indicates the segmentation property for the message.
Service	`CopyPersistenceFromInbound`	Indicates the copy persistence from inbound message to reply message in a synchronous request-response (inbound) scenario.
Service	`PropagateExpiryInterval`	Indicates the propagate expiry interval from the inbound message to the reply message in a synchronous request-response (inbound) scenario.
Service	`QueueName`	Indicates the name of the inbound/outbound queue.
Service	`GetMessageOptions`	Indicates the `Get` message options for an outbound dequeue scenario.
Service	`FilterByMsgId`	Indicates the message ID to use for filtering.
Service	`FilterByCorrelId`	Indicates the correlation ID to use for filtering.
Service	`FilterByGroupId`	Indicates the group ID to use for filtering.
Service	`FilterByMsgSeqNumber`	Indicates the message sequence number to use for filtering.
Service	`FilterByMsgSeqNumber`	Indicates the message sequence number to use for filtering.
Reference	`QueueName`	Indicates the name of the inbound/outbound queue.
Reference	`DistributionList`	Indicates the distribution list in which the message goes.
Reference	`PartialDeliveryForDL`	Set to `false` if the message must go to all queues in the DL. Otherwise, it is set to `true`.
Reference	`SecondaryQueueManagerName`	Indicates the name of the secondary queue manager.
Reference	`MessageFormat`	Indicates the format of the reply/outbound message.
Reference	`ReplyMessageId`	Indicates the correlation scheme for the message ID to be set.
Reference	`ReplyCorrelationId`	Indicates the correlation scheme for the correlation ID to be set.
Reference	`MessageId`	Indicates the correlation scheme for the outbound message.

Table 34–8 (Cont.) Oracle MQ Series Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Reference	`CorrelationId`	Indicates the correlation scheme for the outbound message.
Reference	`ReportCOA`	Indicates the COA report to be set on the message.
Reference	`ReportCOD`	Indicates the COD report to be set on the message.
Reference	`ReportException`	Indicates the exception report to be set on the message.
Reference	`ReportExpiry`	Indicates the expiration report to be set on the message.
Reference	`ReportPAN`	Indicates if `PAN` is required.
Reference	`ReportNAN`	Indicates if `NAN` is required.
Reference	`FallbackReplyToQueueName`	Indicates the `Fallback Reply To` queue name.
Reference	`FallbackReplyToQueueManagerName`	Indicates the `Fallback Reply To` queue manager name.
Reference	`Priority`	Indicates the priority of the reply/outbound message.
Reference	`Expiry`	Indicates the expiry time for the reply/outbound message.
Reference	`Persistence`	Indicates the persistence of the reply/outbound message.
Reference	`Feedback`	Indicates the feedback code of the reply/outbound message.
Reference	`ReplyToQueueName`	Indicates the ReplyToQueue to set on the message. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Reference	`ReplyToQueueManagerName`	Indicates the `ReplyToQueueManager` to be set on the message. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Reference	`OnDeliveryFailure`	Indicates the behavior of the adapter if reply/outbound message delivery fails.
Reference	`SegmentIfRequired`	Indicates the segmentation property for the message.
Reference	`PartialDeliveryForDL`	Set to `true` if partial delivery to the distribution list is allowed. Otherwise, it is set to `false`.
Reference	`SyncSolicitReqRes`	Set to `true` if it is a synchronous request-response (outbound) scenario. Otherwise, it is set to `false`. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Reference	`ResponseOpaqueSchema`	Set to `true` if a reply/report message in a asynchronous request-response scenario has an opaque schema.

Table 34–8 (Cont.) Oracle MQ Series Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Reference	`ResponseWaitInterval`	Indicates the wait time for the reply/report to arrive.
Reference	`ResponseNoMessageAllowed`	Set to `true` if no message is allowed as a reply/report after the specified wait interval.
Reference	`ResponseGetMessageOptions`	Indicates the `Get` message options for a reply/report.
Reference	`ResponseQueueOpenOptions`	Indicates the open options for a reply/report queue.
Reference	`PutMessageOptions`	Indicates the `Put` message options.
Reference	`QueueOpenOptions`	Indicates the open options for the queue. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Reference	`UseMessageEncodingForTranslation`	Set to true if the character set from the header is used during translation.
Reference	`DynamicQueueName`	Indicates the dynamic queue name.
Reference	`AlternateUserID`	Indicates the alternate user ID.
Reference	`WaitInterval`	Indicates the wait interval for the outbound dequeue.
Reference	`NoMessageAllowed`	Set to `true` if no message is allowed in the outbound dequeue scenario after a specified wait interval.
Reference	`UseMessageEncodingForTranslation`	Set to `\"true\"` if the `characterSet` field of MQMD is to be used for translation in the inbound direction. If set to `\"false\"`, the translator uses the encoding from the schema file. You can use the `jca.message.encoding` property to overwrite any encoding specified, either in MQMD or in the schema file.

34.1.2.7 Oracle Socket Adapter

Table 34–9 describes the properties available for the Oracle Socket Adapter.

Note: Properties such as `TransMode` and `XSLT`, which are not listed in Table 34–9, are displayed in the **Properties** tab. However, you cannot edit properties that are not listed in Table 34–9. Making changes to noneditable properties makes the composite invalid and requires remodeling of the composite.

Table 34–9 Oracle Socket Adapter Properties

Service, Reference, or Endpoint	Property Name	Description
Service	`Port`	The port on which the inbound socket adapter listens for incoming socket connections.
Service	`Encoding`	Indicates the character encoding of the native data. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Service	`ByteOrder`	Indicates the byte order of the native data as `bigEndian` or `littleEndian`. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Reference	`Host`	Indicates the host to which the outbound socket adapter opens a socket connection.
Reference	`Port`	Indicates the port to which the outbound socket adapter tries to connect.
Reference	`Encoding`	Indicates the character encoding of the native data. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.
Reference	`ByteOrder`	Indicates the byte order of the native data as `bigEndian` or `littleEndian`. You *cannot* edit this property in Oracle Enterprise Manager Fusion Middleware Control.

34.1.2.8 Oracle JCA Adapters Endpoint Properties

Table 34–10 describes the endpoint properties available for the Oracle JCA Adapters.

You cannot add or remove the endpoint properties without redeploying the composite. However, you can change the endpoint properties by using Oracle Enterprise Manager Fusion Middleware Control without redeploying the composite.

Table 34–10 Oracle JCA Adapters Endpoint Properties

Direction (Inbound/Outbound)	Property Name	Description
Inbound	`activationInstances`	This property increases the number of polling (worker) threads for any inbound JCA resource adapter. It is only meant to help increase concurrency (scalability) for adapters that do not natively support multithreading. Since most of the adapters included with Oracle Fusion Middleware natively support multithreading, this setting is mostly useful to third party (custom) JCA adapters, which do not natively support multithreading. Set this property to the number of threads required for a particular JCA service (endpoint).
Inbound	`UseWorkManager`	By default, JCA adapters use the standard Oracle Fusion Middleware Oracle WebLogic Server work manager for starting polling (worker) threads. However, if you want a particular JCA service (endpoint) to use a custom (user-defined) work manager, then this property enables you to specify the name of a custom work manager. The work manager is only used to start the JCA Service (endpoint) for which this property has been defined.
Inbound	`rejectUncorrelatedMessages`	When native correlation is used to correlate an inbound asynchronous message with a previous outbound message (by way of defining a callback interface (for a reference) or by a midprocess receive (in BPEL)), the JCA framework normally tries to post the message to the composite, whether the inbound message can be correlated or not. By setting this property to `true`, the JCA framework rejects a message, which cannot be correlated (when native correlation is active).
Inbound	`jca.retry.count`	Indicates the maximum number of retries before rejection.
Inbound	`jca.retry.interval`	Indicates the time interval between retries (measured in seconds).
Inbound	`jca.retry.backoff`	Indicates the retry interval growth factor (positive integer).
Inbound	`jca.retry.maxInterval`	Indicates the maximum value of retry interval; that is, a cap if backoff is greater than `1`.
Outbound	`jca.retry.count`	Indicates the maximum number of retries before throwing a retryable error condition back to the invoking service engine.
Outbound	`jca.retry.interval`	Indicates the time interval between retries (measured in seconds).
Outbound	`jca.retry.backoff`	Indicates the retry interval growth factor (positive integer).
Outbound	`jca.retry.maxInterval`	Indicates the maximum value of retry intervals; that is, a cap if the backoff is greater than `1`.
Outbound	`jca.retry.maxPeriod`	Indicates the maximum total retry period. Retries do not occur longer than the value specified in this parameter.

The following examples show how to specify the endpoint properties in the `composite.xml` file in Oracle JDeveloper:

- Example 1: Specifying Endpoint Properties - Inbound
- Example 2: Specifying Endpoint Properties - Outbound

Example 1: Specifying Endpoint Properties - Inbound

```
<service name="Inbound">
  <interface.wsdl interface="http://xmlns...#wsdl.interface(Inbound_PortType)"/>
  <binding.jca config="Inbound_db.jca">
    <property name="jca.retry.interval">5</property>
    <property name="jca.retry.interval">1</property>
    <property name="jca.retry.backoff">2</property>
    <property name="jca.retry.maxInterval">6</property>
  </binding.jca>
</service>
```

Notice that the inbound property `jca.retry.maxPeriod` is not supported (like it is in the outbound direction).

Example 2: Specifying Endpoint Properties - Outbound

```
<reference name="Outbound">
  <interface.wsdl interface="http://xmlns...#wsdl.interface(Outbound_PortType)"/>
  <binding.jca config="Outbound_jms.jca">
    <property name="jca.retry.count">5</property>
    <property name="jca.retry.interval">1</property>
    <property name="jca.retry.backoff">2</property>
    <property name="jca.retry.maxInterval">6</property>
    <property name="jca.retry.maxPeriod">30</property>
  </binding.jca>
</reference>
```

34.1.3 Changing the Endpoint Reference and Service Key for Oracle Service Registry Integration

If a reference binding component of the SOA composite application is integrated with Oracle Service Registry (OSR), you can change the endpoint reference and service key in the **General** section of this page.

The **UDDI ServiceKey** field automatically displays the value of `binding.ws property="oracle.soa.uddi.serviceKey"` from the `composite.xml` file if you selected to use UDDI for runtime resolution of the endpoint.

You can edit the **UDDI ServiceKey** field after the SOA composite application has been deployed to either:

- Change the value as needed.
- Add it to a composite that did not use UDDI for runtime endpoint resolution.

The **Endpoint Address** field represents the endpoint location as defined with the `ws.binding endpointURI` property in the `composite.xml` file. The **Endpoint Address** field is not filled in after the SOA composite application has been deployed, but can override the endpoint location in the concrete WSDL.

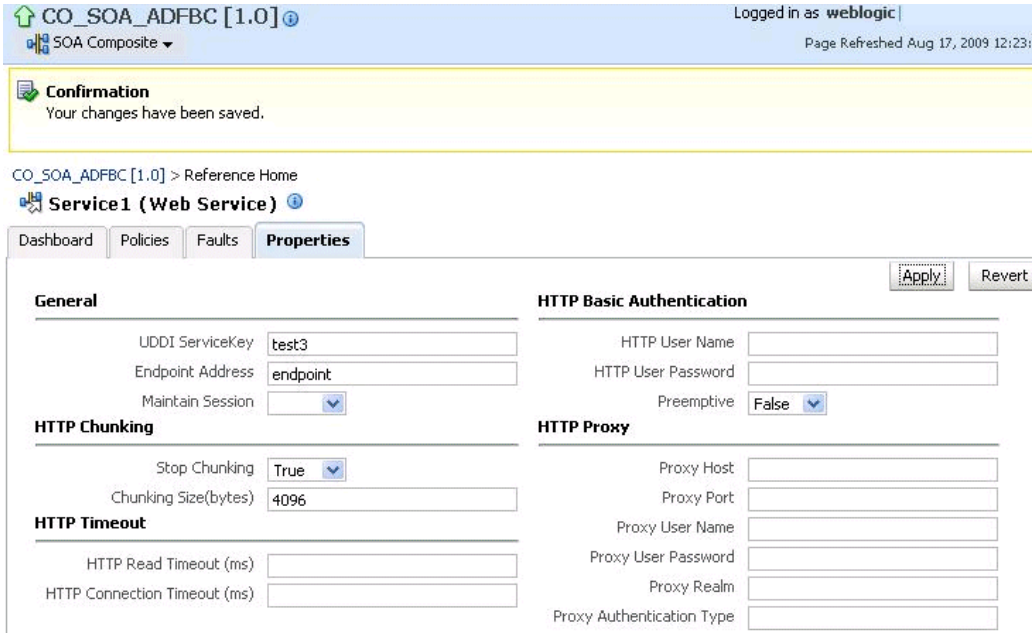
The endpoint location order of precedence is as follows:

- Dynamically set the binding `oracle.soa.uddi.serviceKey` at runtime in the **UDDI ServiceKey** field.
- Dynamically set the binding property `endpointURI` at runtime in the **Endpoint Address** field.

- Use the binding property value for `oracle.soa.uddi.serviceKey` in the `composite.xml` file (viewable and editable in Oracle Enterprise Manager Fusion Middleware Control).
- Use the binding property value for `endpointURI` in the `composite.xml` file (viewable and editable in Oracle Enterprise Manager Fusion Middleware Control).
- Use the location specified in the concrete WSDL.

Figure 34–1 provides shows both fields.

Figure 34–1 Endpoint Reference and Service Key Properties



1. In the **UDDI ServiceKey** field, change the service key to use during runtime.
2. In the **Endpoint Address** field, enter the endpoint address to use during runtime.

You can edit both fields. The value for one field is selected and used based on what you selected in the UDDI Deployment Options dialog during design time. The changes to these fields are persisted in the `composite.xml` file during runtime.

For information about design-time tasks such as how to publish a business service, create a connection to the UDDI registry, and configure a SOA project to invoke a service from the registry, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

For information about how to set the inquiry URL during runtime, see Section 3.1, "Configuring SOA Infrastructure Properties."

34.1.3.1 Configuring Caching of WSDL URLs

Caching of endpoint WSDL URLs occurs by default during runtime. If an endpoint WSDL URL is resolved using the orauddi protocol, subsequent invocations retrieve the WSDL URLs from cache, and not from OSR. You can increase the amount of time that the endpoint WSDL URL is available in cache for inquiry by the service key with the **UddiCacheLifetime** property. This property invalidates the cache at specified time intervals. The default value is `86400` seconds. The minimum value is `300` seconds.

To configure endpoint caching of WSDL URLs:

1. From the **SOA Infrastructure** menu, select **Administration** > **System MBean Browser**.
2. Select **Application Defined MBeans** > **oracle.as.soainfra.config** > **Server: soa_server1** > **SoaInfraConfig** > **soa-infra** > **Attributes**.
3. Click the **UddiCacheLifetime** property on the right side of the page.
4. Enter a value.
5. Click **Apply**.

35

Monitoring Service and Reference Binding Components

This chapter describes how to monitor service and reference binding components in SOA composite applications, including monitoring instances, faults, and rejected messages.

This chapter includes the following topics:

- Section 35.1, "Monitoring Binding Component Instances and Faults"
- Section 35.2, "Monitoring Binding Component Rejected Messages"

For more information, see the following documentation:

- Section 1.2.5, "Introduction to Binding Components" for conceptual details about binding components
- Part XII, "Administering Oracle B2B" for details about Oracle B2B
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

35.1 Monitoring Binding Component Instances and Faults

You can monitor instances and faults for all binding components included in SOA composite applications.

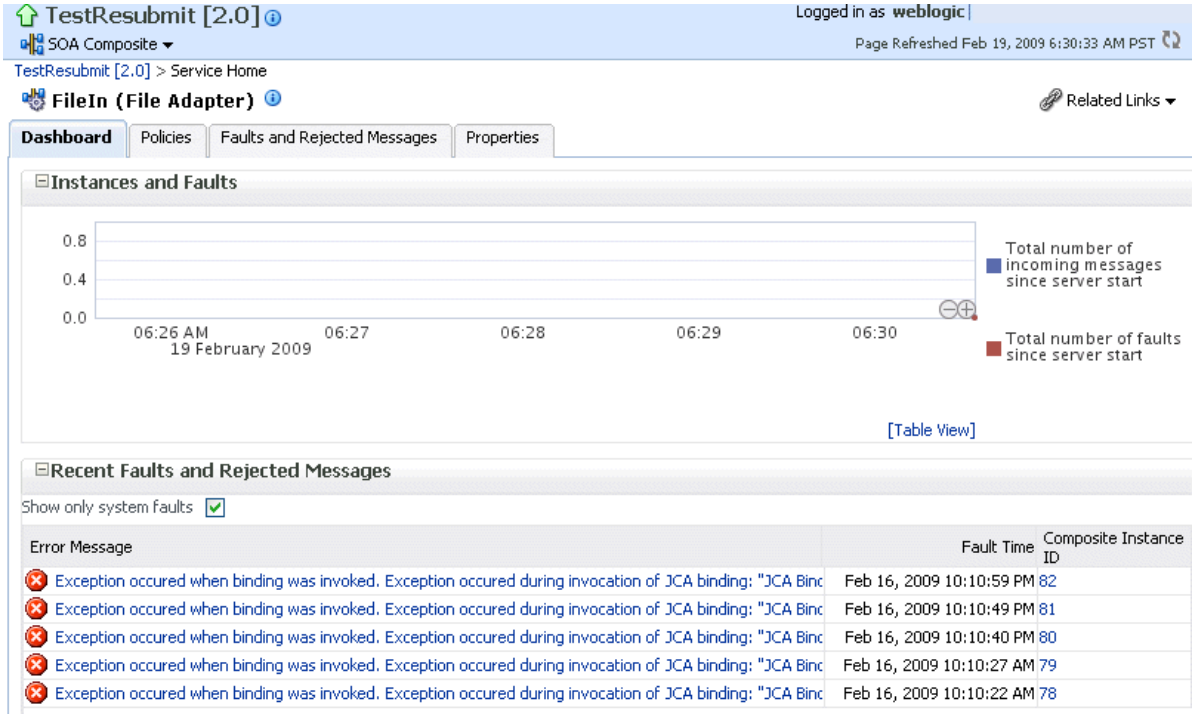
To monitor binding component instances and faults:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, expand the partition. 2. Select a specific SOA composite application.

2. Click **Dashboard** (if it is not selected).
3. Select a specific service or reference in the **Services and References** section.
4. If you select a service that is a JCA adapter or web service, the Dashboard page displays the following details:
   - A graphic representation of the total incoming messages and faults since server startup.

- Recently rejected messages, including the message name, time of the fault, and type of fault (business or system).

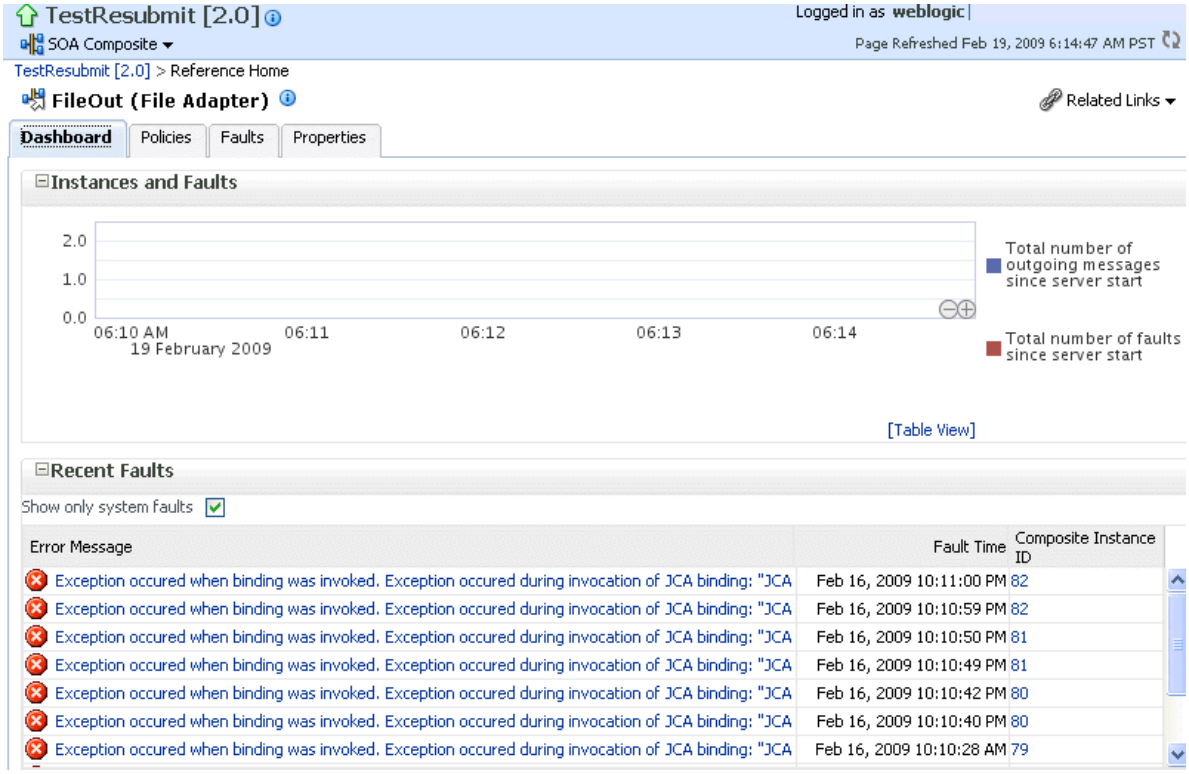


5. If you select a service that invokes a SOA composite application through a remote RMI call (known as direct binding), the Dashboard page displays similar details as described in Step 4. Direct binding enables SOA composite applications to be invoked through an API that supports transaction and security propagation across JVMs.

   The word **Direct** is displayed in the header to indicate that this is a direct binding component. However, no **Polices** and **Properties** tabs are available for direct binding components.

6. If you select a reference, the Dashboard page displays the following details:
   - A graphic representation of the total outgoing messages and faults since server startup.
   - Recent faults, including the time of the fault and the type of fault (business or system).

   Reference binding components are only available for JCA adapters and web services.



35.2 Monitoring Binding Component Rejected Messages

You can monitor rejected messages for all binding components included in a SOA composite application.

To monitor binding component rejected messages:

1. Access this page through one of the following options:

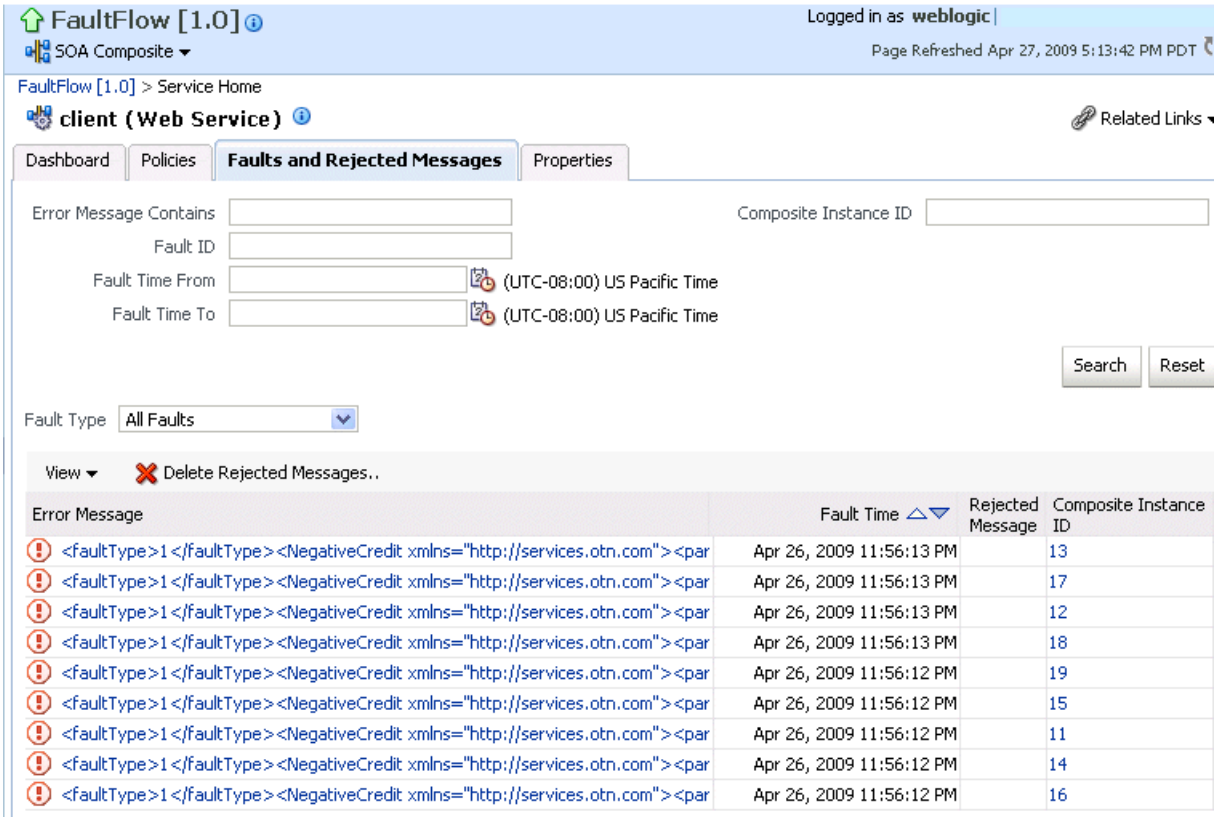
From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**.	1. Under **soa-infra**, expand the partition.
2. Select the **Deployed Composites** tab.	2. Select a specific SOA composite application.
3. In the **Composite** section, select a specific SOA composite application.	

2. Select a service or reference in the **Services and References** section.
3. Click **Faults and Rejected Messages**.

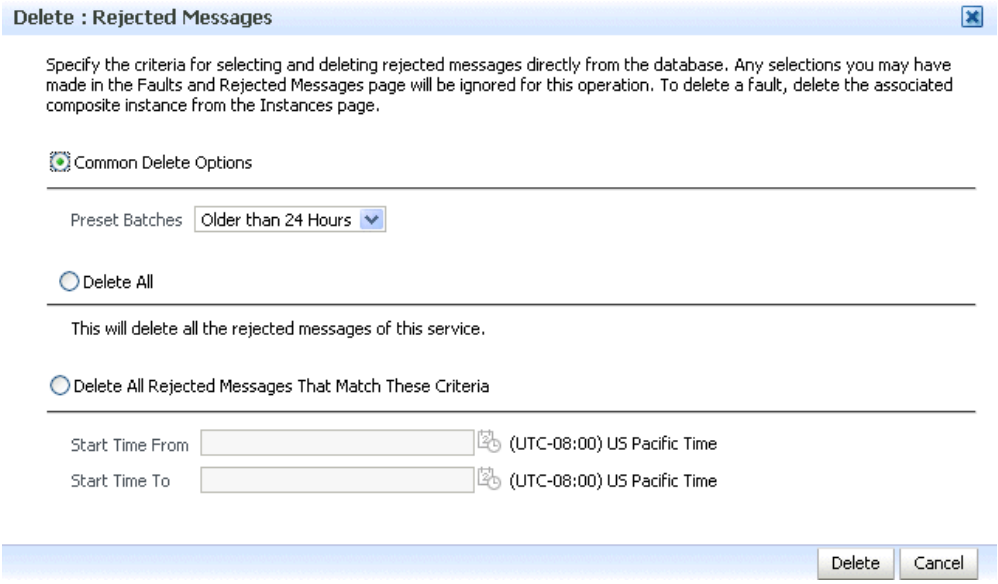
   The Faults and Rejected Messages page shows the list of faults and rejected messages, including details such as the error message, time of the fault, and the

associated composite instance ID. Depending upon the type of the binding component selected, the faults can be on the incoming messages processed by a service binding component or outgoing messages processed by a reference binding component.

You can perform fault recovery from this page.



4. Click a specific message in the **Error Messages** column to display complete fault details, including the fault ID, fault time, fault location, fault type, and error message text. A **Recover Now** option is displayed for recoverable faults. Click **Recover Now** to perform fault recovery.
5. If you want to delete rejected messages., click **Delete Rejected Messages**.
6. This displays a dialog for specifying criteria for deleting rejected messages.



7. Specify criteria, and click **Delete**.

36

Managing Service and Reference Binding Components

This chapter describes how to manage policies for web service and JCA adapter service and reference binding components in SOA composite applications and publish service binding components to the Universal Description, Discovery, and Integration (UDDI) registry from a registered UDDI source.

This chapter includes the following topics:

- Section 36.1, "Managing Binding Component Policies"
- Section 36.2, "Publishing Web Services to the UDDI Registry"

Notes:

- Oracle SOA Suite does not support multiple bindings for service or reference binding components (for example, specifying both SOAP 1.1 and SOAP 1.2 in the `composite.xml` file). Support is only provided for a single web service binding per service or reference. If you specify multiple bindings, remove all but one and redeploy your SOA composite application.
- The only type of binding component to which you can attach a policy is a web service. If you click the **Policies** tab for another type of binding component, such as a JCA adapter, ADF-BC service, or EJB service, the following message is displayed:

  ```
  Policies can be attached to web service bindings and SOA
  components.
  ```

For more information, see the following documentation:

- Section 1.2.5, "Introduction to Binding Components" for conceptual details about binding components
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

36.1 Managing Binding Component Policies

You can attach and detach security policies to and from binding components included in a currently deployed SOA composite application. Policies apply security to the delivery of messages. Oracle Fusion Middleware uses a policy-based model to manage web services.

> **Note:** Before attaching policies, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for definitions of available policies and details about which ones to use in your environment.

To manage binding component policies:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**.	1. Under **soa-infra**, expand the partition.
2. Select the **Deployed Composites** tab.	2. Select a specific SOA composite application.
3. In the **Composite** section, select a specific SOA composite application.	

   The list of currently deployed SOA composite applications appears.

2. In the **Composite** list, select an application.

   The Dashboard page for the selected SOA composite application appears. The **Services and References** section of this page displays the binding components being used in the application.

3. In the **Services and References** section, select a service or reference.

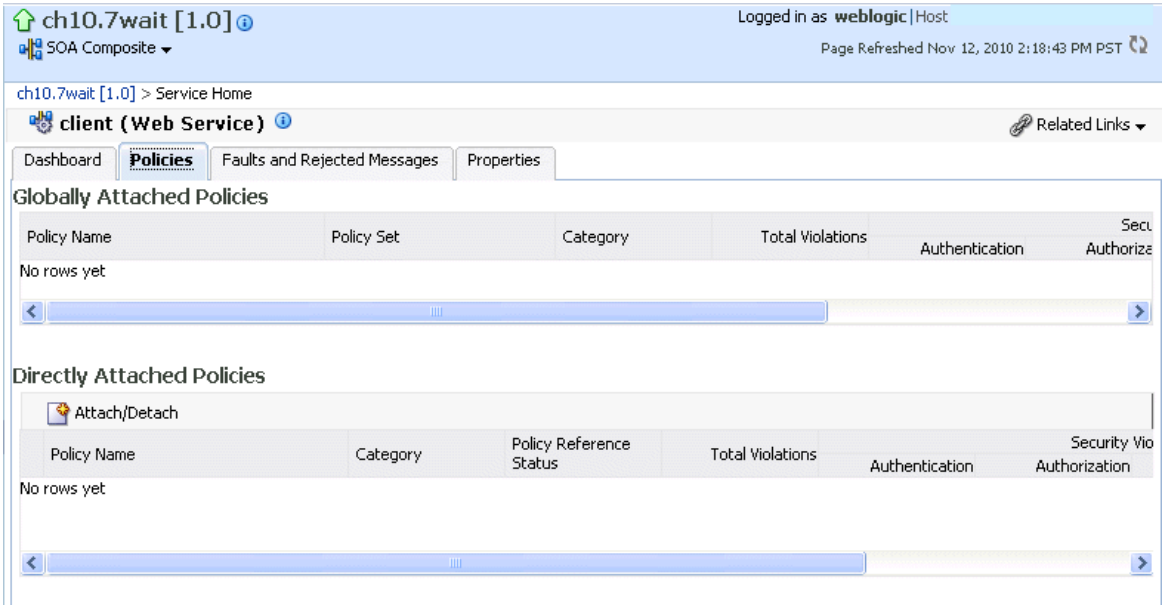
4. Click **Policies**.

   The Policies page enables you to view the globally-attached and directly-attached policies, and to attach or detach security policies to and from a service or reference binding component:

   - The **Globally Attached Policies** table displays the globally-attached policy name, the policy set, the category (Management, Reliable Messaging, MTOM Attachment, Security, or WS Addressing), the violations since the SOA Infrastructure was last restarted, and the authentication, authorization, confidentiality, and integrity failures since the SOA Infrastructure was last restarted.

     Policy sets provide a means to attach policies globally to a range of endpoints of the same type. Attaching policies globally using policy sets enables an administrator to ensure that all subjects are secured in situations in which the developer, assembler, or deployer did not explicitly specify the policies to attach. Policies that are attached using a policy set are considered externally attached. For information about creating and managing policy sets, see Chapter "Creating and Managing Policy Sets" of *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

   - The **Directly Attached Policies** table displays the directly-attached policy name, the policy reference status (enabled or disabled), the category, the violations since the SOA Infrastructure was last restarted, and the authentication, authorization, confidentiality, and integrity failures since the SOA Infrastructure was last restarted.



5. In the **Directly Attached Policies** section, click **Attach/Detach**.

   If multiple components are available, you are prompted to select the service or component for which to perform the attachment or detachment.

   > **Note:** If you attach a policy to a service binding component (client) and initiate an instance of the SOA composite application in the Test Web Service page, and the policy attachment fails, an Oracle Web Services Manager (OSWM) policy error is not generated and viewable in Oracle Enterprise Manager Fusion Middleware Control.
   >
   > If the same SOA composite application instance is initiated externally, a policy error is generated and viewable in Oracle Enterprise Manager Fusion Middleware Control.
   >
   > For service components (such as a BPEL process) or reference binding components, the policy error is always generated and viewable, regardless of whether the application instance was initiated externally or internally through the Test Web Service page.

6. Select the service or component to which to attach or detach a policy.

   This invokes a dialog for attaching or detaching policies.

   Policies currently attached appear in the **Attached Policies** section. Additional policies available for attachment appear in the **Available Policies** section.

7. Select policies to attach that are appropriate to your environment.

8. Click **Attach**.

9. When you are finished attaching policies, click **Validate**.

10. If an error message appears, make the necessary corrections until you no longer have any validation errors.

    The attached policy is displayed in the policies table.

11. Click **OK**.

For more information, see the following documentation:

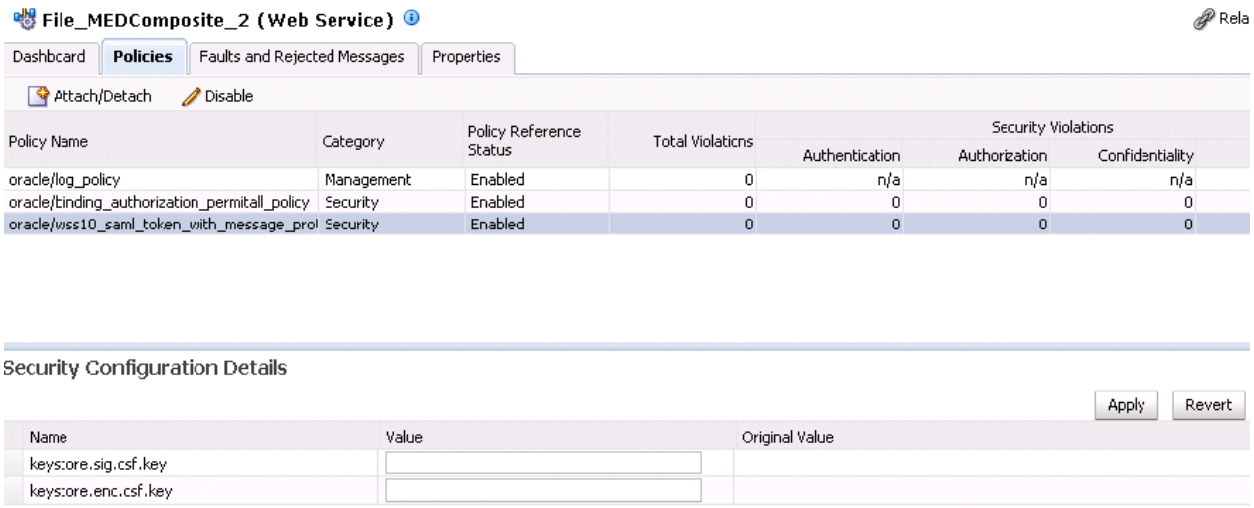
- Section 1.4.3.2, "Introduction to Policies"
- Section 7.6, "Managing SOA Composite Application Policies" for the dialogs that are displayed during policy attachment
- *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for definitions of available policies and details about which ones to use for your environment

36.1.1 Override Policy Configuration Property Values

Your environment may include multiple servers with the same policies. However, each server may have their own specific policy requirements. To satisfy your runtime requirements, you can override the property values for some management and security policies attached to service and reference binding components.

1. Follow the instructions in Section 36.1, "Managing Binding Component Policies" to attach a policy to a service or reference binding component.
2. Select the attached policy in the table.

   The **Security Configuration Details** table is displayed at the bottom of the page.



3. In the **Value** field, enter a value to override the default value in the **Original Value** column.
4. Click **Apply**.

For more information on overriding policy values, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

36.2 Publishing Web Services to the UDDI Registry

You can publish service binding components to the UDDI registry from a registered UDDI source.

Notes:

- You *cannot* publish a reference binding component to the UDDI registry.
- You can *only* publish web services to the UDDI registry. For example, you cannot publish a JCA adapter.

For information about the Oracle Service Registry:

1. Visit the following URL:

   http://www.oracle.com/technetwork/topics/soa/whatsnew/index.html

2. Click the **Products** tab.
3. Click **Governance**.

For information about Oracle Enterprise Repository, see *Oracle Fusion Middleware Configuration Guide for Oracle Enterprise Repository* and *Oracle Fusion Middleware Integration Guide for Oracle Enterprise Repository*.

For more information about publishing web services to the UDDI registry, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

36.2.1 Configuring the Environment for Publishing Web Services to UDDI

Before you can publish a web service to the UDDI registry, you must configure your environment.

If your SOA Infrastructure installation is behind a firewall, the Oracle Enterprise Manager Fusion Middleware Control managed server must have a proxy server configured for outbound URL connections.

To configure the environment for publishing web services to UDDI:

1. Configure the proxy server setting in either of the following ways:

   In the following file:

   a. Open the `FMW_Home/user_projects/domains/soainfra/bin/setDomainEnv.sh` file for Linux (or `setDomainEnv.bat` for Windows operating systems).

   b. Enter the following syntax and provide host, port, and proxy host details appropriate to your environment:

   ```
   EXTRA_JAVA_PROPERTIES="${EXTRA_JAVA_PROPERTIES} -DproxySet=true
   -Dhttp.proxyHost=www-myproxy.us.mycompany.com -Dhttp.proxyPort=80
   -Dhttp.nonProxyHosts=localhost|$HOST_IP_
   ADDRESS|$HOSTNAME|*.us.mycompany.com"
   export EXTRA_JAVA_PROPERTIES
   ```

   c. Run `setDomainEnv.sh` or setDomainEnv.bat.

   With an environment variable:

   a. Set the `PROXY_SETTINGS` environment variable before starting Oracle WebLogic Server.

   ```
   setenv PROXY_SETTINGS "-DproxySet=true
   -Dhttp.proxyHost=www-myproxy.us.mycompany.com -Dhttp.proxyPort=80
    -Dhttp.nonProxyHosts=localhost|*.us.mycompany.com
   ```

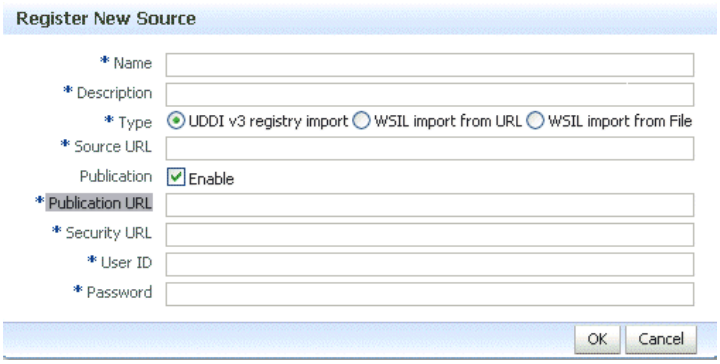
```
|0:0:0:0:0:0:0:1|fe80:0:0:0:250:56ff:fe31"
```

You must set up a UDDI source to use for publication.

1. In the navigator, expand **Weblogic Domain**.
2. Right-click **soainfra** and select **Web Services**> **Registered Services**.

   The Registered Sources and Services page appears.
3. In the **Sources** section, click **Register Web Services**.

   The Register New Source dialog appears.



4. Enter the following information:

Field	Description
Name	Enter a descriptive name for the service.
Description	Enter a descriptive name for the UDDI source to which to publish the WSDL.
Type	Select **UDDI v3 registry import**.
Source Location	Enter the source location. For example: `http://myhost.us.mycompany.com:7001/registry/uddi/inquiry`
Publication	Select the **Enable** checkbox.
Publication URL	Enter the publication URL. For example: `http://myhost.us.mycompany.com:7001/registry/uddi/publishing`
Security URL	Enter the security URL. For example: `http://myhost.us.mycompany.com:7001/registry/uddi/security`
User ID	Enter `admin`.
Password	Enter the user ID password.

5. Click **OK**.

36.2.2 Publishing a Web Service to the UDDI Registry

> **Note:** You can publish web services to default Oracle Service Registry businesses from Oracle Enterprise Manager Fusion Middleware Control. To publish to nondefault businesses, use the publish option in Oracle Service Registry.
>
> For more information about Oracle Service Registry, including documentation, visit the following URL:
>
> `http://www.oracle.com/technetwork/middleware/registry/overview/index.html`

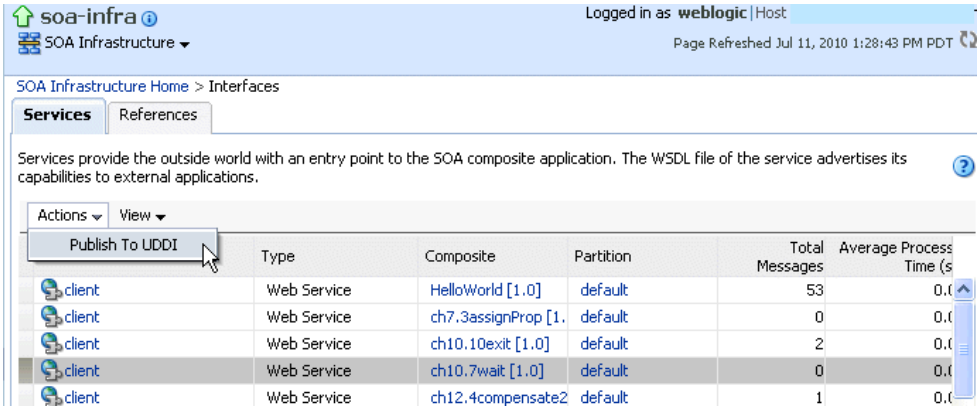
To publish a web service to the UDDI registry:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Services and References**.	1. Right-click **soa-infra**. 2. Select **Services and References.**

   The Services page displays details about the names and types of the services, the SOA composite applications in which the services are used, the partition in which the composite is deployed, the total number of messages processed, the average processing time, and the number of faults occurring in the services.

2. In the **Service** table, select a service to publish to the UDDI registry.
3. From the **Actions** list, select **Publish To UDDI**.



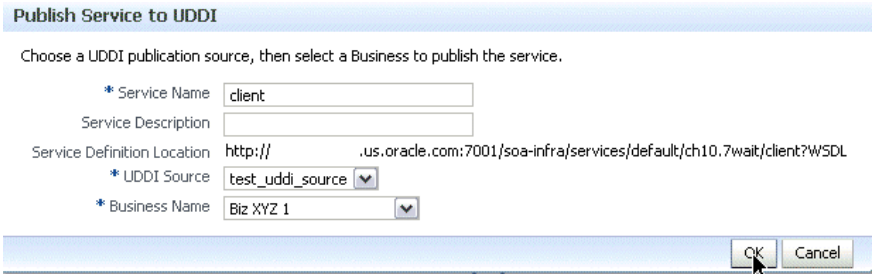
   The Publish Service to UDDI dialog appears.

4. Enter the following information:

Field	Description
Service Name	Displays the name of the selected service.
Service Description	Enter an optional description of the selected service.

Field	Description
System Definition Location	Displays the WSDL URL to publish to the UDDI registry. For example: `http://myhost.mycompany.com:7001/soa-infra/services/default/HelloWorld/client?WSDL`
UDDI Source	Select the UDDI publishing source from which to register the service.
Business Name	Select a business to publish the service. This is the name of the data structure in the UDDI registry. It is assumed that the business has already been registered in the UDDI registry.

When complete, the Publish Service to UDDI dialog looks similar to the following:



5. Click **OK**.

Part XV

Administering Oracle BPMN Process Service Components and Engines

This part describes how to administer Oracle BPMN Process Service Components and Engines.

This part includes the following chapters:

- Chapter 37, "Configuring Oracle BPMN Process Service Components and Engines"
- Chapter 38, "Monitoring BPMN Process Service Components and Engines"
- Chapter 39, "Managing Oracle BPMN Service Components and Engines"

37

Configuring Oracle BPMN Process Service Components and Engines

This chapter describes for administrators how to configure the BPMN process service engine, including configuring properties used by the engine during processing of BPMN service components.

This chapter includes the following topics:

- Section 37.1, "Configuring BPMN Process Service Engine Properties"
- Section 37.2, "Integrating Oracle BPM with Oracle Business Activity Monitoring"

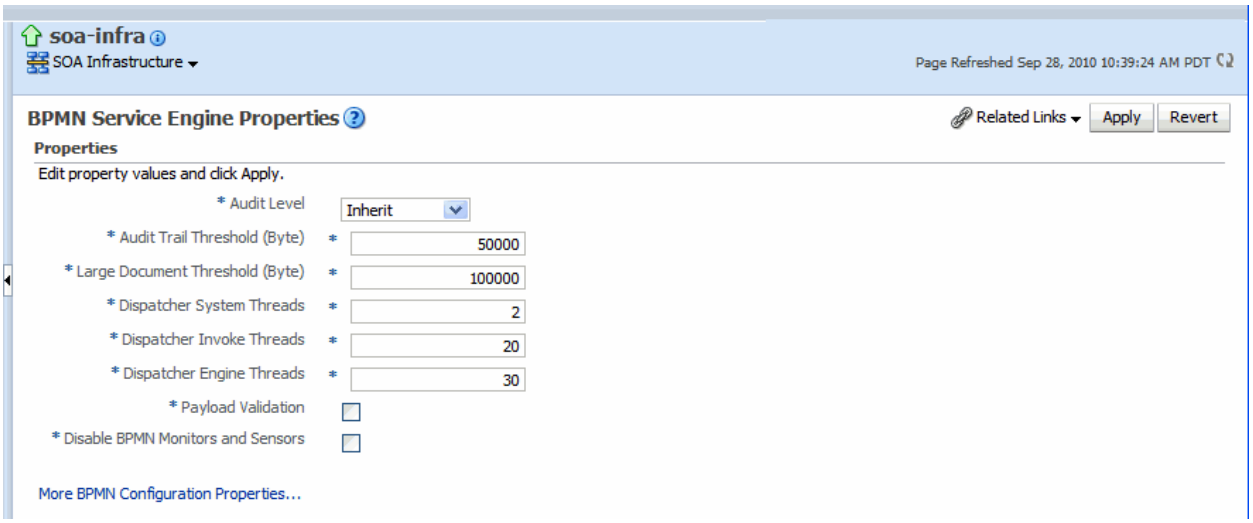
37.1 Configuring BPMN Process Service Engine Properties

To configure BPMN process service engine properties:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **SOA Administration** > **BPMN Properties**.	1. Right-click **soa-infra**. 2. Select **SOA Administration** > **BPMN Properties**.

The BPMN Service Engine Properties page displays properties for setting audit trail and large document thresholds, setting dispatcher thread properties, validating payload schema, and setting the audit trail level.



2. Make changes to the service engine properties that are appropriate to your environment.

Property	Description
Audit Level	Select one of the following options: ▪ **Off**: No logging is performed. Composite instance tracking and payload tracking information are not collected. If measurement is enabled, then this level is overridden to Minimal. ▪ **Inherit** (default): Logging equals the SOA Infrastructure audit level. This allows the BPMN audit level to automatically change when the global setting is changed. Setting a different audit level tracking in this page overrides the tracking set at the SOA Infrastructure level. ▪ **Minimal**: Instance tracking information is collected, but not payload details; no payload details are available in the flow audit trails. ▪ **Production**: Instance tracking information is collected, payload details are collected only for out data associations for asynchronous activities. This level is optimal for most normal operations and testing. ▪ **Development**: Allows both the composite instance tracking and payload tracking. However it may impact the performance. This level is useful mostly for debugging purposes. **Note:** If you do not want audit entries to be displayed, then you must turn off both the audit level and the metrics. If metrics are enabled, then audit entries are displayed even if the audit level is set to Off.
Audit Trail Threshold	Enter the maximum size in bytes of an instance audit trail before it is chunked and saved in a dehydration store table separate from the audit trail. If the threshold is exceeded, the **View XML** link is shown in the audit trail instead of the payload.
Large Document Threshold	Enter the maximum size of a generated document within a BPMN process component instance before it is stored in a separate table in the dehydration store.

Property	Description
Dispatcher System Threads	Specify the total number of threads allocated to process system dispatcher messages. System dispatcher messages are general clean-up tasks that are typically processed quickly by the server (for example, releasing stateful message beans back to the pool). Typically, only a small number of threads are required to handle the number of system dispatch messages generated during runtime. The default value is `2` threads. Any value less than `1` thread is changed to the default.
Dispatcher Invoke Threads	Specify the total number of threads allocated to process invocation dispatcher messages. Invocation dispatcher messages are generated for each payload received and are meant to instantiate a new instance. If the majority of requests processed by the service engine are instance invocations (as opposed to instance callbacks), greater performance may be achieved by increasing the number of invocation threads. Higher thread counts may cause greater CPU utilization due to higher context switching costs. The default value is `20` threads. Any value less than `1` thread is changed to the default.
Dispatcher Engine Threads	Specify the total number of threads allocated to process service engine dispatcher messages. Service engine dispatcher messages are generated whenever an activity must be processed asynchronously. If the majority of processes deployed are durable with a large number of dehydration points (midprocess receive, onMessage, onAlarm, and wait activities), greater performance may be achieved by increasing the number of service engine threads. Higher thread counts can cause greater CPU utilization due to higher context switching costs. The default value is `30` threads. Any value less than `1` thread is changed to the default.
Payload Validation	Select to enable validation of inbound and outbound messages. Nonschema-compliant payload data is intercepted and displayed as a fault. **Note:** This setting is independent of the SOA composite application and SOA Infrastructure payload validation level settings. If payload validation is enabled at both the service engine and SOA Infrastructure levels, data is checked twice: once when it enters the SOA Infrastructure, and again when it enters the service engine.
Disable BPMN Monitors and Sensors	Select this checkbox to disable all BPMN monitors and sensors defined for all BPMN components across all deployed SOA composite applications.

3. Click **Apply**.
4. If you want to configure advanced BPMN properties in the System MBean Browser, click **More BPMN Configuration Properties**. Properties that display include the following. Descriptions are provided for each property.
   - **AuditDetailThreshold**: The maximum size (in bytes) an audit trail details string can be before it is stored separately from the audit trail.
   - **AuditLevel**: Controls the amount of audit events logged by a process; currently supported logging levels are: `off`: absolutely no logging performed whatsoever; may result in a slight performance boost for processing instances.
   - **BpelcClasspath**: The extra class path must be included when compiling BPMN generated java sources.
   - **ConfigMBean**: If true, it indicates that this MBean is a Config MBean.

- **CubeInstanceExpiration**: The expiration time in hours of performance data. This parameter is disabled by default. You can enable it.
- **CubeTimerMaxErrorCount**: Maximum allowed number of consecutive errors during cube timer processing. Once number of errors reaches `CubeTimerMaxErrorCount`, the cube timer skips processing.
- **CubeTimerMaxSkipOnErrorCount**: Specifies the number of times the timer skips processing once the number of errors reaches *`CubeTimerMaxErrorCount`*.

Note: Once the cube timer errors out consecutively for *`CubeTimerMaxErrorCout`* times, the cube timer skips the timeout processing for *`CubeTimerMaxSkipOnErrorCount`* times. It then resumes the normal timeout processing. The time for which the cube timer skips processing can be calculated as *`CubeTimerMaxErrorCount`* * *`CubeUpdateFrequency`*.

If the lockout time must be less, you must either reduce the time to a lesser value or change the number of times to skip timeout processing.

- **CubeUpdateFrequency**: Frequency in seconds at which cube action calculates the workload.
- **CubeWorkloadExpiration**: The expiration time for workload records. It is set to 48 hours by default, which means that if a workload record's age is more than 48 hours, it is purged.
- **DisableActions**: Comma-delimited list of disabled measurement actions (such as `CubeCommand`, `BAMCommand`). Also see **PublishMaxTrackBackCount**.
- **DisableProcessTracking**: If set to `true`, the audit disables process tracking. The default value is `false`.
- **DisableSensors**: If set to `true`, the service engine disables all calls to sensors. The default value is `false`.
- **DispatcherEngineThreads**: The total number of threads that are allocated to process engine dispatcher messages.
- **DispatcherInvokeThreads**: The total number of threads that are allocated to process invocation dispatcher messages.
- **DispatcherMaxRequestDepth**: Maximum number of internal messages the service engine processes. If this number is exceeded, new messages are not dispatched. The default value is `600`.
- **DispatcherSystemThreads**: The total number of threads that are allocated to process system dispatcher messages.
- **eventProvider**: If set to `true`, indicates that this MBean is an event provider as defined by JSR-77.
- **eventTypes**: All the event's types emitted by this MBean.
- **ExpirationMaxRetry**: The maximum number of times a failed expiration call (wait/onAlarm) is retried before failing.
- **ExpirationRetryDelay**: The delay between the expiration retries. The default value is `120` seconds.

- **InstanceKeyBlockSize**: The size of the block of instance IDs to allocate from the dehydration store during each fetch.
- **LargeDocumentThreshold**: The maximum size (in bytes) a BPMN variable can be before it is stored in a separate location from the rest of the instance scope data.
- **MaximumNumberOfInvokeMessagesInCache**: Specify the number of invoke messages that can be kept in the in-memory cache, once the service engine reaches this limit, it pushes the message to dispatcher in-memory cache, instead it saves the message in the database, and these saved messages can be recovered using recovery job. Use value `-1` to disable this property.
- **objectName**: The MBean's unique JMX name.
- **OneWayDeliveryPolicy**: Changes whether the one-way invocation messages are delivered.
- **PeopleQueryTimeout**: Specify quartz cron expression People Query. People Query in Logical People Group is reevaluated based on this cron expression.
- **PublishMaxTrackBackCount**: Maximum number of records traversed from last audit record, checked for measurement and action enabled and published if so.
- **QualityOfService**: Flag to enable or disable Oracle Coherence cache for BPMN service engine. Use **CacheEnabled** for enabling Oracle Coherence.
- **ReadOnly**: If set to `true`, indicates that this MBean is a read only MBean.
- **RestartNeeded**: Indicates whether a restart is needed.
- **StatsLastN**: The size of the most recently processed request list.
- **SystemMBean**: If set to `true`, indicates that this MBean is a System MBean.
- **ValidateXML**: If set to `true`, the service engine applies schema validation for incoming and outgoing XML documents. The default value `false`.
- **Version**: Version of the configuration file.
- **WFOracleUcmUrl**: Specify a URL for Oracle WebCenter Content repository administration server.

37.2 Integrating Oracle BPM with Oracle Business Activity Monitoring

When a BPMN composite application is deployed, the following Oracle BAM data objects are generated automatically:

- Data object for the following business indicator:

  `TEMPLATE_BI_Partition_Composite_Process`

  Once this is imported, you should rename it to:

  `BI_DEFAULT_`*`Compositename_ProcessName`*

  This is created in the target folder. If the data object already exists, new columns are added to it, assuming the old columns match data types. Otherwise, an error is thrown.
- COMPONENT, INTERVAL, COUNTER data objects if they are not present in the target folder.

The target Oracle BAM Server is specified by JNDI name parameter.

> **Note:** Data objects can be created only automatically at deployment. You cannot create them manually.

To configure Oracle BPM for use with Oracle Business Activity Monitoring, you perform these tasks:

- Task 1: Configure the Oracle BAM Adapter on Oracle BPM Server
- Task 2: Enable Oracle BAM on the Oracle BPM Server

37.2.1 Task 1: Configure the Oracle BAM Adapter on Oracle BPM Server

You must configure the Oracle BAM Adapter to use either SOAP or RMI for communicating with Oracle BAM.

To configure the Oracle BAM adapter on Oracle BPM server:

1. In the Oracle WebLogic Server Administration Console, under **Domain Structure**, click **Deployments**.
2. Click **OracleBAMAdapter** > **Configuration** > **Outbound Connection Pools**.
3. Expand **oracle.bam.adapter.adc.soap.SOAPConnectionFactory**.
4. Click either **eis/bam/soap** or **eis/bam/rmi**.

   The JNDI name used to configure the Oracle BAM adapter is used. For example, if you configured the Oracle BAM adapter to use SOAP, then the default JNDI name is eis/bam/soap. Similarly, if you configure the Oracle BAM adapter to use RMI, then the default JNDI name is eis/bam/rmi.
5. Modify properties to match Oracle BAM Server. (Remember to press **Enter** after text entry).
6. Click **Save**.
7. Select the location for the deployment plan—for example, `bam/Plan.xml`, then complete the dialogs.
8. Return to **Deployments**.
9. Select **OracleBAMAdapter**.
10. Click **Update** and complete the dialogs.

For more information about integrating Oracle Business Activity Monitoring with Oracle SOA Suite composite applications, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

37.2.2 Task 2: Enable Oracle BAM on the Oracle BPM Server

To do this task, you use the Oracle Enterprise Manager Fusion Middleware Control.

To enable Oracle BAM on the Oracle BPM server:

1. In Oracle Fusion Middleware Control, under **WebLogic Domain**, select your domain and server.
2. Go to **System MBean Browser**.
3. Select **oracle.as.soainfra.config** > **Server** > **BPMNConfig** > **bpmn**.
4. Go to **Disable Actions**. If you find **BAMAction** there, then clear the field.

5. Click **Apply**.

38

Monitoring BPMN Process Service Components and Engines

This chapter describes how to monitor BPMN process service components and service engines.

This chapter includes the following topics:

- Section 38.1, "Viewing the Audit Trail and Process Flow of a BPMN Process Service Component"
- Section 38.2, "Monitoring BPMN Process Service Component Instances and Faults"
- Section 38.3, "Monitoring BPMN Process Service Component Instances"
- Section 38.4, "Monitoring BPMN Process Service Engine Instances and Faults"
- Section 38.5, "Monitoring BPMN Process Service Engine Request and Thread Statistics"
- Section 38.6, "Monitoring BPMN Process Service Engine Instances"
- Section 38.7, "Monitoring Deployed BPMN Processes in the Service Engine"

For more information, see the following sections:

- Section 1.2.4, "Introduction to Service Components and Service Component Instances"
- Section 1.2.6, "Introduction to Service Engines"

38.1 Viewing the Audit Trail and Process Flow of a BPMN Process Service Component

This section describes how to view the audit trail and process flow of a BPMN process service component in a SOA composite application instance.

Note:

- This section assumes a SOA composite application instance has been initiated. If not, see Section 8.1, "Initiating a SOA Composite Application Test Instance" for instructions.
- When several messages are thrown in a short interval, they are not processed in the same order as they were sent. This can be apparent when you are examining the audit trail of a process instance.

To view the audit trail and process flow of a BPMN process service component:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, select a specific SOA composite application.

   The Dashboard page for the selected composite application appears.

2. Use one of the following methods to select an instance of the application:
   - For recent instances of this application, click the instance number of an instance in the **Instance ID** column of the **Recent Instances** section.
   - For all instances of this application, click the **Instances** tab, then click a specific instance in the **Instance ID** list.

   The Flow Trace page displays the following details:

   - The **Faults** section shows the faults occurring in the services, service components, and references that comprise the SOA composite application. Sensors enable you to monitor BPMN process activities, variables, and faults during runtime. Selecting a fault highlights the row in the **Trace** section in which the fault occurred. Closing the fault clears the selection in the **Trace** section.
   - The **Sensors** section displays details about composite sensors included in the service and reference binding components of the SOA composite application. The total number of sensors is shown in the section header. Composite sensors can be added to service and reference binding components during design time in Oracle JDeveloper. You cannot add composite sensors to service components. Selecting a composite sensor in this section highlights the service or reference in the **Trace** section in which composite sensor data was collected. Closing the sensor clears the selection in the **Trace** section.

     Note: Expand the **Faults** or **Sensors** sections one at a time. The fault or sensor information is displayed only for viewing in this way.

   - The **Trace** section shows the sequence of the message flow through the services, service components, and references that comprise the SOA composite application.

   The flow trace is a runtime trail of a message flow identified by an execution context ID (ECID) that is displayed in the upper right corner of the page. An ECID enables you to track a message flow that crosses instances of different composites. The flow trace lists all services, references, components across composites participating in the flow.

Data Refreshed Aug 27, 2010

Flow Trace

This page shows the flow of the message through various composite and component instances.

ECID **484ae83ec35e9489:1bab839d:12a9e920b8b:-8000-00**
Started **Aug 23, 2010 7:28:36 AM**

Faults (0)

Faults

Select a fault to locate it in the trace view.

Error Message	Recovery	Fault Time	Fault Location	Composi
No faults found				

Sensors (0)

Trace

Click a component instance to see its detailed audit trail.

Show Instance IDs

Instance	Type	Usage	State	Time	Composite Instance
Caller.service	Web Service	Service	Completed	Aug 23, 2010 7:28:36 AM	MessageEventOnUserTask of 20
Caller	BPMN Component		Completed	Aug 23, 2010 7:28:54 AM	MessageEventOnUserTask of 20
Process	BPMN Component		Completed	Aug 23, 2010 7:28:52 AM	MessageEventOnUserTask of 20
NonIntTimer	JCA Adapter	Reference	Completed	Aug 23, 2010 7:28:38 AM	MessageEventOnUserTask of 20
nonIntMessageRecd	JCA Adapter	Reference	Completed	Aug 23, 2010 7:28:40 AM	MessageEventOnUserTask of 20
IntMessageRecd	JCA Adapter	Reference	Completed	Aug 23, 2010 7:28:50 AM	MessageEventOnUserTask of 20
IntTimer	JCA Adapter	Reference	Completed	Aug 23, 2010 7:28:52 AM	MessageEventOnUserTask of 20
correctPathTakenBeforeProcResponse	JCA Adapter	Reference	Completed	Aug 23, 2010 7:28:52 AM	MessageEventOnUserTask of 20
ParallelGatewayWithCatchEvents	JCA Adapter	Reference	Completed	Aug 23, 2010 7:28:54 AM	MessageEventOnUserTask of 20
nonIntMsgResponse	JCA Adapter	Reference	Completed	Aug 23, 2010 7:28:54 AM	MessageEventOnUserTask of 20
ProcessResponse	JCA Adapter	Reference	Completed	Aug 23, 2010 7:28:54 AM	MessageEventOnUserTask of 20
IntMsgResponse	JCA Adapter	Reference	Completed	Aug 23, 2010 7:28:54 AM	MessageEventOnUserTask of 20
InclusiveGatewayWithServiceTask	JCA Adapter	Reference	Completed	Aug 23, 2010 7:28:54 AM	MessageEventOnUserTask of 20

For the flow example in the **Trace** section, the service binding component and reference binding component involved in the flow have successfully received and processed messages.

Note the following restrictions with ECIDs:

- A separate ECID is displayed for each instance of a composite application and not for the composite level ECID that can track the complete flow of any instances for the composite application.
- To get complete flow information, you must find the composite level ECID in the log files. Use that value to get all information for a particular composite and therefore all its executed instances.
- ECIDs are not propagated through business events. This can limit the amount of logging information that is collected. For example, if you publish an event that is subscribed to in the same composite application, limited logging information is available.

3. Select a fault in the **Faults** section.

   This highlights the row in the **Trace** section in which the fault occurred.

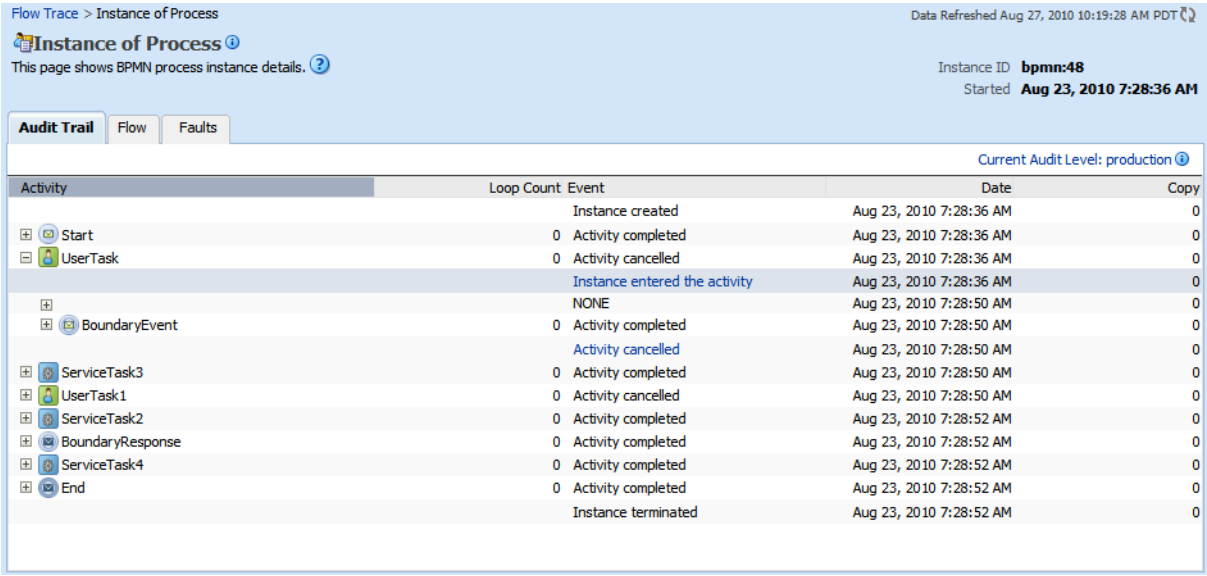
4. Close the fault to clear the selection in the **Trace** section.
5. Expand the **Sensors** section to display composite sensors.
6. Select a sensor in the **Sensors** section.

   This highlights the row in the **Trace** section in which the composite sensor data was collected.

7. In the **Instance** column of the **Trace** section, click a specific BPMN process service component instance. Service component instances can be accessed from this section; services and references cannot be accessed.

The Instance page appears, as shown in Figure 38–1.

Figure 38–1 Instance Page



Use these pages to view the audit trail, flow and faults of a BPMN process service component instance. The following links provide additional details about the instance:

- **Flow Trace link**: Click the breadcrumbs in the upper left corner of the page to access the flow trace for the ECID (composite instance) that contains this BPMN component instance.
- **Information icon**: Click the information icon to the right of the name of the BPMN component (in the page title) to see biographical information about this BPMN instance. This information includes a summary of the instance, including instance ID, ECID, instance startup time or last modification time, instance state (for example, running), and number of faults.

  This icon is displayed only on the Audit Trail pages of BPMN processes and Oracle Mediators, and not on the pages of human tasks and business rules.
- **Current Audit Level**: Click to display information details, such as the audit level used by this instance.

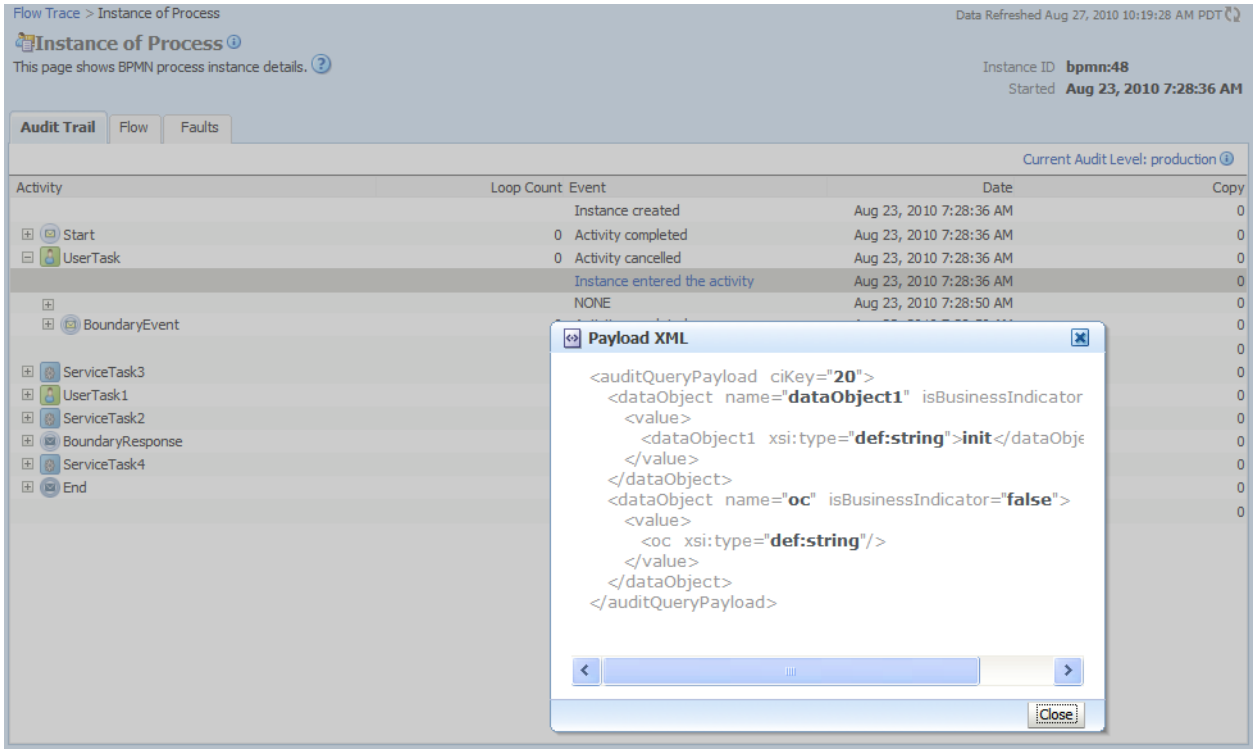
When you first open the Instance page, the **Audit Trail** page is displayed by default. It provides execution details about the activities in the BPMN process.

Column	Description
Activity	Lists all the BPM constructs available in a process in the order they are executed. These include: ▪ Events: start, end, signal, throw, catch message. ▪ Activities: user task, business rules task, service task, call activity, subprocess. ▪ Gateways: inclusive, exclusive, parallel, event based, and complex.

Column	Description
Loop Count Event	If the activity referred to in a row of the audit trail table is a subprocess with the loop characteristic set to either Loop or Multi-instance, this column shows the value of the `loopCounter` variable.
Event	When any BPMN construct executes, audit is logged twice: once when it enters that activity, and once when it leaves that activity to move to next activity. If the node is collapsed, it shows you whether the activity is completed, processing, or canceled.
Date	Time stamp showing when the item was posted.
Copy	If some activities are executed simultaneously—for example, on different paths in a parallel gateway—the copy keeps the number of threads used for this purpose.

8. Scroll through the audit trail to check for errors and expand the payload links to view their contents at a given point in the flow.

   When you click a payload link, the Payload XML page appears. This page shows the value of data objects which had out data association at that particular point in the process.



9. Click the **Flow** tab.

   A flow diagram of the BPMN process activities appears. This flow diagram shows a fault highlighted in a BPMN process activity.

10. Click an activity to view the flow of the payload through the process.

> **Note:** If using Microsoft Internet Explorer, you can click **Copy details to clipboard** to copy the activity details to the clipboard. If using Mozilla Firefox, this link does not appear. Instead, you must manually select the text and copy and paste it to a file.

11. Scroll through the flow diagram to check for errors and click the highlighted activity to view error messages.
12. Close the window.
13. Click the **Faults** tab.

    This page shows the error message, whether you can recover from the fault, the time at which the fault occurred, and the activity in which the fault occurred. This page displays the faults in the BPMN component instance (but not the faults that occurred in a service or reference binding component).

    You can recover from instance faults identified as recoverable. This page lists all instance faults, recoverable or not. The component instance faults that occurred in a service or reference are not listed here.

    This page enables you to target individual faults from which to recover, and provides a degree of fault recovery granularity not available on other pages.

    However, you cannot perform bulk fault recoveries on this page. To perform bulk fault recovery, use one of the following pages:

    - Faults and Rejected Messages page of a specific SOA composite application or of the SOA Infrastructure.
    - Faults page of the BPMN process service engine or of a specific BPMN process service component.
14. Select a fault for recovery that has been identified as recoverable through one of the following methods. The page refreshes to display a fault recovery section at the bottom of the page.
    - If you click a fault in the **Error Message** column, a popup message displays details about the fault, including the fault ID, fault time, fault location, fault type, and complete error message text. If the fault is identified as recoverable, a **Recover Now** button is displayed that you can click.
    - You click a fault identified as recoverable in the **Recovery** column.
15. Select an action from the **Recovery Action** list.

Action	Description
Retry	Retries the instance with an option to provide a retry success action. An example of a scenario in which to use this recovery action is when the fault occurred because the service provider was not reachable due to a network error. The network error is now resolved.
Abort	Terminates the entire instance.
Replay	Replays the entire scope again in which the fault occurred.
Rethrow	Rethrows the current fault. BPMN fault handlers (catch branches) are used to handle the fault. By default, all exceptions are caught by the fault management framework unless an explicit rethrow fault policy is provided.

Action	Description
Continue	Ignores the fault and continues processing (marks the faulting activity as a success).

Your selection causes additional fields to appear. For example, the following fields display if you select **Rethrow**.

16. Use the **After Successful Retry** list to select defined actions to invoke after a successful retry. If you select a variable in the **Variable** list, you can edit the value in the **Value** text box.
17. Click the **Back** button of your browser to exit the flow diagram.

38.2 Monitoring BPMN Process Service Component Instances and Faults

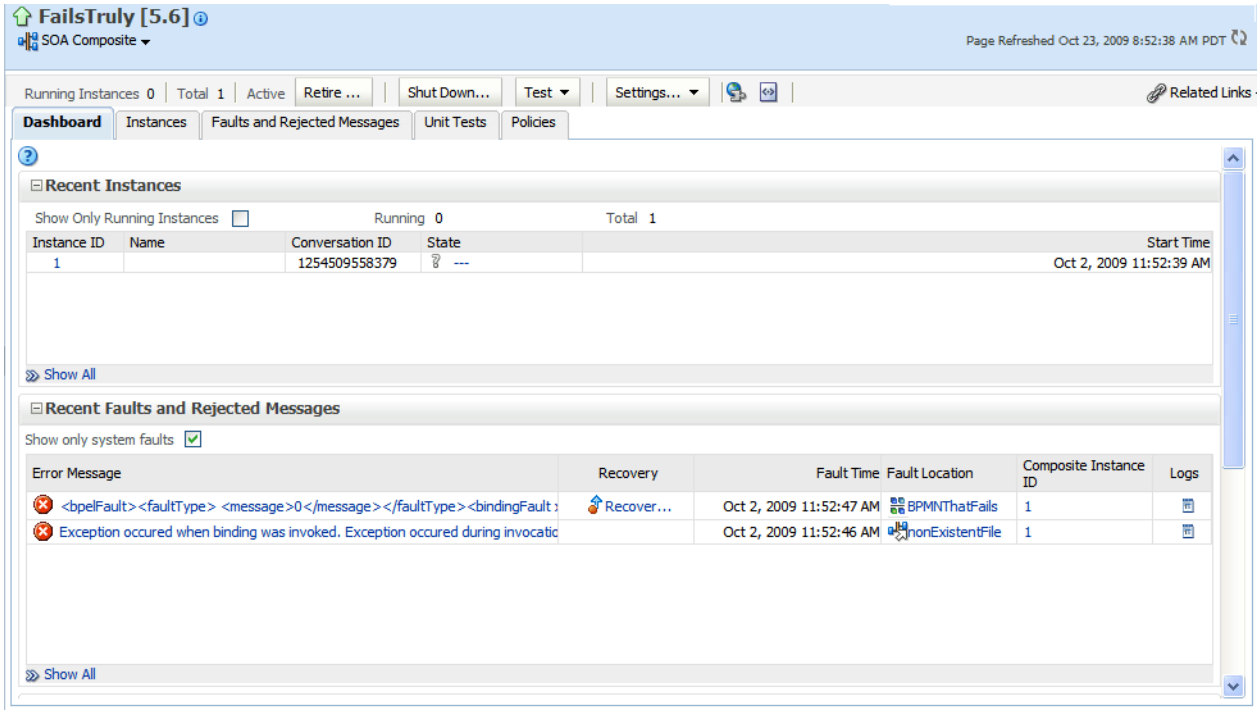
You can monitor BPMN process service component recent instances and faults. Each service component in a SOA composite application has its own instance ID. These IDs are different from the overall instance ID of the SOA composite application of which each service component is a part.

To monitor BPMN process service component instances and faults:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, select a specific SOA composite application.

2. In the **Component Metrics** section, select the BPMN process service component.
3. Click **Dashboard**.



The upper part of the Dashboard page displays the following details:

- Recent instances of the BPMN process service component, including the instance ID, the state of the instance (for example, completed successfully or faulted), the start time, the last modification time, and logs describing the instance.
- Recent faults in the BPMN process service component, including the error message, whether you can recover from the fault, the time at which the fault occurred, the instance ID of the BPMN service component, the BPMN activity in which the fault occurred, and logs describing the fault.
- The average processing time for each activity in the BPMN process service component.

4. In the **Recent Instances** section, you can perform the following tasks:
   - View the audit trail, process flow and faults of a service component. To do this task, in the **Instance ID** column, click the instance ID of that service component.
   - Access the Log Messages page with filtered messages specific to that instance. To do this task, in the **Logs** column, click a specific log.
   - Access the Instances page of the service component. To do this: Click **Show All** below the section.
5. In the **Recent Instances and Faults** section, you can perform the following tasks:
   - Display complete information about a fault. To do this task, in the **Error Message** column, click an error message. If the fault is identified as recoverable, you can perform fault recovery by clicking the **Recover Now** link.
   - Perform fault recovery at the component instance level. To do this task, in the **Recovery** column, click a fault identified as **Recoverable**.

- Access the Log Messages page with filtered messages specific to that instance. To do this task, in the **Logs** column, click a specific log.
- Access the Faults page of the service component. To do this task, click **Show All** below the section.

The lower part of the Dashboard page displays the following details:

- A graphical representation of the number of successful, faulted, and incoming (pending) instances of the BPMN process service component over a specific time range.
- The number of faults and message processed by any reference binding component with which this BPMN process service component communicated.

For more information, see Section 1.2.3, "Introduction to SOA Composite Application Instances."

38.3 Monitoring BPMN Process Service Component Instances

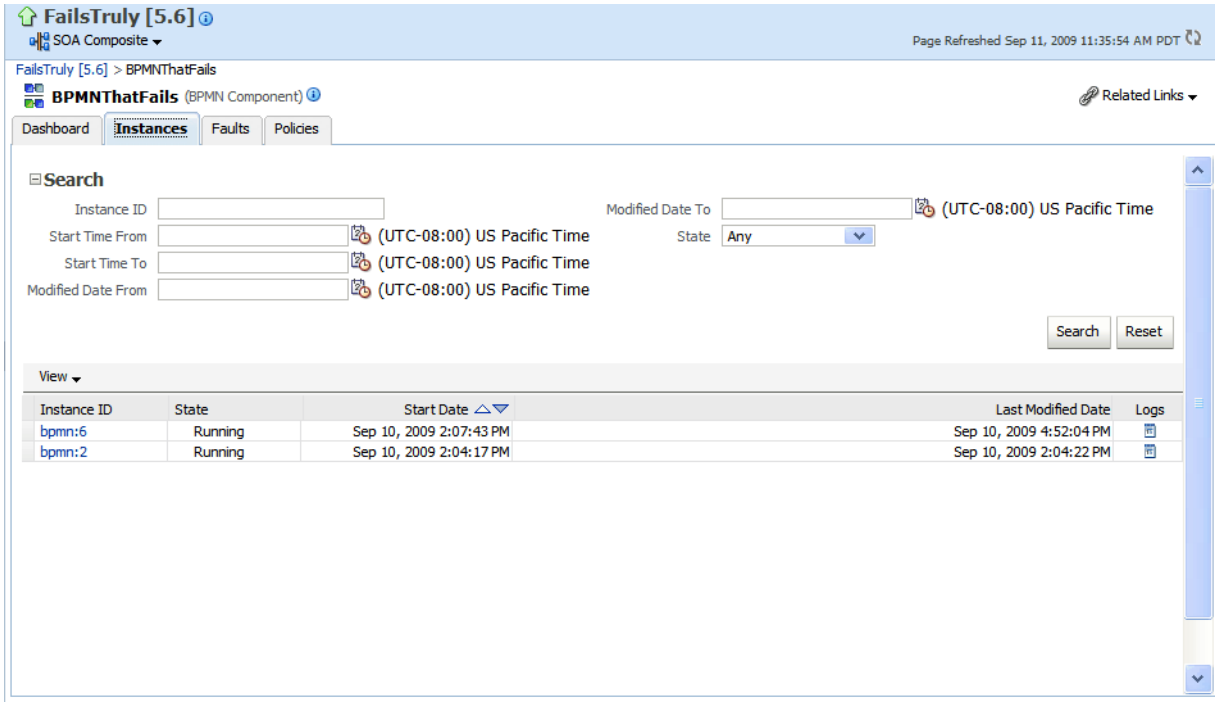
You can monitor BPMN process service component instances. Each service component has its own unique instance ID. This ID is in addition to the instance ID of the overall SOA composite application of which this service component is a part.

To monitor BPMN process service component instances:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, select a specific SOA composite application.

2. In the **Component Metrics** section, select the BPMN process service component.
3. Click the **Instances** tab.



The Instances page displays the following details:

- A utility for searching for a specific BPMN service component instance by specifying criteria and clicking **Search**.
- BPMN process service component instances, including the instance ID, instance state (for example, completed or faulted), instance start time, last instance modification time, and log files describing the instance.

In this page, you can perform the following tasks:

- View the audit trail, process flow, and faults of a service component. To do this task, in the **Instance ID** column, click the instance ID for a service component.
- Access the Log Messages page with filtered messages specific to that instance. To do this task, in the **Logs** column, click a specific log.

For more information, see Section 1.2.3, "Introduction to SOA Composite Application Instances."

38.4 Monitoring BPMN Process Service Engine Instances and Faults

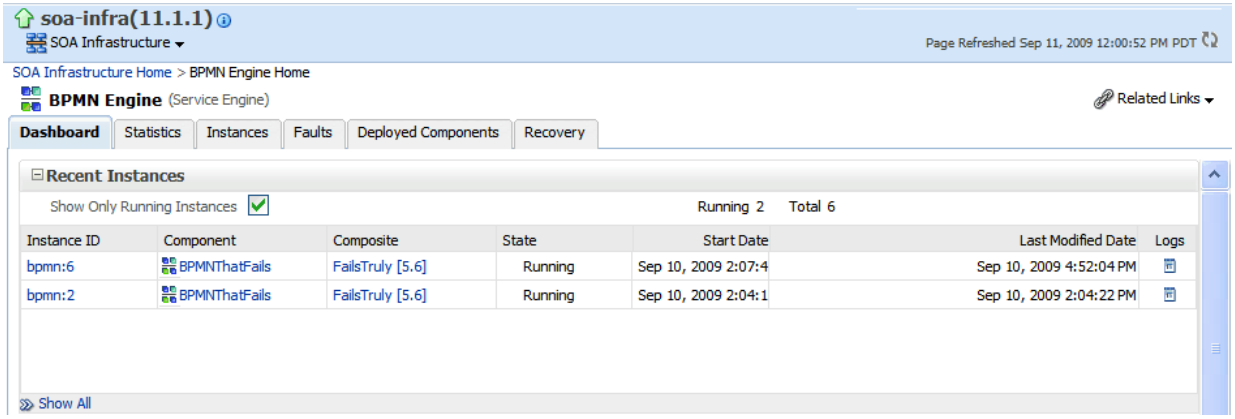
You can monitor instances and faults of all BPMN process service components running in the BPMN process service engine. These BPMN process service components can be part of separate SOA composite applications.

To monitor BPMN process service engine instances and faults:

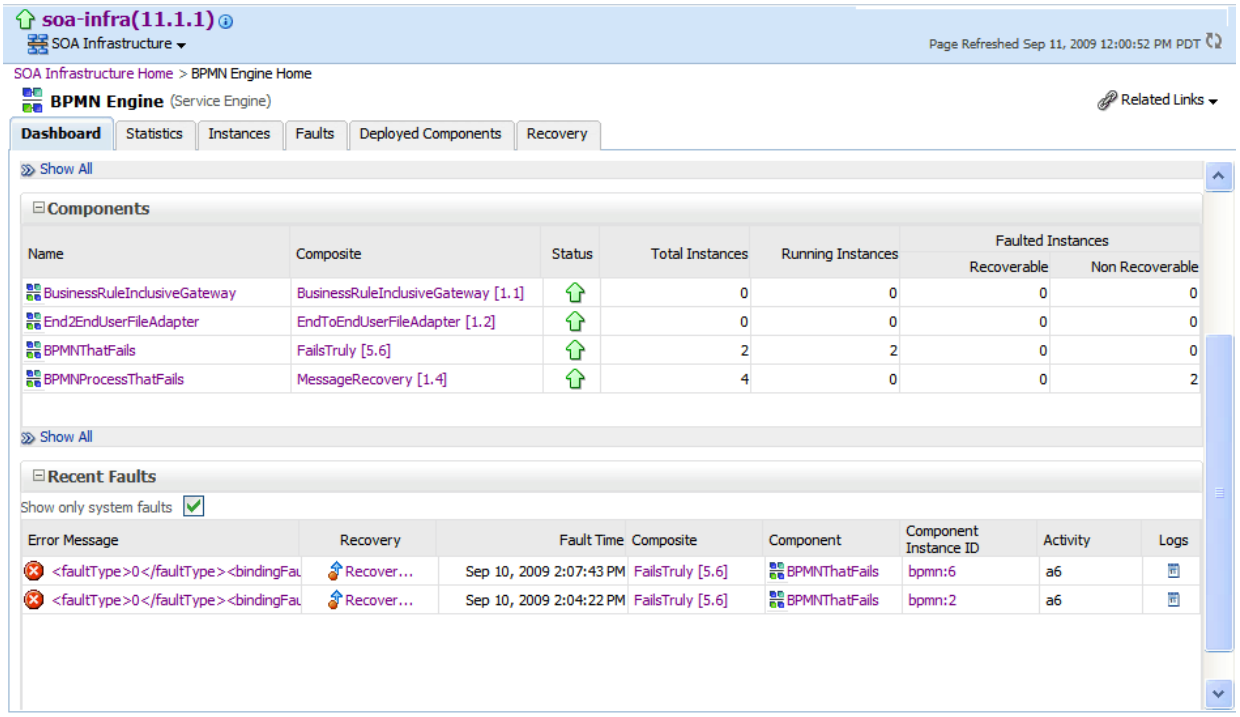
1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **BPMN**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **BPMN**.

2. Click **Dashboard**.



The upper part of the Dashboard page displays recent instances of all BPMN process service components running in the BPMN process service engine, including the instance ID of the service component, the service component name, the SOA composite application of which the service component is a part, the state of the instance (for example, completed successfully or faulted), the instance start time, the last modification time, and logs describing the instance.



The lower part of the Dashboard page displays the following details:

- The service components running in the service engine, the SOA composite applications of the service components, the state of the applications (for example, running), and the total, running, and faulted instances in the service engine.
- The recent faults in the service engine, including the error message, whether you can recover from the fault, the time at which the fault occurred, the SOA composite application in which the fault occurred, the service component, the

instance ID of the service component, the activity in which the fault occurred, and log files describing the fault.

3. In the **Recent Instances** section, you can perform the following monitoring tasks:
   - View the audit trail, process flow and faults of a service component. To do this task, in the **Instance ID** column, click an instance ID for the service component.
   - Access the home page of a service component. To do this task, in the **Component** column, click a specific service component.
   - Access its home page of a SOA composite application. To do this task, in the **Composite** column, click the specific SOA composite application.
   - Access the Log Messages page with filtered messages specific to an instance. To do this task, in the **Logs** column, click the specific log.
   - Access the Instances page of the service engine. To do this task, click **Show All** below the section.
4. In the **Components** section, you can perform the following tasks:
   - Access the home page of a specific service component. To do this task, in the **Name** column, click the specific service component.
   - Access the home page of a specific SOA composite application. To do this task, in the **Composite** column, click the specific SOA composite application.
   - Access the Deployed Components page of the service engine. To do this task, click **Show All** below the section.
5. In the **Recent Faults** section, you can perform the following tasks:
   - Display complete information about a fault. To do this task, in the **Error Message** column, click an error message. If the fault is identified as recoverable, click the **Recover Now** link to perform fault recovery.
   - Perform fault recovery at the component instance level. To do this task, in the **Recovery** column, click a fault identified as **Recoverable**.
   - Access the home page of a specific SOA composite application. To do this task, in the **Composite** column, click the specific SOA composite application.
   - Access the home page of a specific service component. To do this task, in the **Component** column, click a specific service component.
   - View the audit trail, process flow and faults of a service component. To do this task, in the **Component Instance ID** column, click an instance ID for a service component.
   - Access the Log Messages page with filtered messages specific to a fault. To do this task, in the **Logs** column, click the specific log.

For more information, see Section 1.2.4, "Introduction to Service Components and Service Component Instances."

38.5 Monitoring BPMN Process Service Engine Request and Thread Statistics

You can monitor request and thread statistics for all BPMN process service components running in the service engine.

To monitor BPMN process service engine request and thread statistics:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **BPMN**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **BPMN**.

2. Click **Statistics**.

   The upper part of the Statistics page displays the following details. Click the **Help** icon for additional details.

   - Pending requests in the service engine
   - Active requests in the service engine
   - Thread statistics for the service engine

   The lower part of the Statistics page displays details about the count and minimum, maximum, and average request processing times.

38.6 Monitoring BPMN Process Service Engine Instances

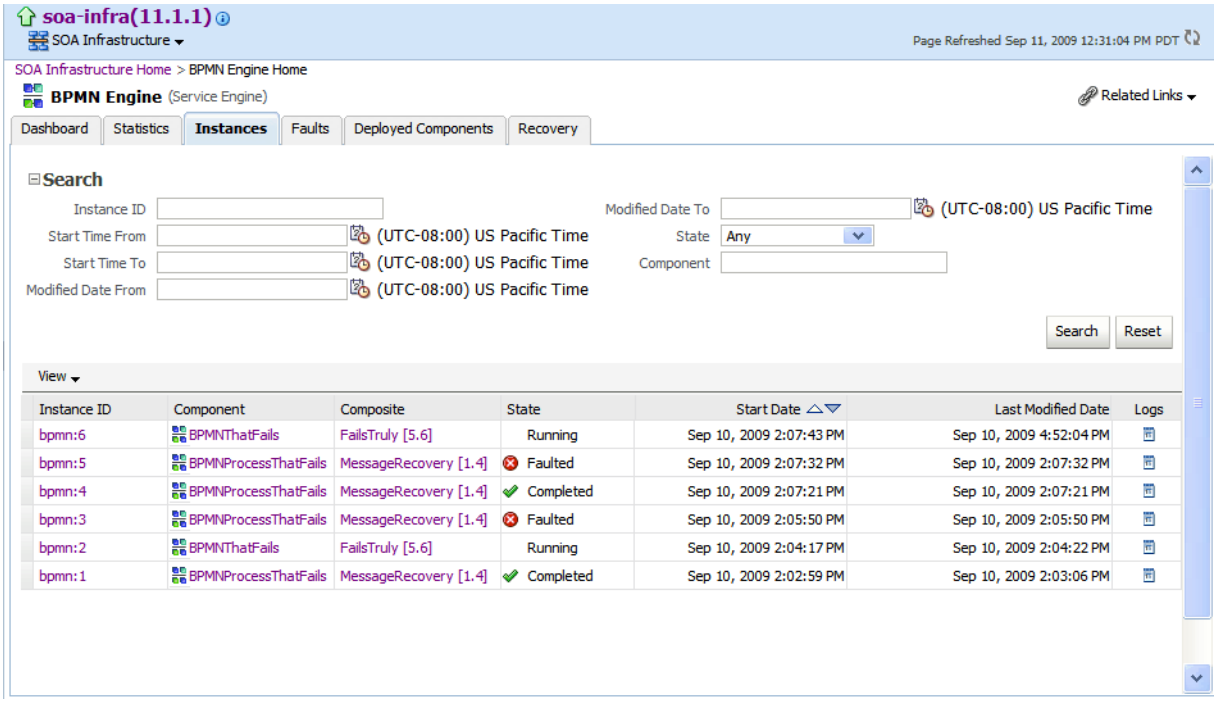
You can monitor all BPMN process service component instances running in the service engine. These BPMN process service components can be part of separate SOA composite applications.

To monitor BPMN process service engine instances:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **BPMN**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **BPMN**.

2. Click **Instances**.



The Instances page displays the following details:

- A utility for searching for a specific instance by specifying criteria and clicking **Search**.
- Instances, including the instance ID of the service component, the service component name, the SOA composite application name, the state of the instance (for example, completed successfully, running, or faulted), the instance start time, the last modification time, and log files describing the instance.

3. In the **Instances** section, you can perform the following monitoring tasks:
   - View the audit trail, process flow, sensor values, and faults of a service component. To do this task, in the **Instance ID** column, click an instance ID for a service component.
   - Access the home page of a specific service component. To do this task, in the **Component** column, click the specific service component.
   - Access the home page of a specific SOA composite application. To do this task, in the **Composite** column, click the specific SOA composite application.
   - Access the Log Messages page with filtered messages specific to an instance. To do this task, in the **Logs** column, click the specific log.

For more information, see Section 1.2.4, "Introduction to Service Components and Service Component Instances."

38.7 Monitoring Deployed BPMN Processes in the Service Engine

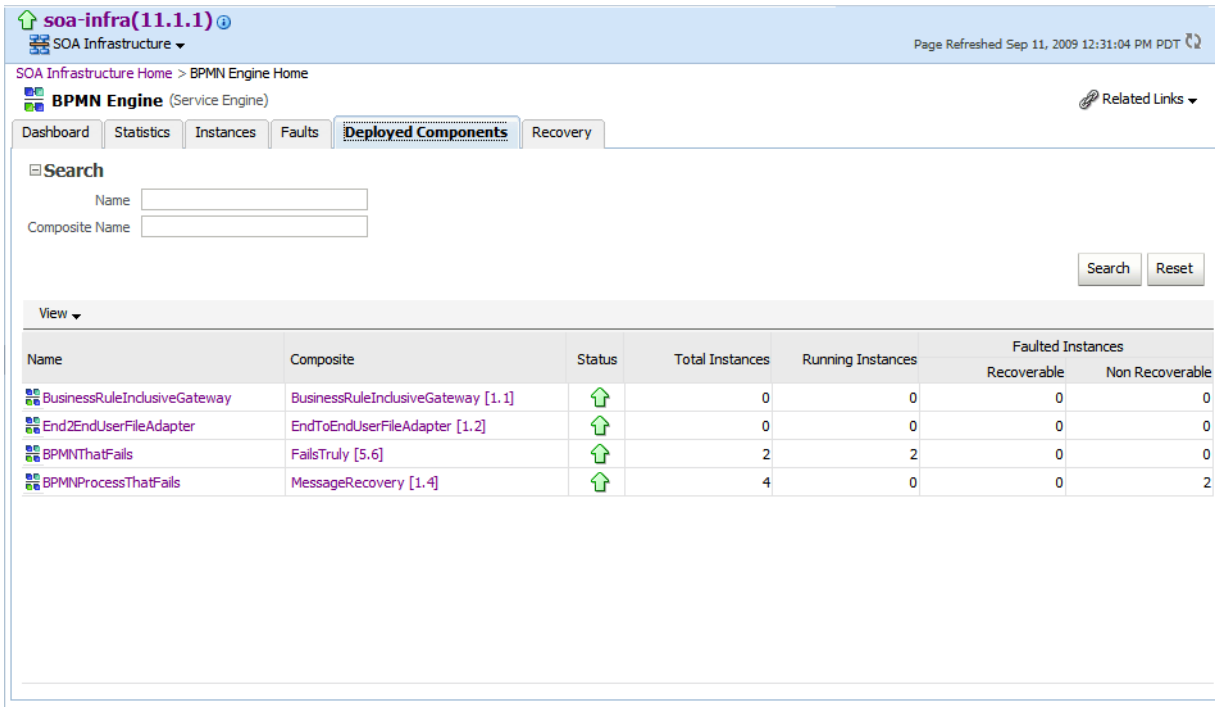
You can monitor all deployed SOA composite applications with BPMN process service components running in the service engine.

To monitor deployed BPMN processes in service engines:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **BPMN**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **BPMN**.

2. Click **Deployed Components**.



The Deployed Components page displays the following details:

- A utility for searching for a specific deployed SOA composite application by specifying criteria and clicking **Search**.
- Details about deployed SOA composite applications with BPMN process service components running in this service engine, including the service component name, the SOA composite application, the current status, and the total, running, and faulted instances in the service engine.

To access the home page of a specific service component, in the **Name** column, click the specific service component.

To access the home page of a specific SOA composite application, in the **Composite** column, click the specific SOA composite application.

39

Managing Oracle BPMN Service Components and Engines

This chapter describes how to manage BPMN process service components and service engines.

This chapter includes the following topics:

- Section 39.1, "Recovering from BPMN Process Service Component Faults"
- Section 39.2, "Managing BPMN Process Service Component Policies"
- Section 39.3, "Recovering from BPMN Process Service Engine Faults"
- Section 39.4, "Performing BPMN Process Service Engine Message Recovery"

39.1 Recovering from BPMN Process Service Component Faults

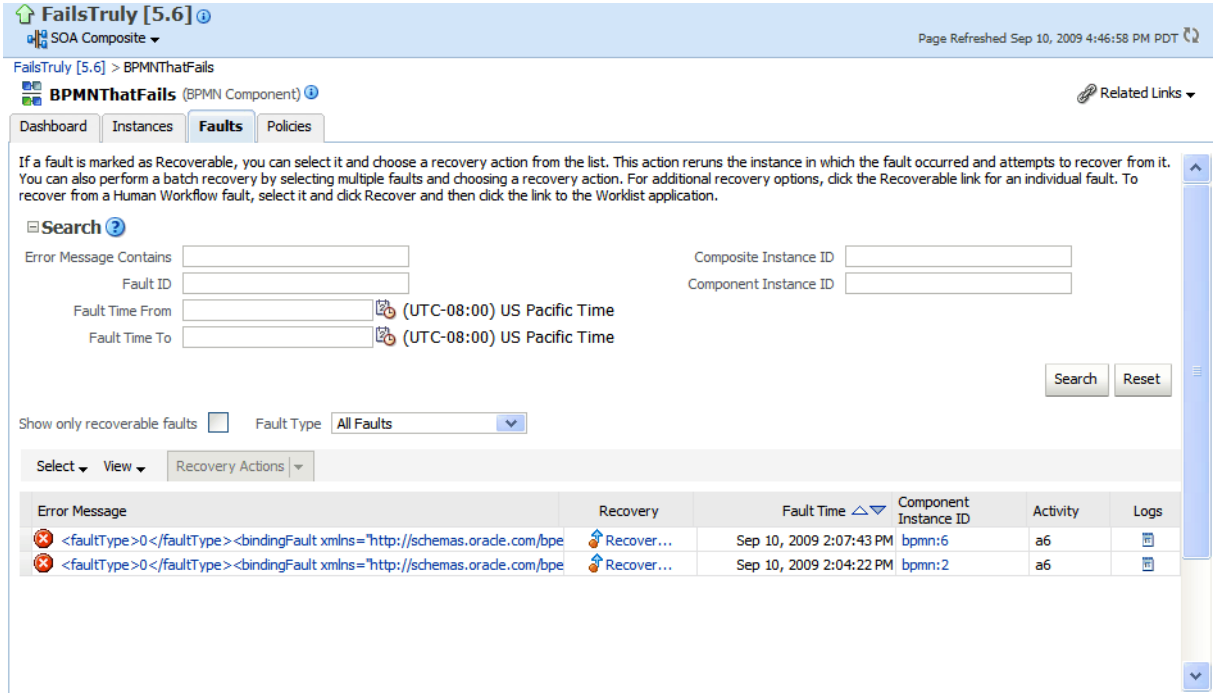
You can monitor and perform individual and bulk fault recoveries for BPMN process service components that are identified as recoverable. For BPMN process faults to be identified as recoverable, there must be a fault policy defined that is bound to the fault (through the `fault-bindings.xml` file) and which triggers the action `ora-human-intervention`. However, without defining any fault policies, the fault takes its normal course as either a recoverable or nonrecoverable fault.

To recover from BPMN process service component faults:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, select a specific SOA composite application.

2. In the **Component Metrics** section, select the BPMN process service component.
3. Click **Faults.**



The Faults page displays the following details:

- A utility for searching for a specific fault by specifying criteria and clicking **Search**. Click the **Help** icon for details.
- Faults that occurred in the service component, including the fault ID, error message, whether you can recover from the fault, time at which the fault occurred, service component instance ID, activity in which the fault occurred, and a link to a log file describing the fault.

BPMN process service component faults identified as recoverable can be recovered.

4. Select faults for recovery using one of the following methods. Fault recovery selection at the BPMN process service component level equals the SOA Infrastructure level, SOA composite application level, and Oracle Mediator service component level.

For...	Then...
Single fault recovery	There are three options from which to choose for single-fault recovery: **1.** Click the row of the fault that has been identified as recoverable. With the row highlighted, select a specific action from the **Recovery Action** list, as described in Step 5. **2.** In the **Recovery** column, click the **Recover** link to access the Faults page of the instance audit trail to perform fault recovery. **3.** In the **Error Message** column, click the message of a fault that has been identified as recoverable. This displays complete fault details, including the fault ID, fault time, fault location, fault type, and error message text. A **Recover Now** option is displayed for recoverable faults. Click **Recover Now** to access the Faults page of the instance audit trail to perform fault recovery.

For...	Then...
Bulk fault recovery	There are two options from which to choose for bulk-fault recovery: **1.** Use Shift+Click or Control+Click to select specific faults in the rows. or **2.** From the **Select** menu, choose **Select All Recoverable**. Then use Shift+Click or Control+Click to deselect the faults to *not* include in the recovery operation. Then: **3.** Select an action from the **Recovery Action** list, as described in Step 5. **Note:** Only the actions applicable to all selected faults are available.
Recovery of all faults	**1.** From the **Select** menu, choose **Select All Recoverable**. **2.** Select an action from the **Recovery Action** list, as described in Step 5. **Note:** Only the actions applicable to all selected faults are available.

Note: In most cases, fault policy actions are automatically executed. The only exception is if you defined a fault policy that uses the action `ora-human-intervention`. This action creates a recoverable fault that can be recovered from Oracle Enterprise Manager Fusion Middleware Control.

5. Select an action from the **Recovery Action** list.

Action	Description
Retry	Retries the instance directly. An example of a scenario in which to use this recovery action is when the fault occurred because the service provider was not reachable due to a network error. The network error is now resolved.
Abort	Terminates the entire instance.
Replay	Replays the entire scope again in which the fault occurred.
Rethrow	Rethrows the current fault. BPMN fault handlers (catch branches) are used to handle the fault. By default, all exceptions are caught by the fault management framework unless an explicit rethrow fault policy is provided.
Continue	Ignores the fault and continues processing (marks the faulting activity as a success).

6. Perform the following additional monitoring tasks from within the faults table:
   a. Click the **Show only recoverable faults** checkbox to display only faults from which you can recover.
   b. From the **Fault Type** list, select to display all faults, system faults, business faults, or Oracle Web Service Manager (OWSM) faults in the faults table. Click the **Help** icon for a description of these fault types.
   c. From the **View** list, select **Columns** > **Fault ID** to display the fault IDs for each error message. The fault ID is automatically generated and uniquely identifies a fault. The fault ID is also displayed when you click an error message.

d. In the **Component Instance ID** column, click a specific service component ID to access task details about the instance (for example, the current state of a task). Rejected messages do not have a component instance ID.

e. In the **Logs** column, click a link to access the Log Messages page with filtered messages specific to that instance.

For more information, see the following documentation:

- Section 1.4.3.1, "Introduction to Fault Recovery"
- Section 8.4.2, "Examples of Fault Recovery for BPMN Processes"
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

39.2 Managing BPMN Process Service Component Policies

You can attach and detach policies to and from BPMN process service components in currently deployed SOA composite applications. Policies apply security to the delivery of messages. Oracle Fusion Middleware uses a policy-based model to manage web services.

Note: Before attaching policies, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for definitions of available policies and details about which ones to use in your environment.

To manage BPMN process service component policies:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Home**. 2. Select the **Deployed Composites** tab. 3. In the **Composite** section, select a specific SOA composite application.	1. Under **soa-infra**, select a specific SOA composite application.

2. Select the BPMN process service component in the **Component Metrics** section.
3. Click **Policies**.

   The Policies page enables you to attach and detach policies to and from BPMN process service components. The policies table displays the attached policy name, the policy reference status (enabled or disabled) that you can toggle, the category (Management, Reliable Messaging, MTOM Attachment, Security, or WS Addressing), the violations, and the authentication, authorization, confidentiality, and integrity failures since the SOA Infrastructure was last restarted.
4. Click **Attach/Detach**.

   If multiple components are available, you are prompted to select the service or component for which to perform the attachment or detachment.
5. Select the service or component to which to attach or detach a policy.

   This invokes a dialog for attaching or detaching policies.

   Policies currently attached appear in the **Attached Policies** section. Additional policies available for attachment appear in the **Available Policies** section.

6. Select to attach policies appropriate to your environment.
7. Click **Attach**.
8. When you are finished attaching policies, click **Validate**.
9. If an error message appears, make the necessary corrections until you no longer have any validation errors.
10. Click **OK**.

    The attached policy is displayed in the policies table.

For more information, see the following documentation:

- Section 1.4.3.2, "Introduction to Policies"
- Section 7.6, "Managing SOA Composite Application Policies" for the dialogs that display during policy attachment.
- *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for definitions of available policies and details about which ones to use for your environment.

39.3 Recovering from BPMN Process Service Engine Faults

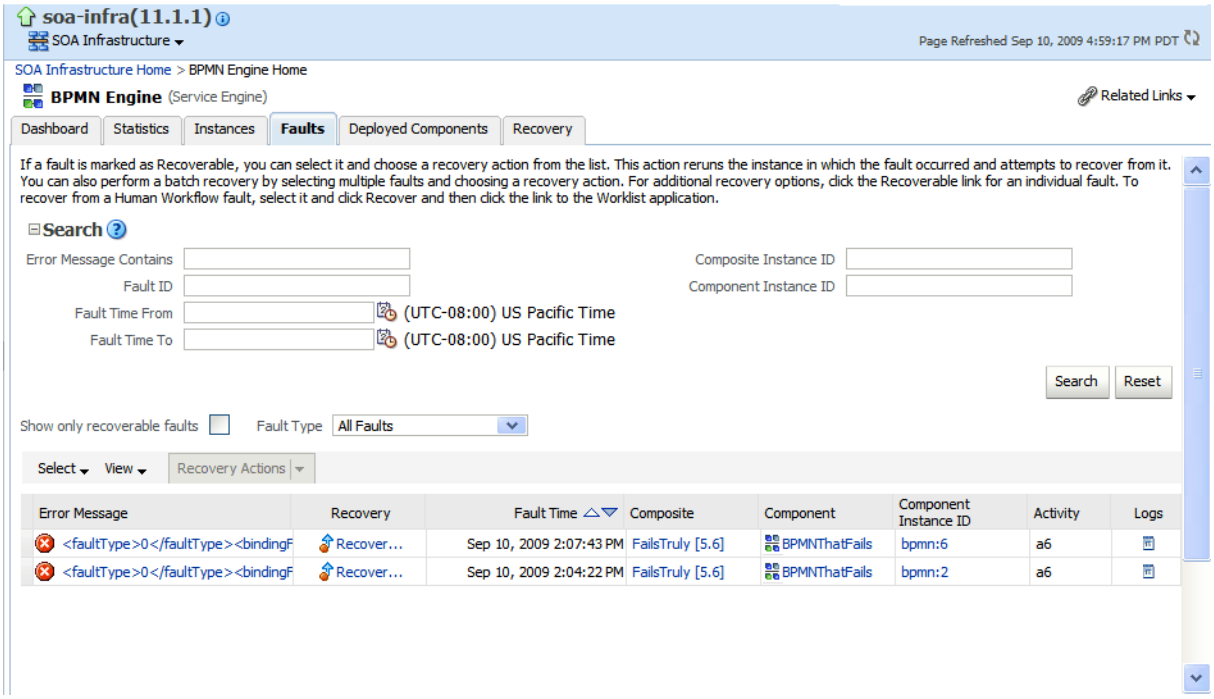
You can monitor and perform individual and bulk recoveries of faults occurring in BPMN process service engines that are identified as recoverable. All BPMN process service component faults, regardless of the SOA composite application instance of which they are a part, can be viewed in the BPMN process service engine. For BPMN process faults to be identified as recoverable, there must be a fault policy defined that is bound to the fault (through the `fault-bindings.xml` file) and which triggers the action `ora-human-intervention`. However, without defining any fault policies, the fault takes its normal course as either a recoverable or nonrecoverable fault.

To recover from BPMN process service engine faults:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **BPMN**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **BPMN**.

2. Select the **Faults** tab.



The Faults page displays the following details:

- A utility for searching for a specific fault by specifying criteria and clicking **Search**. Click the **Help** icon for details.
- Faults that occurred in the service engine, including the fault ID, error message, whether you can recover from the fault, the time at which the fault occurred, the SOA composite application and service component in which the fault occurred, and the service component instance ID.

BPMN process service engine faults identified as recoverable can be recovered.

3. As with fault recovery at the SOA Infrastructure level, SOA composite application level, and Oracle Mediator service component level, you can perform single fault recovery, bulk fault recovery, and recovery of all faults. See Step 4 of Section 39.1, "Recovering from BPMN Process Service Component Faults" for instructions on selecting faults to perform these types of recovery.

> **Note:** In most cases, fault policy actions are automatically executed. The only exception is if you defined a fault policy that uses the action `ora-human-intervention`. This action creates a recoverable fault that can be recovered from Oracle Enterprise Manager Fusion Middleware Control.

4. Select an action from the **Recovery Action** list.

Action	Description
Retry	Retries the instance with an option to provide a retry success action. An example of a scenario in which to use this recovery action is when the fault occurred because the service provider was not reachable due to a network error. The network error is now resolved.
Abort	Terminates the entire instance.

Action	Description
Replay	Replays the entire scope again in which the fault occurred.
Rethrow	Rethrows the current fault. BPMN fault handlers (catch branches) are used to handle the fault. By default, all exceptions are caught by the fault management framework unless an explicit rethrow fault policy is provided.
Continue	Ignores the fault and continues processing (marks the faulting activity as a success).

5. Perform the following additional monitoring tasks from within the faults table:
   a. Click the **Show only recoverable faults** checkbox to only display faults from which you can recover.
   b. From the **Fault Type** list, select to display all faults, system faults, business faults, or OWSM faults in the faults table. Click the **Help** icon for a description of these fault types.
   c. From the **View** list, select **Columns** > **Fault ID** to display the fault IDs for each error message. The fault ID is automatically generated and uniquely identifies a fault. The fault ID is also displayed when you click an error message.
   d. In the **Composite** column, click a specific SOA composite application to access its home page.
   e. In the **Component** column, click a specific service component to access its home page.
   f. In the **Component Instance ID** column, click a specific service component ID to access task details about the instance (for example, the current state of a task). Rejected messages do not have a component instance ID.

For more information, see the following sections:

- Section 1.4.3.1, "Introduction to Fault Recovery" for conceptual details about faults.
- Section 8.4.2, "Examples of Fault Recovery for BPMN Processes"

39.4 Performing BPMN Process Service Engine Message Recovery

You can perform a manual recovery of undelivered invoke or callback messages due to a transaction rollback in the process instance. Recovery of invoke messages applies to asynchronous BPMN processes only. Synchronous BPMN processes return an error to the calling client and are not recoverable from this page. Recoverable activities are activities that failed and can be recovered. For example, if you are using the file adapter to initiate an asynchronous BPMN process and your system crashes while the instance is processing, you can manually perform recovery when the server restarts to ensure that all message records are recovered.

Note: If you encounter the error message `ORA-01000: maximum open cursors exceeded`, then do the following:

1. Shut down the Oracle database.
2. Increase the value of `OPEN_CURSORS` to 1500.
3. Restart the Oracle database.

To perform BPMN process service engine message recovery:

1. Access this page through one of the following options:

From the SOA Infrastructure Menu...	From the SOA Folder in the Navigator...
1. Select **Service Engines** > **BPMN**.	1. Right-click **soa-infra**. 2. Select **Service Engines** > **BPMN**.

2. Click **Recovery**.

   The Recovery page displays the following details:

   - A utility for searching for a specific message failure by specifying criteria and clicking **Search**. Click the **Help** icon for details.
   - Message failure in the service engine, including the conversation ID, whether you can recover from the message failure, the service component and composite application in which the failure occurred, and the time at which the fault occurred.

3. Select a fault in the table.
4. Select one of the following options:

Action	Description
Recover	Retries the message in which the fault occurred. If an asynchronous BPMN process encounters a transaction rollback scenario because of any underlying exception error, it rolls back to the last dehydration activity. If this is a new instance, and a receive activity was the first dehydration activity, the BPMN process service engine creates a recoverable invoke. When you click **Recover** to recover the invoke, the service engine creates a new instance. This instance may run to completion with no exception error. However, you continue to see the older instance identified as faulted.
Mark Cancelled	Marks the message so it is never delivered.

Once a message is submitted for recovery, the BPMN process service engine may take time to complete the action. This typically takes less than several seconds. During this time, the message remains visible in the Recovery page. Duplicate attempts to recover the same message in that period are ignored. Refresh the page every few seconds to receive the latest recovery status.

Part XVI

Appendixes

This part includes the following appendixes:

- Appendix A, "Demo User Community"
- Appendix B, "Troubleshooting Oracle SOA Suite and Oracle BPM Suite"
- Appendix C, "Oracle Enterprise Manager Roles"

A

Demo User Community

This appendix describes how to install and use the organizational hierarchy of users and groups known as the demo user community in the database.

- Section A.1, "Using the Demo User Community"
- Section A.2, "Users"
- Section A.3, "Groups"
- Section A.4, "soa-infra Application Roles"
- Section A.5, "SOATestDemoApp Application Roles"
- Section A.6, "Roles Granted to and Owned by Users"
- Section A.7, "WorkflowPermission Class"

A.1 Using the Demo User Community

After installing Oracle SOA Suite, you must seed the user demo community in the database. The demo user community is an organizational hierarchy of users and groups.

To seed the demo user community:

1. Download the `workflow-001-DemoCommunitySeedApp` sample from the Oracle SOA Suite samples.
2. Follow the instructions in the `README.txt` file included with this sample. This file describes how to seed the user demo community described in this appendix.

> **Note:** You must run the script for seeding the user demo community locally from within a domain, and not remotely.

A.2 Users

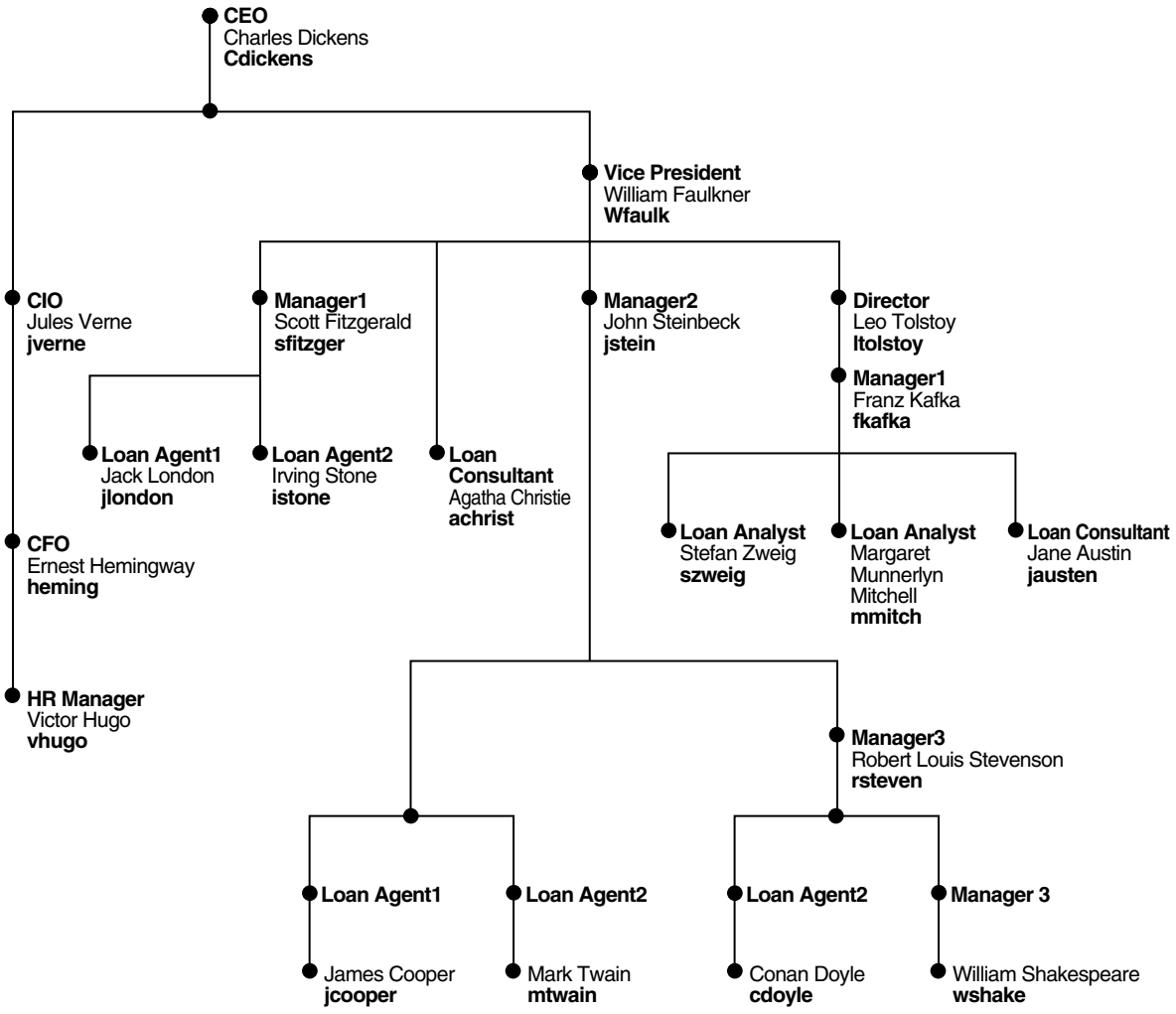
Table A–1 lists the users in the demo community.

Table A–1 Users in the Demo Community

User	User Name	First Name	Last Name	Title	Manager	Email
1	achrist	Agatha	Christie	Loan Consultant	wfaulk	achrist@emailExample.com
5	cdickens	Charles	Dickens	CEO	--	cdickens@emailExample.com
6	cdoyle	Conan	Doyle	Loan Agent 2	rsteven	cdoyle@emailExample.com
3	EHEMING	Ernest	Hemingway	CFO	JVerne	EHEMING@emailExample.com
7	fkafka	Franz	Kafka	Manager 1	ltolstoy	fkafka@emailExample.com
8	istone	Irving	Stone	Loan Agent 2	sfitzger	istone@emailExample.com
9	jausten	Jane	Austen	Loan Consultant	fkafka	jausten@emailExample.com
10	jcooper	James	Cooper	Loan Agent 1	jstein	jcooper@emailExample.com
11	jlondon	Jack	London	Loan Agent 1	sfitzger	jlondon@emailExample.com
12	jstein	John	Steinbeck	Manager 2	wfaulk	jstein@emailExample.com
2	JVerne	Jules	Verne	CIO	cdickens	JVerne@emailExample.com
13	ltolstoy	Leo	Tolstoy	Director	wfaulk	ltolsoy@emailExample.com
14	mmitch	Margaret	Mitchell	Loan Analyst	fkafka	mmitch@emailExample.com
15	mtwain	Mark	Twain	Loan Agent 2	jstein	mtwain@emailExample.com
16	rsteven	Robert	Stevenson	Manager 3	jstein	rsteven@emailExample.com
17	sfitzger	Scott	Fitzgerald	Manager 1	wfaulk	sfitzger@emailExample.com
18	szweig	Stefan	Zweig	Loan Analyst	fkafka	szweig@emailExample.com
4	VHUGO	Victor	Hugo	HR Manager	EHEMING	VHUGO@emailExample.com
19	wfaulk	William	Faulkner	Vice President	cdickens	wfaulk@emailExample.com
20	wshake	William	Shakespeare	Manager 3	rsteven	wshake@emailExample.com

Figure A–1 shows the organizational hierarchy of the demo community.

Figure A–1 Demo Community Organizational Hierarchy



A.3 Groups

Table A–2 lists the following:

- Groups in the demo community
- Users and groups that are granted each group role (direct grantees and all grantees)
- Group roles and application roles granted to each group (direct-granted roles and all granted roles)

See Table A–6 for the roles granted to users sorted by user.

Table A–2 Groups in the Demo Community: Grant Relationships

Group	Direct Grantees	All Grantees	Direct-Granted Roles	All Granted Roles
RegionalOffices	CentralRegion, WesternRegion, EasternRegion	szweig, wshake, jcooper, WesternRegion, mmitch, EasternRegion, jlondon, CentralRegion, istone, cdoyle, mtwain, California, fkafka	-	-
EasternRegion	szweig, wshake, mmitch, fkafka	szweig, wshake, mmitch, fkafka	RegionalOffices	RegionalOffices
CentralRegion	jlondon, mtwain	jlondon, mtwain	RegionalOffices	RegionalOffices
WesternRegion	cdoyle, California	jcooper, istone, cdoyle, California	RegionalOffices	RegionalOffices
California	jcooper, istone	jcooper, istone	WesternRegion	RegionalOffices, WesternRegion
LoanAgentGroup	jlondon, wshake, LoanAnalyticGroup , jcooper, istone, cdoyle, mtwain	szweig, jlondon, wshake, LoanAnalyticGroup, jcooper, istone, cdoyle, mtwain, mmitch, fkafka	-	-
LoanAnalyticGroup	szweig, mmitch, fkafka	szweig, mmitch, fkafka	BPMWorkflowCustomize, LoanAgentGroup	BPMWorkflowCustomize, LoanAgentGroup
Supervisor	jcooper, mtwain, rsteven	jcooper, mtwain, rsteven	-	-
Executives	cdickens, JVerne, EHEMING, VHUGO	cdickens, JVerne, EHEMING, VHUGO	-	-

Table A–3 continues information for several groups listed in Table A–2. It lists the following:

- Users and groups (direct owners and all owners) that own each group
- Group roles (direct-owned roles and all owned roles) that each group owns

See Table A–6 for the roles owned by users sorted by user.

Table A–3 Groups in the Demo Community: Ownership Relationships

Group	Direct Owners	All Owners	Direct Owned Roles	All Owned Roles
EasternRegion	jstein	jstein	-	-
WesternRegion	jstein	jstein	-	-
California	fkafka	fkafka	-	-
LoanAgentGroup	jcooper, fkafka	jcooper, fkafka	-	-
LoanAnalyticGroup	jstein	jstein	-	-
Supervisor	jstein	jstein	-	-

A.4 soa-infra Application Roles

Table A–4 lists the following:

- `soa-infra` application roles
- Users, groups, and roles that are granted each application role (direct grantees and all grantees)

- Roles granted to each application role (direct-granted roles and all granted roles)

See Table A–6 for the application roles granted to users sorted by user.

Table A–4 Application Roles in soa-infra

Application Role	Direct Grantees	All Grantees	Direct-Granted Roles	All Granted Roles
SOAAdmin	Administrators	Administrators	BPMWorkflowAdmin	BPMWorkflowCustomize BPMWorkflowAdmin
BPMWorkflowAdmin	SOAAdmin, demoadmin	SOAAdmin, demoadmin, Administrators	BPMWorkflowCustomize	BPMWorkflowCustomize
BPMWorkflowCustomize	LoanAnalyticGroup, BPMWorkflowAdmin	szweig, LoanAnalyticGroup, SOAAdmin, BPMWorkflowAdmin , mmitch, fkafka, Administrators, demoadmin	-	-

For more information about application roles, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

A.5 SOATestDemoApp Application Roles

Table A–5 lists the roles in the SOATestDemoApp application.

Table A–5 SOATestDemoApp Roles

Application Role	Direct Grantees	All Grantees	Direct-Granted Roles	All Granted Roles
DevTeam	rsteven	rsteven, mmitch, fkafka, jcooper, istone		
QATeam	jlondon, Supervisor	jlondon, jcooper, mtwain, rsteven, mmitch, fkafka, istone		
ProductionTeam	mmitch, fkafka, California	mmitch, fkafka, jcooper, istone	DevTeam, QATeam	DevTeam, QATeam

A.6 Roles Granted to and Owned by Users

Table A–6 lists the following:

- Roles granted to each user (direct-granted roles and all granted roles)
- Roles owned by each user (direct-owned roles and all owned roles)

Table A–6 Roles for Each User

User Name	Direct-Granted Roles	All Granted Roles	Direct-Owned Roles	All Owned Roles
achrist	Executives	-	-	-
cdickens	-	-	Executives	-
cdoyle	WesternRegion, LoanAgentGroup	RegionalOffices, WesternRegion, LoanAgentGroup	-	-
EHEMING		Executives		

Table A–6 (Cont.) Roles for Each User

User Name	Direct-Granted Roles	All Granted Roles	Direct-Owned Roles	All Owned Roles
fkafka	LoanAnalyticGroup, EasternRegion, ProductionTeam	BPMWorkflowCustomize, RegionalOffices, LoanAnalyticGroup, LoanAgentGroup, EasternRegion, ProductionTeam, DevTeam, QATeam	LoanAgentGroup, California	LoanAgentGroup, California
istone	LoanAgentGroup, California	RegionalOffices, WesternRegion, LoanAgentGroup, California, DevTeam, QATeam, ProductionTeam	-	-
jausten	-	-	-	-
jcooper	Supervisor, LoanAgentGroup, California	RegionalOffices, WesternRegion, Supervisor, LoanAgentGroup, California, DevTeam, QATeam, ProductionTeam	LoanAgentGroup	LoanAgentGroup
jlondon	CentralRegion, LoanAgentGroup, QATeam	CentralRegion, RegionalOffices, LoanAgentGroup, QATeam	-	-
jstein	-	-	LoanAnalyticGroup WesternRegion, Supervisor, EasternRegion	LoanAnalyticGroup WesternRegion, Supervisor, EasternRegion
JVerne		Executives		
ltolstoy	-	-	-	-
mmitch	LoanAnalyticGroup, EasternRegion, ProductionTeam	BPMWorkflowCustomize RegionalOffices, LoanAnalyticGroup, LoanAgentGroup, EasternRegion, DevTeam, QATeam, ProductionTeam	-	-
mtwain	CentralRegion, Supervisor, LoanAgentGroup	CentralRegion, RegionalOffices, Supervisor, LoanAgentGroup, QATeam	-	-
rsteven	Supervisor, DevTeam	Supervisor, DevTeam	-	-
sfitzger	-	-	-	-
szweig	LoanAnalyticGroup, EasternRegion	BPMWorkflowCustomizeDemoApp/FlexFieldRole, RegionalOffices, LoanAnalyticGroup, LoanAgentGroup, EasternRegion		
vhugo		Executives		
wfaulk	-	-	-	-
wshake	LoanAgentGroup, EasternRegion	RegionalOffices, LoanAgentGroup, EasternRegion	-	-

A.7 WorkflowPermission Class

Table A–7 lists the permissions defined in the `WorkflowPermission` class and the application roles associated with each permission.

Table A–7 WorkflowPermission Class

Permission	Application Role with Permission
workflow.mapping.publicFlexField	BPMWorkflowAdmin, BPMWorkflowCustomize
workflow.mapping.protectedFlexField	BPMWorkflowAdmin
workflow.admin	BPMWorkflowAdmin
workflow.admin.evidenceStore	BPMWorkflowAdmin

B

Troubleshooting Oracle SOA Suite and Oracle BPM Suite

This appendix describes how to troubleshoot issues you can encounter when using Oracle SOA Suite and Oracle BPM Suite, including server startup best practices, page loading optimizations, message failures, tablespace issues, connection timeouts, event delivery network issues, and human workflow issues. It also describes how to set logging levels for troubleshooting.

This appendix includes the following topics:

- Section B.1, "Setting Logging Levels for Troubleshooting"
- Section B.2, "Connection and Transaction Timeout Troubleshooting"
- Section B.3, "Runtime Diagnostics Troubleshooting"
- Section B.4, "Human Workflow Troubleshooting"
- Section B.5, "Event Delivery Network Troubleshooting"
- Section B.6, "Performance Troubleshooting"
- Section B.7, "Server Troubleshooting"
- Section B.8, "Browser Troubleshooting"
- Section B.9, "Additional Troubleshooting Documentation"

B.1 Setting Logging Levels for Troubleshooting

To simplify troubleshooting, it is recommended that you set logging levels to the **TRACE:32 FINEST** level in Oracle Enterprise Manager Fusion Middleware Control. This section describes loggers to which to set to this level.

To set logging levels for troubleshooting:

1. See Section 3.4, "Configuring Log Files" for instructions on accessing the Log Configuration page.
2. From the **Oracle Diagnostic Logging Level (Java Level)** list, set the following parent loggers to the **TRACE:32 FINEST** level:
   - **oracle.soa**
   - **oracle.fabric**
   - **oracle.integration**
   - **oracle.wsm** (Setting this logger to the `ERROR` level may also be sufficient because this setting logs the required error messages.)

3. If you want finer-grained control over logging, expand the parent loggers and set any of the following loggers:

Component	Logger
Human workflow/approval management extensions (AMX)/rules	▪ **oracle.soa.services.common** ▪ **oracle.soa.services.identity** ▪ **oracle.soa.services.notification** ▪ **oracle.soa.services.rules** ▪ **oracle.soa.services.rules.obrtrace** ▪ **oracle.soa.services.workflow** ▪ **oracle.soa.services.workflow.common** ▪ **oracle.soa.services.workflow.evidence** ▪ **oracle.soa.services.workflow.metadata** ▪ **oracle.soa.services.workflow.persistency** ▪ **oracle.soa.services.workflow.query** ▪ **oracle.soa.services.workflow.report** ▪ **oracle.soa.services.workflow.runtimeconfig** ▪ **oracle.soa.services.workflow.soa** ▪ **oracle.soa.services.workflow.task** ▪ **oracle.soa.services.workflow.task.dispatch** ▪ **oracle.soa.services.workflow.task.routing** ▪ **oracle.soa.services.workflow.user** ▪ **oracle.soa.services.workflow.verification** ▪ **oracle.soa.services.workflow.performance** - Workflow Performance API ▪ **oracle.soa.services.workflow.worklist** - Oracle BPM Worklist
SOA Infrastructure	▪ **oracle.fabric.common.wsdl** - WSDL/schema management ▪ **oracle.integration.platform.blocks.deploy** - Deployment ▪ **oracle.integration.platform.blocks.soap** - WS binding - Calling web services ▪ **oracle.integration.platform.blocks.local** - Local binding ▪ **oracle.integration.platform.kernel** - Startup issues ▪ **oracle.integration.plaform.blocks.mesh** - Message routing ▪ **oracle.integration.platform.common** - Metadata/MDS ▪ **oracle.integration.platform.instance** - Instance ▪ **oracle.integration.platform.instance.activity** - Instance ▪ **oracle.integration.platform.instance.store** - Instance
Event Delivery Network (EDN)	▪ **oracle.integration.platform.blocks.event** ▪ **oracle.integration.platform.blocks.event.saq** ▪ **oracle.integration.platform.blocks.event.jms**
Deployment	**oracle.integration**

Component	Logger
Oracle Mediator	▪ **oracle.soa.mediator.common** - Logs the processing events related to Oracle Mediator audit message persistence. ▪ **oracle.soa.mediator.common.cache** - Metadata cache (`RuntimeMetadataCache`) and runtime related cache (`RuntimeCache`). ▪ **oracle.soa.mediator.common.error** - Logs related to error enqueuing and handling. ▪ **oracle.soa.mediator.common.error.recovery** - Logs only the error recovery processing. All fault policy handlers are part of this logger. ▪ **oracle.soa.mediator.common.listener** - Processing related to common infrastructure for Oracle Mediator parallel routing rules and resequencer. ▪ **oracle.soa.mediator.common.message** - Logs related to creation and modification of the Oracle Mediator message. (Oracle Mediator wraps a normalized message into the Oracle Mediator message.) ▪ **oracle.soa.mediator.common.persistence** - Logs related to the persistence of Oracle Mediator deferred messages to the database. ▪ **oracle.soa.mediator.dispatch** - Logs everything related to message routing inside Oracle Mediator. This includes sequential, parallel, and dynamic routing. ▪ **oracle.soa.mediator.dispatch.db** - Logs related to the deferred message and container ID infrastructure. ▪ **oracle.soa.mediator.dispatch.resequencer.toplink** - Logs related to database interactions of the resequencer. This includes execution of all three resequencer strategies. ▪ **oracle.soa.mediator.monitor** - Logs all events related to instance tracking in Oracle Mediator. ▪ **oracle.soa.mediator.resequencer** - Everything related to Oracle Mediator resequencer processing such as locker-worker infrastructure, group, and sequencer ID expression evaluation. ▪ **oracle.soa.mediator.resequencer.besteffort** - For best effort strategy of the resequencer; in particular, the locking stage processing. ▪ **oracle.soa.mediator.resequencer.fifo** - For the FIFO strategy of the resequencer. ▪ **oracle.soa.mediator.resequencer.standard** - For standard strategy of the resequencer. ▪ **oracle.soa.mediator.service** - Logs events related to the processing of various message exchange patterns (one way, two way (synchronous, asynchronous), and so on) in Oracle Mediator. ▪ **oracle.soa.mediator.service.common.functions** - Oracle Mediator XPath extension function-related logs. ▪ **oracle.soa.mediator.service.filter** - Evaluation of filter criteria for routing rules. ▪ **oracle.soa.mediator.service.transformation** - Logs transformation-related events such as the payload before and after the transformation. ▪ **oracle.soa.mediator.serviceEngine** - Logs lifecycle events of the Oracle Mediator service engine. Also logs the entry and exit of messages for Oracle Mediator.

Component	Logger
Oracle BPEL Process Manager	▪ **oracle.soa.bpel** ▪ **oracle.soa.bpel.console** ▪ **oracle.soa.bpel.engine** ▪ **oracle.soa.bpel.engine.activation** ▪ **oracle.soa.bpel.engine.agents** ▪ **oracle.soa.bpel.engine.bpel** ▪ **oracle.soa.bpel.engine.compiler** ▪ **oracle.soa.bpel.engine.data** ▪ **oracle.soa.bpel.engine.delivery** ▪ **oracle.soa.bpel.engine.deployment** ▪ **oracle.soa.bpel.engine.dispatch** ▪ **oracle.soa.bpel.engine.sensor** ▪ **oracle.soa.bpel.engine.translation** ▪ **oracle.soa.bpel.engine.ws** ▪ **oracle.soa.bpel.engine.xml** ▪ **oracle.soa.bpel.entity** ▪ **oracle.soa.bpel.jpa** ▪ **oracle.soa.bpel.system**
Oracle BPM Suite	**oracle.bpm.bpmn.engine**
Oracle B2B	▪ **oracle.soa.b2b.apptransport** - Runtime logger ▪ **oracle.soa.b2b.engine** - Runtime logger ▪ **oracle.soa.b2b.transport** - Runtime logger ▪ **oracle.soa.b2b.ui** - User interface logger ▪ **oracle.soa.b2b.repository** - Repository access detailed logger
Oracle adapters	**oracle.soa.adapter**
Oracle Web Services Manager (OWSM)	**oracle.wsm** - Defaulted to the **ERROR** level; logs all `WSM-0xxxx` errors. **TRACE:32** results in significant details.

4. From the **Oracle Diagnostic Logging Level (Java Level)** list, change the logger level to one of the following settings:
   - **TRACE:1 (FINE)**
   - **TRACE:16 (FINER)**
   - **TRACE:32 (FINEST)** - Most verbose level (recommended for troubleshooting)

   The change takes effect within several minutes.

For information about monitoring Oracle Business Activity Monitoring (BAM) logs, see Section 24.6, "Monitoring Oracle BAM Logs." For information about monitoring Oracle User Messaging Service (UMS) logs, see Section 27.2, "Viewing Log Files."

B.1.1 Log Files and Thread Dumps from All Managed Servers

Table B–1 describes the log files to view and thread dumps to obtain.

Table B–1 Log Files and Thread Dumps

Output	Description
Server diagnostic log	View the following file: `$DOMAIN_HOME/servers/server_name/logs/server_name-diagnostic.log` For example, `soa_server1-diagnostic.log`, if *server_name* is `soa_server1`. This is where the log output is available. By default, only the last 100 MB of the diagnostic logs are retained.
Server log	*server_name*.log (for example, `soa_server1.log`, if *server_name* is `soa_server1`)
Server console output	`<stdout>` is also useful, especially for deployment and patching issues.
Server thread dump	Enter the following at the operating system command prompt: `kill -3 managed_server_process_ID` You can also use Oracle WebLogic Server Administration Console. 1. In the navigation tree of Oracle WebLogic Server Administration Console, select **Environment** > **Servers**. 2. In the table, select the server. 3. Select the **Monitoring** tab. 4. Select the **Threads** tab. 5. Click **Dump Thread Stacks**. The output is in the console logs.
OWSM message log	The following log captures all SOAP messages on the wire. `$DOMAIN_HOME/servers/server_name/logs/owsm/msglogging/diagnostic.log` This log is not enabled by default. To enable this log: 1. Go to **Fusion Middleware Control** > **Weblogic Domain** > **Web Services** > **Policies**. 2. Select the security level for which to enable logging. 3. Edit the policy to enable the log assertion.

B.2 Connection and Transaction Timeout Troubleshooting

This section describes how to troubleshoot connection and transaction timeout issues.

B.2.1 Resolving Connection Timeouts

You can receive a connection timeout error under circumstances such as the following:

- You run a SOA composite application with a large payload that takes more than 30 seconds to process.
- You are invoking a stress test using a large payload from the Test Web Service page of Oracle Enterprise Manager Fusion Middleware Control.
- You are passing a large number of message files (one million) into a composite with a file adapter service.
- You are retrieving instance and fault count metrics in Oracle Enterprise Manager Fusion Middleware Control.

To avoid receiving timeout errors, increase the transaction timeout property as follows:

1. Log into Oracle WebLogic Administration Console.
2. Click **JTA**.
3. Change the value of **Timeout Seconds** (the default is `30`).
4. Click **Save**.
5. Restart Oracle WebLogic Server.

B.2.2 Increasing Database Connection Values

You can receive the following error message because of slow connections to the database.

```
Exception [TOPLINK-4002] (Oracle TopLink - 11g Release 1 (11.1.1.1.0) (Build
090304)): oracle.toplink.exceptions.DatabaseException
Internal Exception: java.sql.SQLException: Internal error: Cannot obtain
XAConnection weblogic.common.resourcepool.ResourceDeadException: Pool
SOADataSource has been disabled because of hanging connection tests, cannot
allocate resources to applications.
```

If this occurs, perform the following steps:

1. Open the *DOMAIN_HOME*`\bin\setSOADomainEnv.cmd` file.
2. Uncomment the lines shown in bold.

```
# 8331492: Value of weblogic.resourcepool.max_test_wait_secs is 10
# seconds. It can be increased by uncommenting line below if your database
# connections are slow. See SOA documentation for more details.
 EXTRA_JAVA_PROPERTIES="${EXTRA_JAVA_PROPERTIES}
-Dweblogic.resourcepool.max_test_wait_secs=30"
export EXTRA_JAVA_PROPERTIES
```

3. Save your changes and restart the managed Oracle WebLogic Server.

B.2.3 Updating the EJB Transaction Timeout Value in the Deployment Archive After SOA Infrastructure Failure

Updating the transaction time out value for the **FacadeFinderBean** property in Oracle WebLogic Server Administration Console under **Deployments** > **expanded SOA Infrastructure Application** > **FacadeFinderBean** > **Configuration tab** can result in the following error after restarting the SOA Infrastructure:

```
java.lang.IllegalArgumentException: Cannot convert value of type [$Proxy223
implementing
 oracle.bpel.services.workflow.verification.IVerificationService,org.springframe
work.aop.SpringProxy,org.springframework.aop.framework.Advised]
to required type
[oracle.bpel.services.workflow.verification.IVerificationService] for
property 'verificationService': no matching editors or conversion strategy found
Message icon - Warning Errors were encountered while performing this operation.
```

The SOA Infrastructure status is also displayed as failed.

This error is not specific to **FacadeFinderBean**; it also applies to any EJB that is part of the SOA Infrastructure application.

To resolve this error, you must manually modify the transaction timeout setting in your deployment archive.

To update the transaction timeout setting:

1. Open the `fabric-ejb.jar` file in your deployment archive.
2. Increase the transaction timeout value in the `META-INF/weblogic-ejb-jar.xml` file to a larger value.
3. Rejar the file.
4. Restart the managed server that includes the SOA Infrastructure by following the instructions in Section 3.2, "Stopping and Starting the Managed Server and SOA Infrastructure."

Note: This issue may also occur while updating any EJBs deployed as part of the SOA Infrastructure application. If this issue occurs, you must update the corresponding contained JAR file for those EJBs in a similar fashion.

B.2.4 Long Running, Synchronous Calls To Remote Web Services Error Out or Asynchronous Transactions Return with an Error after a Long Time

You may have long running, synchronous calls to remote web services that end with JTA transaction rolled-back errors. When executing a transaction making an asynchronous call (for example, to the SOA server), the application may return with an error. The server logs show JTA transaction timeouts, which can cause this behavior.

To check the JTA transaction timeout in the Oracle WebLogic Server Administration Console.

1. Log in to Oracle WebLogic Server Administration Console.
2. In the **Domain Structure**, select **Services** > **JTA** to check the timeout value.

   If the transaction is always timing out beyond 30 seconds and is a custom composite synchronous client invocation, then you may need to revisit the design approach. It may be best for the external web service to be invoked as an asynchronous transaction.

   Increasing the JTA for supporting long running, synchronous transactions is simply an interim mechanism. For information on changing the JTA transaction timeout setting, see Section B.2.1, "Resolving Connection Timeouts."
3. If synchronous client invocations take a long time, check for any performance issues with the system and try to resolve them.
4. Check for appropriate values for the **syncMaxWaitTime** property and BPEL's EJB transaction timeout settings in relation to the JTA timeout settings and only then increase the value of the JTA timeout, if needed.

   For information on viewing and changing the **syncMaxWaitTime** property, see Section 12.1, "Configuring BPEL Process Service Engine Properties."
5. To view and change the BPEL EJB transaction timeout settings, perform the following steps:

   a. Log in to Oracle WebLogic Server Administration Console.

   b. In the **Domain Structure**, click **Deployments**.

c. Expand **soa-infra** > **EJBs**.

d. Update the following EJBs:

- **BPELActivityManagerBean**
- **BPELDeliveryBean**
- **BPELDispatcherBean**
- **BPELEngineBean**
- **BPELFinderBean**
- **BPELInstanceManagerBean**
- **BPELProcessManagerBean**
- **BPELSensorValuesBean**
- **BPELServerManagerBean**

e. Click **Save**.

f. Restart Oracle WebLogic Server.

6. For asynchronous transactions, check the values for both the BPEL EJB transaction timeout and the JTA transaction timeout and adjust as needed.

B.3 Runtime Diagnostics Troubleshooting

This section describes how to troubleshoot runtime issues.

B.3.1 Oracle SOA Suite Runtime Failure with a "Cannot read WSDL" Error

The following error is displayed if either the endpoint is not available for a reference or a composite is deployed with an incorrect deployment plan file.

```
oracle.fabric.common.FabricException: Cannot read WSDL
```

To diagnose this issue:

1. Ensure that the endpoint for the reference is up and running.

   The WSDL or endpoint is stated in the error. The service can then be looked up from Oracle Enterprise Manager Fusion Middleware Control to check if it is active. Once the service is active and the endpoint is reachable, search for the SOA instance and retry it through the **Recovery** tab for the BPEL process service engine in Oracle Enterprise Manager Fusion Middleware Control.

2. For extensions/customizations, ensure that the correct URL is updated in the deployment configuration plan.

   For information about deployment plans, including examples of using `sca_extractPlan` to extract plans, see the "Customizing Your Application for the Target Environment Prior to Deployment" section of *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

B.3.2 Automatic Recovery of BPEL Instances is Not Recovering A Specific Instance

BPEL processes have an automatic recovery feature that attempts to automatically recover activities that are recoverable such as unresolved invoke/callback messages, activities not completed over a provided threshold time, and so on. However, the automatic recovery feature only tries to recover a few instances and only retries a fixed

number of times. If some instances are not being automatically recovered, they are likely not being picked up because of the configuration of the automatic recovery parameters.

To resolve this issue:

- Set the maximum number of messages to automatically recover.

  By default, the automatic recovery feature of Oracle BPEL Process Manager processes 50 messages to submit for each recovery attempt. This is controlled by the **maxMessageRaiseSize** property.

  1. In the navigation pane, right-click **soa-infra (*SOA_cluster_name*)**.
  2. Select **SOA Administration** > **BPEL Properties** > **More BPEL Configuration Properties** > **RecoveryConfig**.
  3. Expand both **RecurringScheduleConfig** > **maxMessageRaiseSize** and **StartupScheduleConfig** > **maxMessageRaiseSize**.

     The default value is `50` for each. A negative value causes all messages selected from the database to be submitted for recovery. A value of `0` causes no messages to be selected from the database (effectively disabling recovery). To recover more than 50 messages, set the property value to that value. Use this property to limit the impact of recovery on the server.

- Set the maximum number of automatic recovery attempts on a given message.

  You can also configure the number of automatic recovery attempts to submit in the same recoverable instance. The value you provide specifies the maximum number of times that invoke and callback messages are recovered. If the value is `0` (the default value), it recovers all messages. Once the number of recovery attempts on a message exceeds the specified value, a message is marked as nonrecoverable.

  To configure automatic recovery attempts for invoke and callback messages in Oracle Enterprise Manager Fusion Middleware Control:

  1. In the navigation pane, right-click **soa-infra (*SOA_cluster_name*)**.
  2. Select **SOA Administration** > **BPEL Properties** > **More BPEL Configuration Properties**.
  3. Select **MaxRecoverAttempt**, and enter a value in the **Value** field.
  4. Click **Apply**.

     It may not be desirable in all cases to use automatic recovery. If services are not idempotent, then corruption can occur. Moreover, the automatic recovery restores the composite to the last save point that can be immediately after an asynchronous invoke, wait, and so on. Therefore, it is important to understand the process behavior and what it does next before performing mass recoveries. Attempt mass automatic recovery only after the root cause of the composite failures has been fixed (for example, a service that was unavailable is now available, a database running out of space was fixed, and so on). Automatic recovery can also trigger an unexpected load during failure scenarios. This causes more threads to block on a remote server that can induce hangs in the SOA server in a cascading fashion.

     For more information, see Section 12.2, "Configuring Automatic Recovery for Oracle BPEL Process Manager."

B.3.3 Some Composites Are Retried Multiple Times on Failure

When a BPEL process flow errors out, it is retried with all its invocations. This is undesirable in some cases.

The property **GlobalTxMaxRetry** (default value is `3`) specifies how many retries are performed if an error is identified as a retriable one. For example, after several web service invocations, if dehydration fails due to a data source error, then this is identified as a retriable error and all activities from the prior dehydration state are retried. If the activities being retried are not idempotent (that is, their state can change with each retry and is not guaranteed to give the same behavior), then multiple retries can be problematic.

To rectify this situation, customize the composite by specifically marking the nonidempotent activities with `idempotentset` set to `false` in the partner link settings section of the `composite.xml` file to prevent retries.

```
<property name="bpel.partnerLink.partner_link_name.idempotent">false</property>
```

You cannot set the `idempotentset` property in Oracle Enterprise Manager Fusion Middleware Control.

You can also set **GlobalTxMaxRetry** to `0` in the Systems MBean Browser.

To set **GlobalTxMaxRetry**:

1. Right-click **soa-infra (*SOA_cluster_name*)**.
2. Select **SOA Administration** > **Common Properties**.
3. Click **More SOA Infra Advanced Configuration Properties**.
4. Click **GlobalTxMaxRetry**.
5. In the **Value** field, enter an appropriate value.
6. Click **Apply**.

For information about the `idempotent` property, see the "BPEL Properties Set Inside a Composite" section of *Oracle Fusion Middleware Performance and Tuning Guide*.

B.3.4 Application Transaction Does Not Complete and the Underlying Composite Is Stuck in a Running State

An application transaction is not completing. For example, a purchase order status may remain processing. Checking the composite instance shows that the composite is stuck in the running state. In this case, the component is probably not running. Instead, it has likely faulted and may need recovery.

Faults may occur for various reasons:

- A BPEL activity faulted with an error (for example, a business error, security authorization error, or some other error).
- A BPEL activity invoked an external web service that was unavailable.
- A BPEL activity has already been terminated by the administrator using Oracle Enterprise Manager Fusion Middleware Control.
- A BPEL activity invoked an asynchronous ADF service and the message is stuck in the AQ/JMS queue.
- A BPEL activity invoked an asynchronous ADF service, but because Oracle SOA Suite was unavailable, the callback message did not arrive.

- A BPEL activity invoked a synchronous ADF service, which is taking a long time (or is hanging).
- A network error occurred.

To diagnose this issue:

1. Log in to Oracle Enterprise Manager Fusion Middleware Control.
2. In the navigator pane, go to ***domain_name*** > **SOA**.
3. Click **soa-infra (*SOA_cluster_name*)**.
4. Click the **Instances** tab.
5. Search for the composite instance, and click the instance ID.

   The Flow Trace page appears.

   If the instance is not visible (and the **Audit Level** is not set to **Off** in the SOA Infrastructure Common Properties page), this implies that the message is stuck outside of Oracle SOA Suite.

   If the message has reached Oracle Mediator, but not instantiated the BPEL flow, the BPEL instance may have been rolled back from the start due to an error.

   If the BPEL flow exists, the **Faults** section of the Flow Trace page typically shows the faulted service that can trace the root cause.
6. In the **Trace** section, click the BPEL process.
7. Expand the BPEL audit trail to see the exact point at which the service faulted.

   This information is also available in the **Faults** tab of the BPEL flow trace. It also indicates whether the fault can be recovered.
8. Click the **View Raw XML** link.

   The same information is also available through this link, where you can see the error. For example:

```
...
<message>Faulted while invoking operation "modifyUserRoles" on provider
 "UserService". </message><details>
...
<tns:message>JBO-27023: Failed to validate all rows in a
 transaction.</tns:message>
<tns:severity>SEVERITY_ERROR</tns:severity>
...
<message>The transaction was rolled back. The work performed for bpel instance
 "451042" was rolled back to the previous dehydration point, but the audit
 trail has been saved.
You can recover the instance from the recovery console by resubmitting the
 callback message or activity for execution.</message
```

   Since the instance was rolled back to its previous dehydration point, the status remains as **Running**.
9. In the **Audit Trail** and **Faults** tabs, make a note of the following:
   - Composite name (for example, **POComposite**)
   - Component (for example, **UpdateGuid** BPEL process)
   - BPEL instance ID (for example, **bpel:451042**)

This is all used in the recovery of the instance, if it is recoverable. The Audit Trail window may mark the error as a nonrecoverable business fault, but the recoverability of the message can be found in the **Recovery** tab of the BPEL process service engine.

10. To attempt to recover the instance, right-click **soa-infra (*SOA_cluster_name*)**, and select **Service Engines** > **BPEL**.
11. Click the **Recovery** tab.
12. From the **Type** list, select **Activity**.
13. Specify the composite and component names captured in step 9, and click **Search**.
14. Find the specific BPEL instance ID. You can recover faults marked as **Recoverable**.
15. Check the other recovery options in the **Type** list (for example, **Invoke** and **Callback**), if they exist.

    Note: You can also search for recoverable messages from the **Faults and Rejected Messages** tab of the SOA Infrastructure by clicking the message and selecting the appropriate action from the **Recovery Actions** list.

16. If the instance is not marked as recoverable, then reinvoking the service is not allowed (most probably because it is not idempotent). In some cases, you may need to provide diagnostic information to Oracle Support Services to resolve issues with nonrecoverable, nonidempotent transactions.
17. If the BPEL activity has invoked an asynchronous ADF service and the message is stuck in the AQ JMS queue, you can view the `server.log` and `server-diagnostic.log` files to see the logging of the message metadata logged by the JRF web services infrastructure.

    In addition, the ADF diagnostic logs are also available to debug, if needed.

    a. Use the ECID field to correlate and track ADF service logging corresponding with the SOA composite application that invoked it.
    b. When viewing the log in Oracle Enterprise Manager Fusion Middleware Control, click the **Broaden Target Scope** dropdown list and select the ***farm_name/domain_name*** (Oracle WebLogic Domain) to view messages across the domain.
    c. In the **Selected Targets** section of the Log Messages page for the Oracle WebLogic Server domain, ensure that the search includes the **ECID** field with the value noted in step a.
    d. Search and view log records for the ECID and note any issues.

       For a specific ECID, you find several root instances (top level clients). You must drill down to the appropriate instance to find a specific fault.
    e. Observe if the components completed successfully or completed with an error. See the "Viewing and Searching Log Files" section in the *Oracle Fusion Middleware Administrator's Guide*.

See the following chapters for the various ways to recover transactions and messages:

- Chapter 8, " Managing SOA Composite Application Instances" to recover from the SOA composite application page in Oracle Enterprise Manager Fusion Middleware Control.

- Chapter 14, "Managing BPEL Process Service Components and Engines" to recover from the BPEL process service component and BPEL process service engine message recovery pages.

B.4 Human Workflow Troubleshooting

This section describes how to troubleshoot human workflow issues.

B.4.1 Task Assignment/Routing/Escalation Issues

Table B–2 describes symptoms, possible causes, and possible solutions for task assignment/routing/escalation issues.

Table B–2 Troubleshooting Task Assignment/Routing/Escalation Issues

Symptom	Possible Cause	Possible Solution
The task completes without any assignment occurring.	The most common problem is that task assignees are specified using XPath expressions, and the expression does not evaluate to any nodes. Other problems can include incorrect skip conditions for participants.	1. Correct any issues with the XPath expressions. 2. Ensure that you get some results for the XPath expression for the given data. 3. Check the skip conditions specified with the **Specify skip rule** checkbox for task participants in the Human Task Editor.
The business rules do not return any list builders.	When participants of a task are specified using business rules, it is expected that business rules return at least one list builder. If business rules determine that no participants are needed, the function `IgnoreParticipant(...)` must be used. If modeled correctly and you still see this error, it is likely that none of the rules fired. See the symptom entitled "'A human workflow task chooses the incorrect user if many rules are defined or it errors with the following message:" in this table.	1. Use the `ignoreParticipant(..)` function to model your rules. 2. Ensure the rules are modeled correctly so that at least one rule is fired.
The business rules return list builders of different types.	When participants in a task are specified using business rules, it is expected that business rules return list builders of the same type.	Correct your rules.

Table B–2 (Cont.) Troubleshooting Task Assignment/Routing/Escalation Issues

Symptom	Possible Cause	Possible Solution
A human workflow task chooses the incorrect user if many rules are defined or it errors with the following message: `Ruleset returned lists with different list builder`	At runtime, when a human workflow task tries to fetch the list of users, it may error out with the following error: `Ruleset returned lists with different list builder` This error is displayed in the **Task Detail comments** field. Alternately, the task may select a user or approver, which may not appear to be the correct or expected one. This is primarily caused by having overlapping rules. When the participants of a task are specified using business rules, it is expected that business rules return list builders of the same type. Moreover, only one rule from a ruleset must be applicable for a transaction. In case many rules are true, the actions associated with the applicable rule with the highest priority get executed. In case multiple applicable rules have the same priority, then the first rule in the list is picked and its actions executed.	Avoid writing overlapping rules. Constraints from different list builders are different and cannot be mixed. If multiple rules get triggered with a different list builder, this error occurs. In addition, only one set of constraints is honored. Check that all rules in the ruleset have priorities defined so that multiple rules with the same priority are not applicable for the same transaction. For more details., see *Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management*.
Parallel assignees have to approve or reject the task even though the parallel completion criteria is met.	In the Add Participant Type dialog for a parallel participant, there is a selection that enables you to configure the human task during runtime to wait for all parallel participants to complete or to complete when criteria is met.	Make the correct selection for completion in the Add Participant Type dialog.
The task is assigned to the group/role when the expectation is that it goes to every user in the group/role individually.	When a group or a role is used as a task assignee, the task is assigned to the group or role directly. Task runtime does not assign it separately. One of the users in the group/role has to claim the task and work on it. When used with a parallel or serial participant, often times it is expected that this resolution to users is automatic, which it is not.	To assign separately to the members of the group or role, use the XPath functions `ids:getUsersInGroup` and ids:`getUsersInAppRole`.
A task errors out when invoking the decision service for evaluation of routing rules or rule-based participants.	Payload validation is enabled on the SOA infrastructure instance.	Deselect the **Payload Validation** checkbox for the instance. For more information, see Section 3.1, "Configuring SOA Infrastructure Properties."

B.4.2 Task Action Issues

Table B–3 describes symptoms, possible causes, and possible solutions for task action issues.

Table B–3 Troubleshooting Task Action Issues

Symptom	Possible Cause	Possible Solution
A user is not allowed to perform an action on a task.	The most common problem is that the user does not have permissions to perform that action on that task at that point in time.	Find out if the user can be an assignee, owner, or creator of the task, or if they are an administrator. If the user should have been allowed to perform the action, check the server log file for a detailed log message, which includes information such as the task state, task assignees, user who acquired it, permitted actions, roles played by this user for the given task, and so on.
A task is locked and cannot be updated in Oracle BPM Worklist. An `ORA-30279` error is displayed (with an underlying ORA SQL exception of `ORA-20001` also displayed).	Two users are trying to update the same task concurrently.	Wait for the other user to finish.

B.4.3 Notification Issues

Table B–4 describes symptoms, possible causes, and possible solutions for notification issues.

Table B–4 Troubleshooting Notifications Issues

Symptom	Possible Cause	Possible Solution
The task email notification is not being sent out.	**Notification Mode** is set to **NONE** on the Workflow Notification Properties page in Oracle Enterprise Manager Fusion Middleware Control.	Change this setting to **Email** or **All**. For information, see Section 20.1, "Configuring Human Workflow Notification Properties."
The email notification is not being sent out.	Incorrect outgoing server settings are used in the email driver configuration.	Check the SMTP port/SMTP host/user name/password/email values. **Tip**: Validate the values by using them in any email client for connecting to the SMTP server. Perform the following steps to verify the settings in Oracle Enterprise Manager Fusion Middleware Control: 1. In the navigator, right-click **User Messaging Service**. 2. Select **usermessagingdriver-email** > **Email Driver Properties**. 3. Check the settings.
The notifications are sent, but are not actionable.	The **Actionable Address** field is not configured.	In Oracle Enterprise Manager Fusion Middleware Control, configure the **Actionable Address** field with a valid email address. For information, see Section 20.1, "Configuring Human Workflow Notification Properties." Ensure that the same email address is used when configuring the incoming server setting in the Oracle User Messaging Server email driver. For information, see Section 20.1, "Configuring Human Workflow Notification Properties."

Table B–4 (Cont.) Troubleshooting Notifications Issues

Symptom	Possible Cause	Possible Solution
Notifications are sent, but are not actionable.	The human workflow task is not set to send actionable notifications.	In the Human Task Editor (you can double click the `.task` file in Oracle JDeveloper to start the editor), expand the **Notification** section, click the **Advanced** tab, and select the **Make notification actionable** checkbox.
Actionable notifications are sent, but no action is taken after responding.	The **Actionable Address** field is incorrect.	Check the IMAP/POP3 server/port values. Ensure the **Actionable Address** field is used in the email driver configuration. **Tip**: Validate the values by using them in any email client for connecting to the IMAP/POP3 server.
Actionable notifications are sent, but no action is taken after responding.	The nondefault email client is configured for receiving notifications.	When the user clicks the approval link, the default mail client page opens, which may send emails to a different email server. Configure the default email client to receive actionable notifications. Enter the correct value in the **Actionable Email Account** field of the Workflow Task Service Properties page as the incoming, actionable email account to use. The default account name is **Default**. For information, see Section 20.2, "Configuring Human Workflow Task Service Properties."

Table B–4 (Cont.) Troubleshooting Notifications Issues

Symptom	Possible Cause	Possible Solution
Actionable notifications are sent but no action is taken after responding.	An email client is configured with the same account used in the email driver.	The mail may be downloaded and marked as *read* or deleted by the email client before the human workflow notification service can download and process the mail. Remove that account from the email client.
The Oracle BPM Worklist link appears in email notifications.	This is the default behavior. By default, email notifications point to Oracle BPM Worklist.	Perform the following steps: **1.** In the **Notification** section of the Human Task Editor, click the **Advanced** tab. **2.** Deselect the **Show worklist URL in notifications** checkbox.
Performance is slow for group notifications.	The group notification performance depends on the number of members in the group (size of group).	**1.** Provide an email ID for the group in LDAP. In this case, human workflow sends one email to the group email ID, instead of individual emails to each group member. **2.** In the **Notification** section of the Human Task Editor, click the **Advanced** tab. **3.** If you want to send one email to all group members, select Send one email containing all user addresses from the **Group notification configuration** list. This action enables all members to see the *to* list and common content is sent to all members (without considering locale, and so on). **4.** If you want to send individual emails but reuse content between members, select **Send individual emails** (the default selection) from the **Group notification configuration** list and unselect **Use separate task forms based on locale**. This action enables group members to receive individual mails in their locale. Task forms generated for creating notification content are reused between members in the same locale.

B.4.4 Task View Issues

Table B–5 describes symptoms, possible causes, and possible solutions for task view issues.

Table B–5 Troubleshooting Task View Issues

Symptom	Possible Cause	Possible Solution
Showing custom (mapped attribute) columns in a view.	Attribute mappings are created for specific task types. The view must be associated with one or more task types to use mapped attributes.	In Oracle BPM Worklist (view/create/edit UI), specify a task type for the view in the **Definition** tab. The attribute labels used in the mappings for that task type are now available as columns that can be used in the view in the **Display** tab. It is possible to associate a view with multiple task types. Multiple task types can be selected from the **Task Type** browser. If multiple task types are selected, then the attribute labels for all those task types are available for use in the view.
View grantees can view and edit tasks belonging to the view owner.	The view is shared as data. This type of sharing allows grantees to use the view as if they are the view owner, and can see and act on the view owner's task.	In the **Definition** tab of Oracle BPM Worklist (view/edit UI), ensure that **Share View** is set to **Definition only**, which enables grantees to use the view against their own tasks. Setting **Share View** to **Data** enables grantees to use the view against the view owner's data.
Creating a new standard view.	Only users with administration privileges can create standard views.	1. Ensure the logged-in user has administration privileges. 2. Define the view as usual, using Oracle BPM Worklist (create/edit UI). 3. Check the **add to standard views** checkbox. The view is created as a standard view.
Internationalizing a standard view name.	The value specified in the name field for standard views can be used as a resource key to look up a display name from the `WorkflowLabels` resource bundle.	Add a new resource key to the `WorkflowLabels` resource bundle. The key is the name you used for the view, prefixed by `STD_VIEW.` (note the required trailing period). For more information about resource bundles, see workflow sample `workflow-110-workflowCustomizations`, available with the Oracle SOA Suite Samples.
Migrating views and standard views you have created on one instance to another SOA server.	You must use the test-to-production utility.	The test-to-production utility enables you to export user views and standard views as an XML file, and to import the views from the XML file into another instance. For information about this utility, see Section 22.6, "Moving Human Workflow Data from a Test to a Production Environment."

B.4.5 Task Attribute Mapping Issues

Table B–6 describes symptoms, possible causes, and possible solutions for task attribute mapping issues.

Table B–6 Troubleshooting Task Attribute Mapping Issues

Symptom	Possible Cause	Possible Solution
No payload attributes are available for mapping to a public attribute in Oracle BPM Worklist.	Oracle BPM Worklist only supports creation of mappings to simple payload attributes. Only simple attributes from the task payload are displayed for creating mappings in Oracle BPM Worklist.	▪ Add simple attributes to the payload. ▪ Create protected mappings at design time. Mappings can be made to be complex payload attributes using XPath expressions for protected attribute mappings at design time in Oracle JDeveloper. ▪ Use the `RuntimeConfigService` API to create public mappings using XPath expressions.
You cannot create mappings for the protected attribute label in Oracle BPM Worklist.	Protected mappings can only be created as part of the task definition at design time. Protected mappings cannot be created or updated at runtime using Oracle BPM Worklist, or the `RuntimeConfigService` API.	▪ Use a public attribute label for the mapping. ▪ Create the mapping to the protected attribute label in the task definition at design time.
You cannot see any attribute labels for which to create mappings in Oracle JDeveloper.	Design-time mappings can only be created for protected attribute labels. Ensure that protected attribute labels have been created in the SOA instance to which you are connected.	1. Log in to Oracle BPM Worklist as an administrator. 2. Go to **Administration** > **Protected Flexfields**. 3. Check that protected attribute labels exist. 4. Create any protected attribute labels as required.
Internationalizing the name of an attribute label.	The attribute label name can be used as a resource key to look up a display name from the `WorkflowLabels` resource bundle.	Add a new resource key to the `WorkflowLabels` resource bundle. The key is the name you used for the label, prefixed by `FLEX_LABEL`. (note the trailing period). For more information on the resource bundle, see the workflow sample `workflow-110-workflowCustomizations`, available with the Oracle SOA Suite Samples.
Migrating attribute labels and mappings from one server to another.	Use the test-to-production utility.	The test-to-production utility enables you to export public attribute labels, public attribute mappings, and protected attribute labels as an XML file, and to import the labels and mappings from the XML file into another instance. For more information, see Section 22.6, "Moving Human Workflow Data from a Test to a Production Environment."

B.4.6 Task Report Issues

Table B–7 describes symptoms, possible causes, and possible solutions for task report issues.

Table B–7 Troubleshooting Task Report Issues

Symptom	Possible Cause	Possible Solution
You receive the following error: `Null Pointer Exception when running Task Productivity Report`	This is caused by an issue with handling of dates when the worklist client locale and server default locale are different.	The workaround is to change the locale for the worklist client to be the same as the server, or to run a report without specifying dates.

B.4.7 Task History Issues

Table B–8 describes symptoms, possible causes, and possible solutions for task history issues.

Table B–8 Troubleshooting Task History Issues

Symptom	Possible Cause	Possible Solution
The **Add Participant** button is disabled.	A current or past participant is selected in the history table,	This is designed behavior. Adding adhoc participants is not allowed with the current or past participant. The current participant means the task is with that participant at that point in time.
All the added adhoc participants disappeared after a page refresh.	You may not have saved your modifications to the history table.	Ensure that you save your changes. Otherwise, all changes disappear. If you think you have saved your changes and the changes still disappear, file a bug.
Do not see future approvers in the history table.	The **Future Approvers** checkbox may not be selected.	Select the **Future Approvers** checkbox in Oracle BPM Worklist (configuration in the task sequence table).
You see the message in the history table about the correlation ID not being passed or any exception related to the correlation ID.	If the task is uninitiated, the correlation ID may not have been passed.	Ensure that you pass the correlation ID to the uninitiated task.
The edit toolbar is disabled or is not displayed.	The user may not have privileges to edit the participants.	1. In the **Assignment** tab of the Human Task Editor, click the **Task will go from starting to final participant** icon in the upper right corner. The Configure Assignment dialog is displayed. 2. Select the **Allow participants to edit new participants** and **Allow initiator to add participants** checkboxes.
You receive the following error: `<Warning> <oracle.adf.controller.internal.metadata.MetadataService> <BEA-000000><ADFc: /META-INF/adfc-config.xml: > <Warning> <oracle.adf.controller.internal.metadata.MetadataService><ADFC-52024> <ADFc: Duplicate managed bean definition for 'aleCompBindings' detected.>`	Shared library `oracle.soa.worklist.webapp` is referenced in `weblogic.xml` and the JAR files `adflibWorklistComponents.jar` and `adflibTasklistTaskflow.jar` are packaged in the web application.	These JARs ideally should not be packaged inside the web application. They should only be referenced as a shared library. Do not package these JARs in the web application.

Table B–8 (Cont.) Troubleshooting Task History Issues

Symptom	Possible Cause	Possible Solution

You receive the following error:
`<Error> <Deployer> <BEA-149265> <Failure occurred in the execution of deployment request with ID '1297964056778' for task '3'. Error is:`
 	
`'weblogic.management.DeploymentException: [J2EE:160149]Error while processing library references. Unresolved application library references, defined in weblogic-application.xml: [Extension-Name: oracle.soa.workflow.wc, exact-match: false].' weblogic.management.DeploymentException: [J2EE:160149]Error while processing library references. Unresolved application library references, defined in weblogic-application.xml: [Extension-Name: oracle.soa.workflow.wc, exact-match: false].`
	Shared library `oracle.soa.workflow.wc` is referenced in `weblogic.xml`, but not available on the server.	Ensure that this shared library is deployed on the server to which you are deploying your application. It may happen that the shared library is deployed, but *not* targeted, for that server.

You receive the following error: `java.lang.IllegalStateException: Attempt to validate an already invalid RegionSite:`	This is a generic exception that sometimes is displayed in the server logs (for example, `AdminServer.log`).	See the real exception in the diagnostic logs (for example, `AdminServer-diagnostic.log`) and provide that exception with the bug you file.

Table B–8 (Cont.) Troubleshooting Task History Issues

Symptom	Possible Cause	Possible Solution
You receive the following error: `[AdminServer] [NOTIFICATION] [J2EE JSP-00008] [oracle.j2ee.jsp] [tid: [ACTIVE].ExecuteThread: '15' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId: weblogic] [ecid: 17011f2a001d6b0e:7e22d6ce:12e3444eb1b:-8000-0000000000002f0a,0] [APP: FederatedApp_application1] unable to dispatch JSP page: The following exception occurred:.[[java.lang.RuntimeException: Cannot find FacesContext at javax.faces.webapp.UIComponentClassicTagBase.getFacesContext(UIComponentClassicTagBase.java:2122)`	This is a common mistake and is not related to any components you are using. You forget to put `faces` in the URL. For example: `http://server:port/FederatedApp/test.jspx`	Put `faces` in the URL as follows: `http://server:port/FederatedApp/faces/test.jspx\\`

B.4.8 Task Form/Action Issues

Table B–9 describes symptoms, possible causes, and possible solutions for task form/action issues.

Table B–9 Troubleshooting Task Form/ Action Issues

Symptom	Possible Cause	Possible Solution
The task form application does not have an empty JSPX page.	N/A	The task forms are now invoked using an ADF task flow and control is returned to the module that initiated the task form task flow when the task flow completes. Therefore, no empty JSPX is needed.
The task form does not load in Microsoft Internet Explorer.	Microsoft Internet Explorer has a URL length limit.	Your task form URL length is too long.

Table B–9 (Cont.) Troubleshooting Task Form/ Action Issues

Symptom	Possible Cause	Possible Solution

Deployment fails with a `class not found` exception.	The shared library entry is missing from `weblogic.xml`.	If you see the following error:
		`Caused By: java.lang.ClassNotFoundException: oracle.bpel.services.datacontrol.types.Number`
		during deployment of a task form, then it is likely due to the missing shared library in `weblogic.xml`. Add the following element in `weblogic.xml`:
		`<library-ref> <library-name>oracle.soa.worklist.webapp</library-name> <specification-version>11.1.1</specification-version> </library-ref>`

Deployment/access of task form fails when the hostname is used.	The DNS entry is missing.	If you are using a server with DHCP, the DNS entry may be missing for the host. Therefore, deployment/access using the IP address may succeed, but deployment/access using a hostname may fail. Update your client computer by manually adding the host/IP address: ■ On Windows operating systems, this is typically is in `%windir%\drivers\etc\lmhosts`. ■ On Linux/UNIX, this is typically in `/etc/hosts`.

Table B–9 (Cont.) Troubleshooting Task Form/ Action Issues

Symptom	Possible Cause	Possible Solution
Task form URL protocol (HTTP or HTTPS).	You are unable to access the task form through HTTPS or HTTP.	▪ DefaultToDoTaskForm: The worklist accesses the default to-do task form using the port/protocol returned by the `getServerInfo` API from the runtime config service. This API uses the frontend host setup for the cluster or managed server. It gives preference to the HTTPS protocol if it is enabled. It is dynamic and can be changed using Oracle Enterprise Manager Fusion Middleware Control for default to-do tasks as the other custom task form. ▪ CustomTaskForm: The custom task form gets the HTTP and HTTPS information from the setup. It uses the JRF API to get the information. This gets the frontend host information from the cluster and then the managed server if the cluster frontend host is not set up. In case the managed server frontend host is not set up, it uses a regular port. If the user enables the HTTPS port during deployment, the task form stores this information in the database. After this occurs, if the user disables the HTTPS port using Oracle WebLogic Server Administration Console, then the user must remove the HTTPS port for the task form using Oracle Enterprise Manager Fusion Middleware Control.

B.4.9 Task Comments/Attachment Issues

Table B–10 describes symptoms, possible causes, and possible solutions for task comments/attachment issues.

Table B–10 Troubleshooting Task Comments/Attachment Issues

Symptom	Possible Cause	Possible Solution

The file is not getting uploaded.	The file is too big	By default, ADF has a size limit of 2000 KB for each request. Add the following parameters in `web.xml` to adjust file size and temporary storage for uploaded files:
		`<context-param>`
		`  <!-- Maximum memory per request (in bytes) -->`
		`<param-name>oracle.adf.view.faces.UPLOAD_MAX_MEMORY</param-name>`
		`  <!-- Use 500K -->`
		`  <param-value>512000</param-value>`
		`</context-param>`
		`<context-param>`
		`  <!-- Maximum disk space per request (in bytes) -->`
		`<param-name>oracle.adf.view.faces.UPLOAD_MAX_DISK_SPACE</param-name>`
		`  <!-- Use 5,000K -->`
		`  <param-value>5120000</param-value>`
		`</context-param>`
		`<context-param>`
		`  <!-- directory to store temporary files -->`
		`<param-name>oracle.adf.view.faces.UPLOAD_TEMP_DIR</param-name>`
		`  <!-- Use an ADFUploads subdirectory of /tmp -->`
		`<param-value>/tmp/ADFUploads/</param-value>`
		`</context-param>`

The file uploaded in the task details application is not visible in the same task flow.	After uploading a file, the attachment link generated in task form is invalid. Clicking this link returns an empty stream.	When you upload a file, you see the attachment link in the table. However, this link does not work. You must reload the task details to view the file.
Adding file attachments creates a new task version, but adding a URL attachment does not create a new version.	Inconsistent behavior of URL attachment and file attachment	When a file is uploaded, the task is saved because the file is uploaded to persistency storage. This creates a new task version. The URL attachments only update the local task object in the user interface application. Therefore, no task version is created.

B.4.10 Design Time at Runtime Issues

Table B–11 describes symptoms, possible causes, and possible solutions for design time at runtime issues. Two design time at runtime tools are available for use:

- Oracle SOA Composer
- **Task Configuration** tab of Oracle BPM Worklist

Table B–11 Troubleshooting Design Time at Runtime UI Issues

Symptom	Possible Cause	Possible Solution
Modifications made to a task in a design time at runtime tool do not appear for the task.	The task was instantiated before you actually edited it using a design time at runtime tool.	Design time at runtime updates go into effect only for instances created after the changes, and not for those that were created before the change. Therefore, if you edit a task using a design time at runtime tool, and then instantiate a new task, the new instance of the task has the changes you made.
Modifications made to a task in a design time at runtime tool do not appear for the task.	The changes made were probably not committed to the MDS repository.	The **Save** button just saves the changes made in a design time at runtime tool to the sandbox. To see these changes in action, click **Commit** to send them to the MDS repository.

B.4.11 Human Workflow API (Including SOAP/EJB) Usage Issues

Table B–12 describes symptoms, possible causes, and possible solutions for human workflow API (including SOAP/EJB) usage issues.

Table B–12 Troubleshooting Human Workflow API Usage Issues

Symptom	Possible Cause	Possible Solution
Location of the JavaDoc for human workflow APIs.	N/A	See *Oracle Fusion Middleware Workflow Services Java API Reference for Oracle BPEL Process Manager*, which is available in the documentation library.
Understanding the API usage.	N/A	1. Refer to the API documentation. 2. See sample `workflow-118-JavaSamples`, available with the Oracle SOA Suite Samples.
Using `.net` to access the APIs.	N/A	It is possible to write a `.net` client that accesses the SOAP web service APIs. See sample `workflow-119-DotNetSample`, available with the Oracle SOA Suite Samples.
You receive `Class not found` errors when attempting to use the Java API clients.	Not all required JAR files are in the client class path.	See the `workflow-118-JavaSamples` definition of `client.classpath` in the `build.xml ant` file, available with the Oracle SOA Suite Samples.The sample includes all required JAR files.
Creating a routing slip for simple patterns to use with a simple approval task or to dynamically route a task during task initiation.	N/A	See `oracle.bpel.services.workflow.task.impl.SimpleApprovalTaskUtil`.

B.4.12 Oracle JDeveloper Data Control / Form Generation Issues

Table B–13 describes symptoms, possible causes, and possible solutions for Oracle JDeveloper data control/form generation issues.

Table B–13 Troubleshooting Oracle JDeveloper Data Control / Form Generation Issues

Symptom	Possible Cause	Possible Solution

`Empty1.jspx` is not generated when creating the task form.	Created an initiator task based on an XSD element and tried to autogenerate the task form.	This file is not required except for the BPM initiator task. If you encounter it, create an `Empty1.jspx` file in the same directory as referenced by `adfc-config.xml` and put the following content in it:
		`<?xml version='1.0' encoding='UTF-8'?>`
		`<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" version="2.1" xmlns:f="http://java.sun.com/jsf/core" xmlns:h="http://java.sun.com/jsf/html" xmlns:af="http://xmlns.oracle.com/adf/faces/rich">`
		`<jsp:directive.page contentType="text/html;charset=UTF-8"/>`
		`<f:view>`
		`<af:document id="d1">`
		`<af:form id="f1"></af:form>`
		`</af:document>`
		`</f:view>`
		`</jsp:root>`

B.4.13 Human Workflow Service/ System MBean Browser Issues

Table B–14 describes symptoms, possible causes, and possible solutions for human workflow service/System MBean Browser issues.

Table B–14 Troubleshooting Human Workflow Service/ System MBean Browser Issues

Symptom	Possible Cause	Possible Solution
Setting commonly used human workflow configuration parameters.	N/A	Use the Workflow Task Service and Workflow Notification pages of Oracle Enterprise Manager Fusion Middleware Control: 1. In the navigator, right-click **soa-infra**. 2. Select **SOA Administration** > **Workflow Config** > **Mailer** tab to access notification properties. 3. Select **SOA Administration** > **Workflow Config** > **Task** tab to access task service properties.
Setting human workflow configuration parameters not available in the Oracle Enterprise Manager Fusion Middleware Control properties pages.	N/A	Use the System MBean Browser in Oracle Enterprise Manager Fusion Middleware Control: 1. In the navigator, right-click **soa-infra**. 2. Select **SOA Infrastructure** >**Administration** > **System MBean Browser**. 3. Select **Application Defined MBeans** > **oracle.as.soainfra.config** > **server** > **WorkflowConfig** > **human-workflow**. 4. Set simple parameters by editing the appropriate field in the **Attributes** tab, and clicking **Apply**. 5. Make more complex parameter changes (for example, adding a new locale) by switching to the **Operations** tab, selecting the appropriate operation, entering required fields, and clicking **Invoke**.

Table B–14 (Cont.) Troubleshooting Human Workflow Service/ System MBean Browser Issues

Symptom	Possible Cause	Possible Solution
The System MBean Browser does not reflect my changes after editing the human workflow configuration MBeans.	The System MBean Browser is showing a previously cached version of beans.	Click the **refresh cached tree data** button in the System MBean Browser.
Human workflow services are not locating resource bundles or classes located at the workflow customizations class path URL.	The protocol is not specified in the URL, or the URL is missing a trailing forward slash (/).	Ensure that the configured URL is formatted correctly, and specifies a protocol. If the class path points to a directory (rather than a JAR file), it is important that the URL has a trailing forward slash character. For example: `file:///home/wstallar/wfcustomizations/`
Manually setting the URL used for displaying task details for a particular task component.	N/A	Use the Administration page in Oracle Enterprise Manager Fusion Middleware Control for the human task service component. See Section 22.3, "Managing the URI of the Human Task Service Component Task Details Application" for instructions. You can edit or delete existing task display URL entries, and add new entries. For task display URLs used from Oracle BPM Worklist, the application name must be set to **worklist**.

B.4.14 AMX Extension Issues

Table B–15 describes symptoms, possible causes, and possible solutions for AMX extension issues.

Table B–15 Troubleshooting AMX Extension Issues

Symptom	Possible Cause	Possible Solution
The dynamic approval group class is not found.	The class file is not accessible in the Oracle SOA Suite class path	To make the dynamic approval group class accessible, the class file must be placed in the following directory: `$FMW_HOME/SOA_HOME/soa/modules/oracle.soa.ext_11.1.1/classes` This directory is part of the SOA class path. The Oracle WebLogic Server must be restarted.
During design time at runtime, while defining a rule based on the Approval Group list builder, a message keeps appearing indicating that the group does not exist.	The Approval Group name is not enclosed in quotes (" ")	Enclose the name in quotes (for example, "Sample Approval Group Name").
In a ruleset, many rules defined are applicable for a transaction. It appears that the correct constraints are not getting applied; therefore, the generated approver list is not correct.	Only one rule from a ruleset must be applicable for a transaction. In case many rules are true, the actions associated with the applicable rule with the highest priority are executed. In case multiple applicable rules have the same priority, the first rule in the list is picked and its actions are executed.	Check that all rules in the ruleset have priorities defined so that multiple rules with the same priority are not applicable for the same transaction.

B.4.15 Oracle BPM Worklist/Task Region Issues

Table B–16 describes symptoms, possible causes, and possible solutions for Oracle BPM Worklist/task region issues.

Table B–16 Troubleshooting Oracle BPM Worklist/Task Region Issues

Symptom	Possible Cause	Possible Solution
You receive the following exception message in the logs: `<Warning> <oracle.adf.controller.internal.metadata.MetadataService> <BEA-000000><ADFc: /META-INF/adfc-config.xml: > <Warning> <oracle.adf.controller.internal.metadata.MetadataService><ADFC-52024> <ADFc: Duplicate managed bean definition for 'aleCompBindings' detected.>`	The shared library `oracle.soa.worklist.webapp` is referenced in the `weblogic.xml` file and also the JAR files `adflibWorklistComponents.jar` and `adflibTasklistTaskflow.jar` are packaged in the web application.	These JARs ideally should not be packaged inside the web application. They should only be referenced as a shared library. Do not package these JARs in the web application.
You receive the following exception message in the logs: `Duplicate default server in client configuration. Configuration needs to have only one default server in client configuration. Specify one default server in client configuration.`	Two default servers are specified in the client configuration file or in the JAXB object passed to the task flow.	Mark only one server as the default in the client configuration file or in the JAXB object passed.
You receive the following exception message in the logs: `The default server is not specified`	The default server is not specified in the client configuration file or in the JAXB object passed to the task flow.	Ensure that the default server is marked in the client configuration file or the JAXB object.
You receive the following exception message in the logs: `Invalid display column. The display column COLUMN NAME is not a valid Task column. Specify a valid column name.`	The column name passed to the task flow parameter `displayColumnsList` is not correct.	Ensure that you pass the correct column name to the task flow parameter.
You receive the following exception message in the logs: `java.lang.IllegalStateException: Attempt to validate an already invalid RegionSite:`	This is a generic exception that sometimes appears in server logs (for example, `AdminServer.log`).	See the real exception in the diagnostic logs (for example, `AdminServer-diagnostic.log`) and provide that exception with the bug you can file.
You receive the following exception message in the logs: `Caused by: oracle.adf.controller.ControllerException: ADFC-02001: The ADF Controller cannot find '/WEB-INF/taskList-task-flow-definition.xml'`	The Oracle BPM Worklist JARs are not provided in the class path, either by referring to the shared library `oracle.soa.worklist.webapp` or by packaging those in the web application.	Ensure either the JARs are referred through the shared library or packaged inside the application.
Filters for the task list are removed when the task list is refreshed.	Because an inbox is not a persisted view, filters set on it are removed when rendering the page again or refreshing the task list.	Instead of setting filters on the task list, create a user view with the required set of filters and pass the `viewId` of that view as the value of parameter `ViewFilter`. This makes that view the default view of the task list.

Table B–16 (Cont.) Troubleshooting Oracle BPM Worklist/Task Region Issues

Symptom	Possible Cause	Possible Solution

You have set the `taskTypesFilterList` parameter, but instead of seeing assigned tasks, you are seeing all tasks.	If you specified the `taskTypesFilterList` parameter, then you forgot to specify the `attributesFilterList` parameter.	You have to use both parameters with the `AND` operator. For example:
		`<parameter id="taskTypesFilterList" value="http://xmlns.oracle.com/HelpDeskRequestSOAApp/HelpDeskRequestComposite/HelpDeskRequestHumanTask,[ http://xmlns.oracle.com/VacationRequestApp/VacationRequest/VacationRequestTask]"/>`
		
		`<parameter id= "attributesFilterOperator" value="and"/>`
		`<parameter id= "attributesFilterList" value="state=ASSIGNED"/>`

You receive the following exception message in the logs: `[AdminServer] [NOTIFICATION] [J2EE JSP-00008] [oracle.j2ee.jsp] [tid: [ACTIVE].ExecuteThread: '15' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId: weblogic] [ecid: 17011f2a001d6b0e:7e22d6ce:12e3444eb1b:-8000-0000000000002f0a,0] [APP: FederatedApp_application1] unable to dispatch JSP page: The following exception occurred:.[[java.lang.RuntimeException: Cannot find FacesContext at javax.faces.webapp.UIComponentClassicTagBase.getFacesContext(UIComponentClassicTagBase.java:2122)`	This is a common mistake that is generic in nature and is not related to any components you are using. You forgot to put `faces` in the URL. For example: `http://server:port/FederatedApp/test.jspx`	Put `faces` in the URL as follows: `http://server:port/FederatedApp/faces/test.jspx`

Table B–16 (Cont.) Troubleshooting Oracle BPM Worklist/Task Region Issues

Symptom	Possible Cause	Possible Solution
You receive the following exception message in the logs: `[AdminServer] [TRACE] [] [] [tid: [ACTIVE].ExecuteThread: '5' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId: weblogic] [ecid: 17011f2a001d6b0e:7e22d6ce:12e3444eb1b:-8000-0000000000001d39,0] [SRC_CLASS: oracle.bpel.services.workflow.client.config.ClientConfigurationUtil] [APP: FederatedApp_application1] [SRC_METHOD: getClientConfiguration] WorkflowServiceClientContext: Cannot find client configuration file: wf_client_config.xml`	There are three possible causes for this issues: **1.** The client configuration file is not provided in the class path. **2.** The JAXB object is not passed to the task flow. **3.** If one of the above is provided, the port number for the remote/SOAP client is incorrect.	Ensure either the client configuration file `wf_client_config.xml` is provided in the class path or the JAXB object is passed to the task flow. If it is already done, ensure that the port number refers to the SOA server.
You receive the following exception message in the logs: `<Error> <Deployer> <BEA-149265> <Failure occurred in the execution of deployment request with ID '1297964056778' for task '3'. Error is: 'weblogic.management.DeploymentException: [J2EE:160149]Error while processing library references. Unresolved application library references, defined in weblogic-application.xml: [Extension-Name: oracle.soa.workflow.wc, exact-match: false].' weblogic.management.DeploymentException: [J2EE:160149]Error while processing library references. Unresolved application library references, defined in weblogic-application.xml: [Extension-Name: oracle.soa.workflow.wc, exact-match: false].`	Shared library `oracle.soa.workflow.wc` is referenced in `weblogic.xml`, but is not available on the server.	Ensure that this shared library is deployed on the server on which you are deploying your application. The shared library may be deployed, but not targeted, for that server.

Table B–16 (Cont.) Troubleshooting Oracle BPM Worklist/Task Region Issues

Symptom	Possible Cause	Possible Solution

You cannot see the mapped attributes mapped columns.
Note: Oracle BPM Worklist flex fields are now known as mapped attributes.
	The right set of parameters is not being passed to the task list task flow.	The correct set of parameters to be passed is as follows:
		`<parameter id= "displayColumnsList" value="assignees,creator, assignedDate,state, textAttribute1,textAttribute2"/>`
		`textAttribute1,textAttribute2` is the correct way to provide a value for the column name. Providing `'label name'(Name of the mapping)` associated with these values does not work.
		You must specifically pass the fully qualified value to parameter `taskTypesFilterList`. Otherwise, the column creation does not work.
		For example:
		`<parameter id="taskTypesFilterList" value="http://xmlns.oracle.com/HelpDeskRequestSOAApp/HelpDeskRequestComposite/HelpDeskRequestHumanTask"/>`

B.4.16 Test-to-Production Issues

Table B–17 through Table B–20 describe symptoms, possible causes, and possible solutions for test-to-production issues.

Table B–17 Troubleshooting Test-to-Production Issues

Symptom	Possible Cause	Possible Solution

Finding the default realm name for a SOA server	N/A	The `defaultRealmName` can be obtained from the identity configuration service. You can get this from the SOAP service test page.
		1. From a browser, go to the following URL:
		`http:HOST:PORT/integration/services/IdentityService/configuration`
		2. Select `getDefaultRealmName` from the **Operation** drop-down menu.
		3. Click the **Invoke** button.
		This retrieves the default realm name. Here is a sample answer from an invocation:
		`<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap /envelope/">`
		`<env:Header/>`
		`<env:Body>`
		`<realmName xmlns="http://xmlns.oracle.com/bpel/servi ces/IdentityService">jazn.com</realmName>`
		`</env:Body>`
		`</env:Envelope>`

Table B–18 Troubleshooting Test-to-Production Issues

Symptom	Possible Cause	Possible Solution
User authentication error (in the following example, the user is `FMW_USERID` and the identity context is `jazn.com`): Assume you encounter the following error: `[java] Error in workflow service Web service operation invocation. The error is ORA-30501:Error in authenticating user. [java] Error in authenticating and creating a workflow context for user jazn.com/FMW_USERID. [java] Verify that the user credentials and identity service configurations are correct.`	This occurs if the given user is not seeded and available in the LDAP provider. To find out if the user is seeded properly, try to log in to Oracle BPM Worklist from a browser as this user. If the user can log in to Oracle BPM Worklist, that means the user is seeded.	If Oracle Internet Directory or another other LDAP provider is used, ensure the configuration of the LDAP provider is completed correctly. Otherwise, you cannot get past this error.
Import of task payload mapped attributes (previously known as flex field) mappings: While importing task payload mapped attribute mappings into the target SOA server, you may encounter the following error in the console logs: `[java] Caused by: java.sql.SQLIntegrityConstraintViolationException: ORA-02291: integrity constraint (UAT_SOAINFRA.SYS_C0018364) violated - parent key not found* *`	The importing of task payload mapped attribute mappings into the target SOA server is a two-step process. Even before the import of task payload mapped attribute mappings into the target SOA server operation is attempted, there is a prerequisite step that must be performed. This is the import of attribute labels into the target SOA server operation.	To be successful, perform the following operations (in the correct order) with the human workflow test-to-production migration tool: **1.** Import attribute labels into the target SOA server. **2.** Import payload mappings into the target SOA server. See Section 22.6, "Moving Human Workflow Data from a Test to a Production Environment" for more details.

Table B–19 Troubleshooting Test-to-Production Issues

Symptom	Possible Cause	Possible Solution
Assume you encounter the following error during rule migration: `[java] Error encountered during migration. [java] Exception in thread "main"[java] UserConfigDataMigrationException:[java] faultString:Invalid parameters for RULE. [java] Invalid user and group: both parameters can not have null values.[java] To migrate User Rules, provide the 'user' parameter only. [java] To migrate Group Rules, provide the 'group' parameter only.[java] oracle.bpel.services.workflow.util. tools.wfUserConfigDataMigrator. UserConfigDataMigrationE xception [java] at oracle.bpel.services.workflow. util.tools.wfUserConfigDataMigrator. implhwfMigrator.parseParametersNode`	There are two properties in the `migration.properties` file that are of interest for this error. ■ `user` ■ `group` During any rule migration (whether export or import) operation, at most one of them (`user` or `group`) should have a value. That is, both `user` and `group` cannot have null or empty values.	Set values for at most one of them. To perform user rule migration, set the `user` parameter alone. To perform group rule migration, set the `group` parameter alone.
Assume you encounter the following error during rule migration: `[java] Error encountered during migration.[java] Exception in thread "main"[java] UserConfigDataMigrationExcepti on: [java] faultString: Invalid parameters for RULE. [java] Invalid user and group: both parameters can not have values. [java] To migrate User Rules, provide the 'user'parameter only. [java] To migrate Group Rules, provide the 'group' parameter only. [java] oracle.bpel.services. workflow.util.tools.wfUserConfigData Migrator.UserConfigDataMigrationE xception`	This is similar to the previous explanation. During any rule migration (whether export or import) operation, both `user` and `group` parameters cannot have a value.	Provide values for at most one of them.

Table B–20 Troubleshooting Test-to-Production Issues

Symptom	Possible Cause	Possible Solution

Testing the health of the installed server	N/A	Before performing a test-to-production migration, it is useful to test the health of the server.
		From a browser, you can test some SOAP services. The following list provides a subset of human workflow services and Oracle BPM Worklist:
		`http://host:port/integration/worklistapp/`
		`http://host:port/integration/services/TaskQueryService/TaskQueryService`
		`http://host:port/integration/services/IdentityService/configuration`
		`http://host:port/integration/services/IdentityService/identity`
		`http://host:port/integration/services/RuntimeConfigService/RuntimeConfigService`
		You can randomly test some operations in these services, and verify that the operation yields results. Similarly, you can log in as a user to Oracle BPM Worklist and see if everything is fine.

B.4.17 Identity Service Issues

Table B–21 and Table B–22 describe symptoms, possible causes, and possible solutions for identity service issues.

Table B–21 Troubleshooting Identity Service Issues

Symptoms	Possible Cause	Possible Solution

Only a subset of users in LDAP can log in to Oracle BPM Worklist.	The user base DN is not configured properly.	Mention the user base under which all the groups are seeded. This can be performed in two ways:
		Add the base DN under which all the required groups are seeded. For instance, if users are seeded under:
		`UserDN 1 : cn=users1,dc=us,dc=oracle,dc=com`
		`UserDN 2 : cn=users2,dc=us,dc=oracle,dc=com`
		`UserDN 3 : cn=users3,dc=us,dc=oracle,dc=com`
		Then mention the group base DN as follows:
		`dc=us,dc=oracle,dc=com`
		This is the common DN. If only some user DNs are required (for example, `UserDN1` and `UserDN2`), then the following property must be added to `serviceInstance` with the name `idstore.ldap` in the `$DOMAIN_HOME/config/fmwconfig/jps-config.xml` file:
		`<serviceInstance name="idstore.ldap"`
		` provider="idstore.ldap.provider">`
		` <property name="idstore.config.provider"`
		`value="oracle.security.jps.wls.internal.idstore.WlsLdapIdStoreConfigProvider"/>`
		` <property name="CONNECTION_POOL_CLASS"`
		`value="oracle.security.idm.providers.stdldap.JNDIPool" />`
		` <extendedProperty>`
		`  <name>user.search.bases</name>`
		`  <values>`
		`  <value>cn=users1,dc=us,dc=oracle,dc=com</value>`
		`  <value>cn=users2,dc=us,dc=oracle,dc=com</value>`
		`  </values>`
		` </extendedProperty>`
		`</serviceInstance>`

Users and groups seeded only in the first authenticator are visible, but not from the other authenticators.	By default, users and groups from the first authenticator are authorized.	Starting with 11.1.1.4, you can authorize users and groups from multiple authenticators. Add the following property to the `idstore` instance in the `$DOMAIN_HOME/config/fmwconfig/jps-config.xml` file.
		`<serviceInstance name="idstore.ldap"`
		`  provider="idstore.ldap.provider">`
		`  .....`
		`  **<property name="virtualize" value="true"/>**`
		`  .....`
		`</serviceInstance>`

Table B–22 Troubleshooting Identity Service Issues

Symptom	Possible Cause	Possible Solution
The following exception appears when `myrealm` (the default realm in the Oracle WebLogic Server Administration Console configuration) is passed as a parameter to the identity context to the Identity Service APIs. `Exception seen : Service" Unknown macro: {0} "in configuration" Unknown macro: {1} " could not be initialized. Error in initializing service "Authentication" in configuration "myrealm".`	The human workflow identity service uses the identity context that is set in the `WorkflowIdentityConfig` file (by default, it is `jazn.com`) and not from the Oracle WebLogic Server configuration. Therefore, in the customer code, if `jazn.com` is passed as the identity context, the authenticate API should work fine.	To change the realm name, the `WorkflowIdentityConfig` file can be edited in the System MBean Browser of Oracle Enterprise Manager Fusion Middleware Control. 1. In the navigator, right-click **soa-infra**. 2. Select **SOA Infrastructure** > **Administration** > **System Mbean Browser**. 3. Select **Application Defined Mbeans** > **oracle.as.soainfra.config** > **Server** > **WorkflowIdentityConfig** > **human-workflow** > **WorkflowIdentityConfig.ConfigurationType**. 4. Select the configuration and rename it by invoking the operation **setRealmName**. This change requires a server restart.

After configuring LDAP with Oracle WebLogic Server, the users are visible in the Oracle WebLogic Server Administration Console, but the following error is thrown:
`No Role found matching the criteria`
	The group's base DN is not configured properly. Either the group that is being looked up is not present in LDAP or it may be seeded outside the group base DN that is mentioned while configuring LDAP.	Mention the group base under which all the groups are seeded. This can be performed in two ways.
		Add the base DN under which all the required groups are seeded. For instance, if groups are seeded under:
		`GroupDN 1 : cn=groups1,dc=us,dc=oracle,dc=com`
		`GroupDN 2 : cn=groups2,dc=us,dc=oracle,dc=com`
		`GroupDN 3 : cn=groups3,dc=us,dc=oracle,dc=com`
		Then mention the group base DN as follows:
		`dc=us,dc=oracle,dc=com`
		This is the common DN. If only some group DNs are required (for example, `GroupDN1` and `GroupDN2`), then the following property must be added to `serviceInstance` with the name `idstore.ldap` in the `$DOMAIN_HOME/config/fmwconfig/jps-config.xml` file.
		`<serviceInstance name="idstore.ldap" provider="idstore.ldap.provider">`
		`<property name="idstore.config.provider" value="oracle.security.jps.wls.internal.idstore.WlsLdapIdStoreConfigProvider"/>`
		`<property name="CONNECTION_POOL_CLASS" value="oracle.security.idm.providers.stdldap.JNDIPool"/>`
		`extendedProperty>`
		`<name>group.search.bases</name>`
		`<values>`
		`<value>cn=groups1,dc=us,dc=oracle,dc=com</value>`
		`<value>cn=groups2,dc=us,dc=oracle,dc=com</value>`
		`</values>`
		`</extendedProperty>`
		`</serviceInstance>`

B.5 Event Delivery Network Troubleshooting

This section describes how to troubleshoot Event Delivery Network (EDN) issues.

B.5.1 Tuning EDN Event Bus and Delivery

You can customize the following EDN properties in the System MBean Browser:

1. Right-click **soa-infra**.
2. Select **SOA Infrastructure** > **Administration** > **System MBean Browser**.
3. Expand **Application Defined MBeans** > **oracle.as.soainfra.config** > **Server:** ***soa_server-x*** > **EDNConfig** > **edn**.
4. Double-click an attribute.

Properties	Type	Details
NumberOfThreads	int	Defaults to `3` for the EDN-DB implementation or `2` for the EDN-JMS implementation. Setting this property to `0` is effectively the same as `2` for EDN-DB or `1` for EDN-JMS.
Paused	boolean	Values are `true` or `false`. If `true`, the EDN listener threads are decreased to `0`, which effectively stops the delivery of events. This works for EDN-DB and EDN-JMS.
NumberOfRetrys	int	The maximum number of EDN delivery and redeliveries before moving the event to the error hospital. This parameter works only for one and only one delivery, and not for guaranteed delivery. The maximum allowed value is `5`. If set to less than `5`, then it is effectively the same as `5`. If **NumberOfRetrys** is set to the following values: ■ >= `2`, then EDN delivers the event once, and may try redeliveries up to (**NumberOfRetrys** `-1`) times in case of XA failures. ■ = `1`, then EDN delivers the event only once. There is no retry with XA failures. ■ = `0`, then this is the same as = `1` (that is, EDN delivers an event only once). There is no retry with XA failures. ■ > `5`, then it is effectively the same as **NumberOfRetrys** = `5`.

5. Enter a value, and click **Apply**.

B.5.2 Rolled Back OAOO Event Delivery Messages are Displayed in the Log Files

If the Java debug level is set to **TRACE:16 (FINER)** or a lower value, you may see log messages such as the following:

```
Began XA for OAOO
Rolled back XA for OAOO
```

These are normal messages of OAOO event delivery when there are no events waiting to be delivered. These are *not* error conditions. You can turn off these messages by setting the Java logging level to **TRACE:1 (FINE)** or a higher value.

B.5.3 Events Are Consumed by Multiple Revisions of the Same Composites

By design, different composites can subscribe to the same event. However, it is an error if multiple revisions of the same composite subscribe to the same event. This occurs when you have multiple active versions of the composite that may be a result of a patch failure.

Perform the following steps to determine whether there are multiple, active revisions of the same composite and to retire the composite revision that should not be active:

1. Go to Oracle Enterprise Manager Fusion Middleware Control.
2. Click **SOA** > **soa-infra**.

   In the **Deployed Composites** section of the **Dashboard** tab, you see a list of deployed composite names, revisions, and modes (for example, active).
3. Identify composites with the same name and with an active mode, but with different revisions.
4. Click the composite revision that should not be active.
5. Click the **Retire** button.

B.5.4 Business Event Is Picked Up Twice (Or More) By SOA Server

Business events may be raised from J2EE applications and picked up more than once. Expected behavior is that they must be picked up only once.

Solution

If a patch fails, it may be possible for two versions of a given composite to be active (that is, the older version has not been retired). When multiple versions of a composite are active, they all become subscribers and the event is picked up more than once. This should not happen under normal scenarios. Reapply the patch and ensure that the deployed composite only has one active version.

B.5.5 Some Messages Are Lost Between EDN and Composites or Composites Across Clusters

Messages primarily may get lost for the following reasons:

1. The EDN message was not delivered.
2. The target asynchronous service did not respond.
3. The message was roll backed (though it was not lost; you still see it in Oracle Enterprise Manager Fusion Middleware Control).

This issue typically requires you to identify where the messages are supposed to be and to diagnose the path.

- Check if the Oracle WSM Policy Manager security configurations allow the client to invoke the SOA composite.
- Verify that the transaction is not transient (nonpersistent).

  Oracle BPEL Process Manager uses the dehydration store database to maintain long-running, asynchronous processes and their current state information in a database while they wait for asynchronous callbacks. Storing the process in a database preserves the process and prevents any loss of state or reliability if a system shuts down or a network problem occurs. There are two types of processes in Oracle BPEL Process Manager. These processes impact the dehydration store database in different ways.

  – Transient processes: This process type does not incur any intermediate dehydration points during process execution. If there are unhandled faults or there is system downtime during process execution, the instances of a transient process do not leave a trace in the system. Instances of transient processes cannot be saved in-flight (whether they complete normally or

abnormally). Transient processes are typically short-lived, request-response style processes. The synchronous process you design in JDeveloper is an example of a transient process.

– Durable processes: This process type incurs one or more dehydration points in the database during execution because of the following activities:

  * Receive activity
  * OnMessage branch of a pick activity
  * OnAlarm branch of a pick activity
  * Wait activity

  Instances of durable processes can be saved in-flight (whether they complete normally or abnormally). These processes are typically long-living and initiated through a one-way invocation. Because of out-of-memory and system downtime issues, durable processes cannot be memory-optimized.

- If a composite instance is not visible in Oracle Enterprise Manager Fusion Middleware Control and the SOA Infrastructure is running, check that the **Audit Level** is not set to **Off** in the SOA Infrastructure Common Properties page. This can be checked in Oracle Enterprise Manager Fusion Middleware Control:

  1. Right-click **soa-infra (*SOA_cluster_name*)**.
  2. Select **SOA Administration** > **Common Properties**.

- If the composite instance is available, check the **bpel.config.oneWayDeliveryPolicy** BPEL property value. You can check the value in Oracle Enterprise Manager Fusion Middleware Control:

  1. In the navigation pane, expand **soa-infra (*SOA_cluster_name*)**.
  2. Expand the partition, and select the composite.

     The Dashboard page for the composite is displayed.
  3. In the upper right corner, click the **Show XML Definition** icon.

     The contents of `composite.xml` for that composite are displayed.

  If this is set to **async.cache**, you may lose messages. Set it to **async.persist** for reliable messages. This is typically specified in the BPEL process service component section of the `composite.xml` file, so this can be set for custom composites. If the value is not set in `composite.xml`, the value for **oneWayDeliveryPolicy** in the System MBean Browser in Oracle Enterprise Manager Fusion Middleware Control is used. The following values are possible:

  – **async.persist**: Messages are persisted in the database hash map.
  – **async.cache**: Messages are stored in memory.
  – **sync**: Direct invocation occurs on the same thread.

  For more information about these settings, see the "Deployment Descriptor Properties" appendix and the "Transaction and Fault Propagation Semantics in BPEL Processes" chapter of the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

  For information about setting the **oneWayDeliveryPolicy** property in the System MBean Browser, see the "Configuring BPEL Process Service Engine Properties" section of *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

It is also possible that the transaction has invoked a target asynchronous service that has not responded back. In this case, the composite instance flow shows the call to the target asynchronous service.

- Check if the instance has rolled back and the message is in recovery.
  1. Log in to Oracle Enterprise Manager Fusion Middleware Control.
  2. Right-click **soa-infra (*SOA_cluster_name*)**, and select **Service Engines** > **BPEL**.
  3. Click the **Recovery** tab.

  This may occur if any external references receiving the message are not reachable (for example, an external web service, enterprise applications such as Siebel, and so on).

B.6 Performance Troubleshooting

This section describes how to troubleshoot performance issues.

B.6.1 Optimizing the Loading of Pages with Instance and Fault Metrics

Since production systems can include numerous composite instances and faults, there is a possibility of time-outs in the Oracle Enterprise Manager Fusion Middleware Control pages as information retrieval becomes relatively slow. To optimize the loading performance of Dashboard pages for the SOA Infrastructure, SOA composite applications, service components, and service engines; Deployed Composites page of the SOA Infrastructure and service engines; and partition home page, you can enable two property settings in the SOA Infrastructure Common Properties page that perform the following:

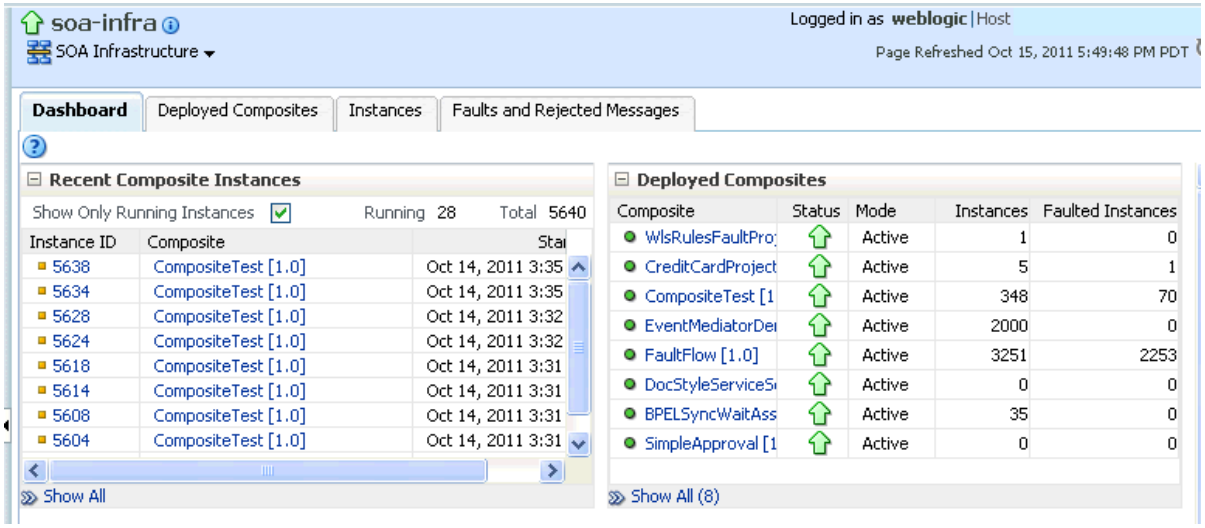
- Disable the loading of all metrics information upon page load. The instances and faults metrics can be obtained on demand from the server.

  Note: This setting disables fault, instance, and count metrics, and not DMS metrics.

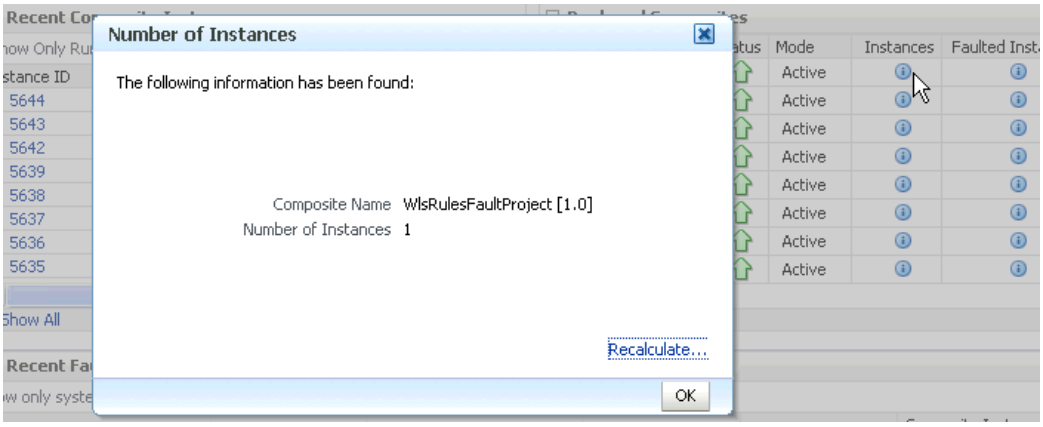
- Restrict the retrieval of instances, faults, and counts to a specified time period (default selection).

To optimize the loading of pages with instance and fault metrics:

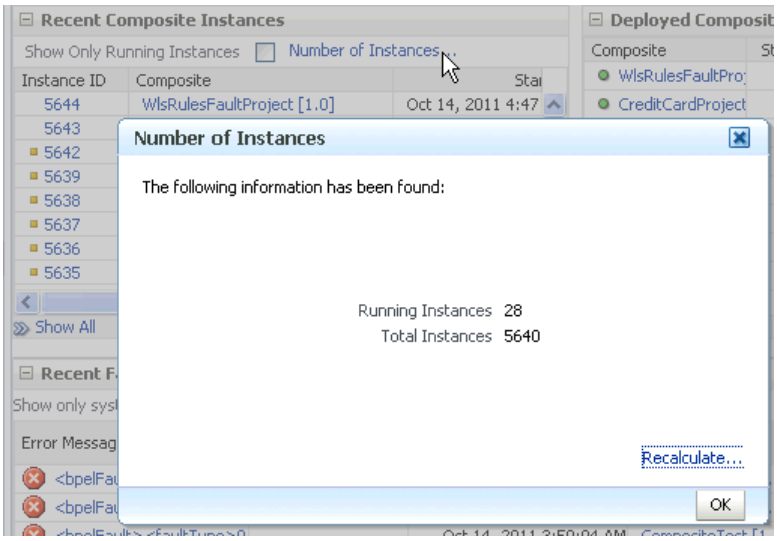
1. In the navigator, click **soa-infra**.
2. Note that values appear in the **Running** and **Total** fields in the **Recent Composite Instances** section and the **Instances** and **Fault Instances** columns of the **Deployed Composites** section. When these values are large, it can take time to load this page and other pages with similar information.



3. From the SOA Infrastructure menu, select **SOA Administration** > **Common Properties**.
4. In the **Display Data Counts** section, select the **Disable fetching of instance and fault count metrics** checkbox.
5. Click **Apply**.
6. Return to the Dashboard page of the SOA Infrastructure.
7. Note that the values that previously displayed have been replaced with links.
8. In the **Deployed Composites** section, click a link for a specific SOA composite application to display details about the number of instances. The values are calculated for the link you select.



9. Click **OK**.
10. Click **Number of Instances**.



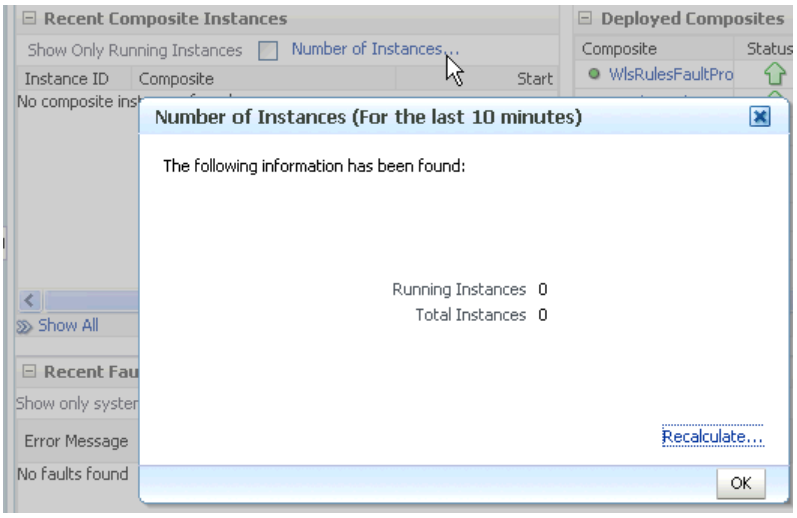
A message displays the number of running and total instances in the SOA Infrastructure.

11. Click **OK**.
12. Return to the SOA Infrastructure Common Properties page.
13. In the **Display Data Counts** section, select the **Restrict display of instances and faults to the last *time_period*** checkbox if it is not already selected. By default, this checkbox is selected and the time period duration is set to **24** hours (one day).
14. Specify a time period during which to retrieve recent instances, faults, and count metrics for display (for example, **10** minutes).
15. Click **Apply**.
16. Return to the Dashboard page of the SOA Infrastructure.
17. Note that the following message is displayed in the upper right section of the page:

    ```
    Recent Instances and Faults for the last 10 minutes
    ```

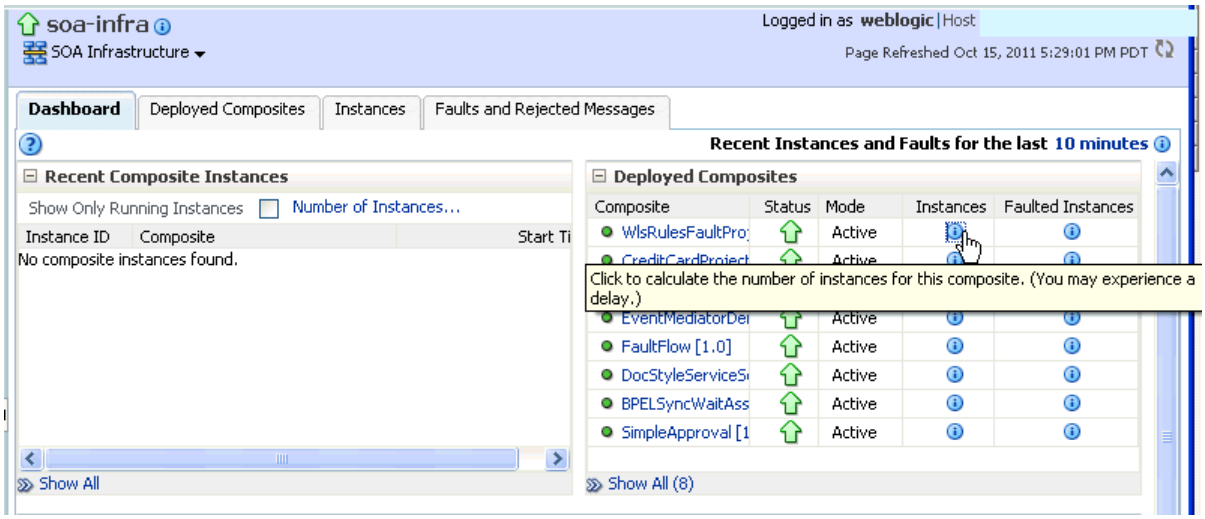
18. Click **Number of Instances**.

    The value that you specified for the time period is displayed in the message title. The number of running and total messages for that time period is also displayed.

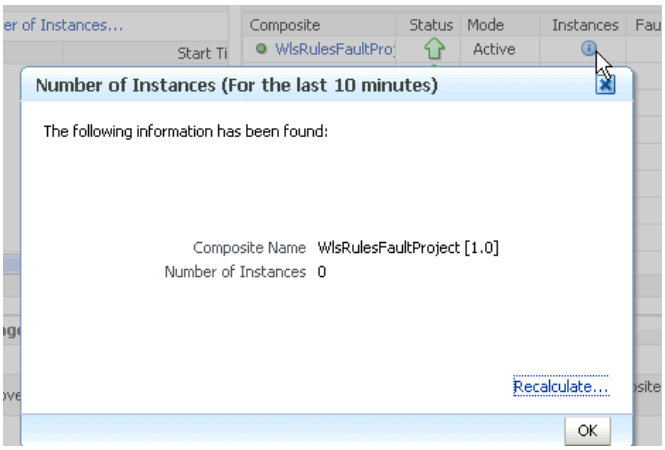


19. Click **OK**.

20. In the **Deployed Composites** section, click a link.



The value that you specified for the time period is displayed in the message title. The number of running and total messages for the selected SOA composite application for that time period is also displayed.



For more information about setting these properties, see Section 3.1, "Configuring SOA Infrastructure Properties."

Notes:

- If you click a link to retrieve instance and fault count metrics, and Oracle Enterprise Manager Fusion Middleware Control times out, increase the transaction timeout property. For more information, see Section B.2.1, "Resolving Connection Timeouts."
- If you click **Recalculate**, and the recalculation occurs quickly, the progress indicator does not have a chance to render. However, any updates to the data are reflected on-screen.

B.6.2 Resolving Message Failure Caused by Too Many Open Files

You can receive the following error at runtime or compilation time, depending on the number of JAR files being used, the use of file descriptors by JDK 6/JRE, or both.

```
Message send failed: Too many open files
```

To resolve this error, increase the number of file descriptors to at least `4096`.

1. Use the `limit` command (for the C shell) or the `ulimit` command (for the Bash shell) to identify the value for `descriptors`. A value of `1024` is typically too low, especially for JDK 6.

```
% limit

cputime      unlimited
filesize     unlimited
datasize     unlimited
stacksize    10240 kbytes
coredumpsize unlimited
memoryuse    unlimited
vmemoryuse   unlimited
descriptors  1024
memorylocked 500000 kbytes
maxproc      46720
```

2. Log in as the `root` user on your operating system.
3. Edit the `/etc/security/limits.conf` file to increase the value for `descriptors`.

   For this example, the `limits.conf` file appears as follows after increasing the limit for all users to `4096`:

```
#<domain>      <type>  <item>         <value>
#

#*               soft    core            0
#*               hard    rss             10000
#@student        hard    nproc           20
#@faculty        soft    nproc           20
#@faculty        hard    nproc           50
#ftp             hard    nproc           0
#@student        -       maxlogins       4

# End of file
@svrgroup    soft    memlock         500000
```

```
@svrgroup    hard    memlock         500000
*           soft    nofile          4096
*           hard    nofile          4096
```

4. Close your terminal and reopen for the change to take effect. A system restart is not required.

B.6.3 Resolving MaxMessageSizeExceededException Errors Caused By Large Payloads

If you provide a large payload (for example, 12 MB) to your deployed SOA composite application, then click **View XML Document** in the audit trail to view the payload, you can encounter `MaxMessageSizeExceededException` errors. This error can be resolved by setting the following JVM parameter.

1. Open the following file:
   - On UNIX operating systems, open `$MIDDLEWARE_HOME/user_projects/domains/`*`domain_name`*`/bin/setDomainEnv.sh`.
   - On Window operating systems, open *`MIDDLEWARE_HOME`*`\user_projects\domains\`*`domain_name`*`\bin\setDomainEnv.bat`.
2. Add the `weblogic.MaxMessageSize` property with the following value:

```
EXTRA_JAVA_PROPERTIES="${EXTRA_JAVA_PROPERTIES}
-Dweblogic.MaxMessageSize=20000000"
export EXTRA_JAVA_PROPERTIES
```

3. Restart the server.

B.6.4 Extending Tablespaces to Avoid Problems at Runtime

If the database tablespace is not extended, runtime processing can be impacted. Messages are not processed or persisted, and exception errors similar to the following can appear in the log files. This is because Oracle BPEL Process Manager relies on the database to store instance data. If the database is not available, runtime processing is impacted.

```
INFO: MediatorServiceEngine returning after processing the request for
operation = processResponse

[EL Warning]: 2009.01.14 11:46:16.783--UnitOfWork(32372128)--Exception
[EclipseLink-4002] (Eclipse Persistence Services - 1.1 (Build
SNAPSHOT-20081007)): org.eclipse.persistence.exceptions.DatabaseException
Internal Exception: java.sql.BatchUpdateException: ORA-01691: unable to
extend lob segment SH_SOAINFRA.SYS_LOB0000145067C00007$$ by 1024 in
tablespace SH_SOAINFRA

Error Code: 1691
Query: InsertObjectQuery(com.collaxa.cube.persistence.dto.AuditTrail@199b33d)
[EL Warning]: 2009.01.14 11:46:16.782--UnitOfWork(32372128)--Exception
[EclipseLink-4002] (Eclipse Persistence Services - 1.1 (Build
SNAPSHOT-20081007)): org.eclipse.persistence.exceptions.DatabaseException
Internal Exception: java.sql.BatchUpdateException: ORA-01691: unable to
extend lob segment SH_SOAINFRA.SYS_LOB0000145067C00007$$ by 1024 in
tablespace SH_SOAINFRA
. . .
. . .
```

Ensure that you set a tablespace to automatically extend itself by a specified amount when it reaches its size limit. If you do not enable autoextend, ensure that you respond when alerted that the tablespace is reaching its critical or warning threshold size. You can respond to size alerts by manually increasing the tablespace size.

B.6.5 Resolving Database Growth Issues Caused by a High Volume of Transactions

If Oracle SOA Suite transactions fail and logs indicate that the database is running out of space, the dehydration store or MDS store may be running out of space due to a high volume of transactions. In the latter example, you see errors such as the following:

```
java.sql.SQLException: ORA-01653: unable to extend table SH_MDS.CUBE_INSTANCE by
16 in tablespace FUSION_TS_TOOLS
```

This indicates that the tablespace is full and the database cannot extend it.

Perform the following tasks.

1. Purge the dehydration store tables periodically, taking into account the appropriate record retention policies and ensuring that the applications have no dependencies on runtime data.

   The purge should be followed by commands to coalesce the space. For the purging strategy to work, it is important to understand how long to retain the data in the database. Factors that drive the retention policy include the following:

   - Legal requirements
   - Line of business requirements
   - Overall company policy on retention of data

   The longer the retention policy, the greater the volume of data that must be stored and, correspondingly, the higher the disk capacity requirements.

   For details on creating a purging strategy, see Chapter 9, "Managing Database Growth."

2. Ensure that the database hardware has sufficient resources to handle the demands of Oracle Database partitioning before configuring your tables for partitioning.

   For the dehydration store, database partitioning using range partitioning and hash partitioning is an optimal solution. Partitioning by definition means storing data in multiple tables to reduce bigger data sets into smaller, more manageable data sets. Partitioning strategies play a large role in easing maintenance overheads (dropping and pruning the partition) and improving performance. Partitioning should at least be done for tables having high activity. This plays a large role in balancing disk I/O and preventing hot disks. One important requirement that you must meet before configuring your tables for partitioning is to ensure that the database hardware has sufficient resources to handle the demands of Oracle Database partitioning. If preproduction testing has indicated that the installation is large, Oracle expects that you have sized your environment (CPU, memory, and disk space) correctly to take advantage of the partitioning features.

3. Tune database parameters for memory, tablespace, and partitions to get maximum performance. For more information, see Section "Tune Database Parameters" of the *Oracle Fusion Middleware Performance and Tuning Guide*.

4. For other tablespaces running out of space, use the following query to check for free tablespace:

   ```
   SELECT Total.tablespace_name "Tablespace Name", Free_space, (total_space-Free_
   ```

```
space) Used_space, total_space, round((Free_space*100/total_space),2) "Free %"
FROM (select tablespace_name, sum(bytes/1024/1024) Free_Space from sys.dba_
free_space group by tablespace_name) Free,
(select tablespace_name, sum(bytes/1024/1024) TOTAL_SPACE from sys.dba_data_
files group by tablespace_name) Total
WHERE Free.Tablespace_name = Total.tablespace_name AND Total.tablespace_name =
 '<tablespacename>'
ORDER BY 5;
```

5. To increase tablespace settings, use the administrator account. For example:

```
alter tablespace tablespace_name add datafile 'datafile_name' size 500m
autoextend on;
```

For more details, see Section B.6.2, "Resolving Message Failure Caused by Too Many Open Files."

B.6.6 Observing Slow Application Performance Such as Longer Time to Serve Pages or Finish Transactions

You can observe slow application performance and/or memory trashing. For example, it may take longer to load and serve pages or to complete composite transactions. Response time may seem slower compared to normal behavior.

There are various reasons for slow performance. It may be due to a large number of servers running on the same host, or there may be a large number of records/sessions/locking in the database. Thread contention can also be a reason for poor performance.

> **Note:** All configuration changes in Oracle Enterprise Manager Fusion Middleware Control may be unavailable due to this problem.

To diagnosis this issue:

1. Check the CPU utilization to see if it is saturated due to a heavy load or too many processes in relation to CPU capacity.

   If CPU utilization is at 100% during normal load hours (the target should be 70-80%), you have no capacity to handle a peak load and the hardware resources are insufficient. Add scale-out servers to handle the additional load.

2. Check applications using Oracle Enterprise Manager Fusion Middleware Control to report on performance. Check the performance of services and the invoke/response times in the BPEL process audit trail.

3. To maximize performance, it is recommended that you not set the logging level higher than the default **INFO** level.

   For debugging purposes, you must set the logging level to the **FINEST** level. However, once issues are resolved, reset the logging level to the default level for best performance. It is also recommended that you set **Audit Level** to **Production** in the SOA Infrastructure Common Properties page. This can be set in Oracle Enterprise Manager Fusion Middleware Control as follows:

   a. Right-click **soa-infra (*SOA_cluster_name*)**.

   b. Select **SOA Administration** > **Common Properties**.

   c. Set **Audit Level** to **Production**.

4. Purge periodically based on retention requirements to maintain any service level agreements (SLAs).

   To identify the tables where data growth may lead to performance issues, see the "Tables Impacted By Instance Data Growth" section in the *Oracle Fusion Middleware Performance and Tuning Guide*.

   For information about using the purge scripts, see Chapter 9, "Managing Database Growth."

5. Check the memory/IO/paging/swapping/CPU usage load statistics using Top or Glance or another monitoring tool.

6. Optimize the JVM to avoid full garbage collection or out-of-memory errors.

   Frequent garbage collection can be either due to higher memory usage or memory leaks.

   a. Ensure that the sum of the maximum heap size of all the JVMs running on your system does not exceed the amount of available physical RAM to avoid operating system level paging activity.

   b. Use the JRockit mission control memory profiling tools to get thread dumps and memory snapshots, which helps Oracle Support Services debug any code issues. For more information about thread dumps, see Chapter 11, "Diagnosing Problems with SOA Composite Applications."

7. Optimize threads to avoid contention. Get a thread dump to investigate and submit it to Oracle Support Services.

8. Check the JVM and thread dumps for methods invoking the database to identify if database performance is a bottleneck.

9. Run database Automatic Workload Repository (AWR) snapshots to identify causes of database performance issues.

10. Ensure that database statistics are updated at regular intervals and other tunable parameters for memory, tablespace, and partitions are used effectively to obtain maximum performance.

    Here are some common tuning recommendations. For more information, see the "Tune Database Parameters" section in the *Oracle Fusion Middleware Performance and Tuning Guide*.

    - Put indexes and tables in as physically separate disk areas if possible.
    - Never put rollback segments with data or index segments.
    - Separate highly active tables and indexes into their own tablespaces.
    - Partition high activity tables and indexes to help balance disk I/O and prevent hot disks.
    - Have processes in place to generate database table statistics at regular intervals.

11. Tune database tables to control the high watermark (HWM) contention of large objects. Tune database advanced queues (AQ) to control high watermark (HWM) contention and ensure consistent performance of producing and consuming messages from AQ.

    The EDN AQ names to be aware of are `EDN_EVENT_QUEUE_TABLE` and `EDN_OAOO_DELIVERY_TABLE`.

12. Tune the BPEL and EDN thread counts to ensure optimal settings (for example, the **Dispatcher Invoke Threads** and **Dispatcher Engine Threads** properties on the BPEL Service Engine Properties page in Oracle Enterprise Manager Fusion Middleware Control). If the thread configuration is too high, the servers run out of memory. If they are too low, the messages start backing up.
13. Tune the BPEL properties to reduce overhead (for example, disable the **ValidateXML** and **StatsLastN** (statistics gathering batch size) properties on the BPEL Service Engine Properties page in Oracle Enterprise Manager Fusion Middleware Control) if they are not needed.
14. In case of integration with packaged applications (for example, Siebel), check if the issue lies with the legacy applications.

For more information about performance tuning the various components, see the "Top Performance Areas" chapter and the "SOA Suite Components" part in the *Oracle Fusion Middleware Performance and Tuning Guide*.

For information about setting properties on the BPEL Service Engine Properties page, see Chapter 12, "Configuring BPEL Process Service Components and Engines."

B.7 Server Troubleshooting

This section describes how to troubleshoot server issues.

B.7.1 Best Practices for Starting and Stopping a Managed Server

As a best practice, it is always recommended that you start and stop a managed server through one, but not both, of the following methods. Do *not* mix these methods, such as starting the managed server from the command line and stopping it from Oracle Enterprise Manager Fusion Middleware Control, or vice versa.

- Oracle Enterprise Manager Fusion Middleware Control

  With this method, the node manager must be up and running. The node manager tracks all managed server startups and shutdowns performed from Oracle Enterprise Manager Fusion Middleware Control. With this method, the server state is not an issue.

- Command line

  With this method, the node manager does not track the server state. Therefore, if you start the server from the command line and shut it down from Oracle Enterprise Manager Fusion Middleware Control, the Oracle WebLogic Administration Server accesses the node manager to determine its status, which returns a state of Unknown.

Perform the following steps to stop and start the server from Oracle Enterprise Manager Fusion Middleware Control.

1. Expand the WebLogic domain.
2. Select the managed server (for example, named soa_server1).
3. Select **Control** > **Shut Down**.
4. Select **Control** > **Start Up**.

For information on starting and stopping managed servers from the command line, see *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

B.7.2 Diagnosing SOA Server Startup Problems

If the SOA server (`soa_infra`) does not start, there can be various reasons for server startup issues. A discovery-based approach to finding the root cause is required. Check the server and diagnostic logs as a first attempt to diagnose the issue. In addition, check the following.

To diagnose server startup issues:

1. Check if the database is not available or there are not enough connections available. Some failures with Oracle SOA Suite runtime can result from database outage/connectivity issues. Perform the following steps:
   a. Log in to Oracle WebLogic Server Administration Console.
   b. In the **Domain Structure**, view the status by selecting **Services** > **Data Sources** > **SOADatasource** > **Monitoring** > **Testing** to test the data source. You can also check this from a SQL prompt.
2. Check the list of ports used for port conflicts.
3. Check if the MDS repository cannot load shared documents (incorrect MDS configuration or the database that holds the MDS schema is not reachable).
4. Check the Coherence configuration if using an Oracle SOA Suite cluster (use of unicast versus multicast).

   This issue only applies after provisioning is complete. Clustering of SOA servers may fail if there are port conflicts (used by Coherence). For example, one SOA server may be picking up the deployment, but the other server is not. In some cases, this may only present the following error:

   ```
   [soa_server1] [ERROR] [] [Coherence] [tid: Logger@352821903 3.6.0.4]
   [ecid: 46f620208907e045:63f295ec:12dd091ec2e:-8000-0000000000000003,1:27187]
   [APP: soa-infra] 2011-01-28 23:06:19.463/414.816 Oracle Coherence GE 3.6.0.4
   <Error> (thread=[ACTIVE] ExecuteThread: '0' for queue: 'weblogic.kernel.Default
    (self-tuning)', member=n/a):
   Error while starting cluster: com.tangosol.net.RequestTimeoutException: Timeout
    during service start: ServiceInfo(Id=0, Name=Cluster, Type=Cluster[[...
   ```

   In this example, Coherence timeouts are prominently available. However, sometimes it presents itself with an unrelated error, such as the following:

   ```
   Error creating bean with name 'SensorManager' defined in ServletContext
   resource
   ```

   The root cause of this is still primarily related to Coherence configuration.
5. Check if both the Administration Server and Managed Server ports are open and accessible.
6. Check if Managed Server startup failed because the Administration Server is not reachable.
7. Check network issues (for example, IP routing filtering/rules that may be causing issues).
8. Check Oracle WebLogic Server LDAP security corruption.

   The Managed Server may report that policies for the application System MBeans Browser already exist or do not exist.

For more information, see the "Setting the Frontend URL for the Administration Console and Setting Redirection Preferences" section and the "Setting the Frontend

HTTP Host and Port" section in the *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite*.

B.7.3 Specifying the Proxy Server

To use system properties to specify the proxy server, write your client application in the standard way, and then specify Java system properties when you execute the client application.

```
setenv PROXY_SETTINGS "-DproxySet=true
-Dhttp.proxyHost=www-myproxy.us.mycompany.com -Dhttp.proxyPort=80
 -Dhttp.nonProxyHosts=localhost|*.us.mycompany.com
  |0:0:0:0:0:0:0:1|fe80:0:0:0:250:56ff:fe31"
```

Note: When you specify values for proxy properties such as `http.proxyHost` and `http.proxyPort`, also specify the `http.nonProxyHosts` property.

B.7.4 Flow Diagram Does Not Display The First Time on Some Lower End Hosts

The flow diagram for an instance ID of a deployed SOA composite application in Oracle Enterprise Manager Fusion Middleware Control may not display the first time on some lower end hosts. Instead, you receive a `failed to load resource` message.

As a workaround, close the flow trace page and click the instance ID to return to the flow trace page.

B.7.5 Accessing Oracle Enterprise Manager Fusion Middleware Control on Dual Stack Hosts that Support IPv4 and IPv6

If you run Oracle SOA Suite on a dual stack host that supports both IPv4 and IPv6, you must update the `etc/hosts` file as shown in Table B–23 for IPv4 clients to access IPv6 URLs in Oracle Enterprise Manager Fusion Middleware Control.

Table B–23 IPv4 and IPv6 Settings in etc/hosts File

On The...	Edit the etc/hosts File as Follows....
On the IPv4 client:	*xx.xxx.xxx.xxx* `myhost10-ipv6` where *xx.xxx.xxx.xxx* is the IP address of IPv6.
On the IPv6 client	`fdf5:74cc:db0a::0:1  myhost10-ipv6 myhost10-ipv6.us.oracle.com` **Note**: Replace `fdf5:74cc:db0a::0:1` with a value appropriate to your host environment.

B.8 Browser Troubleshooting

This section describes how to troubleshoot browser issues.

B.8.1 Limitation on Using the Safari Browser to View WSDL File Content

If you are using the Safari browser, note the following limitation and workaround for viewing WSDL file contents in Oracle Enterprise Manager Fusion Middleware Control. Note also that Mozilla Firefox works correctly and does not require this workaround.

1. Go to the home page for a SOA composite application.

2. Click the **Show WSDL and endpoint URI** link at the top of the page.
3. Click the WSDL link that is displayed.

   This opens a blank page that does *not* display the contents of the selected WSDL.

   As a workaround, perform the following additional steps.
4. In the upper right corner of this page, click the **Display a menu for the current page** icon.
5. Select **View Source** from the menu that is displayed.

   This displays the contents of the selected WSDL in another page.

B.9 Additional Troubleshooting Documentation

Table B–24 describes documentation to see for additional troubleshooting issues.

Table B–24 Additional Troubleshooting Documentation

For Information About...	See...
SOA Infrastructure startup and shutdown issues	Section 3.2, "Stopping and Starting the Managed Server and SOA Infrastructure."
Using WebLogic Diagnostic Framework (WLDF) and Diagnostics Framework (DFW) to diagnosis Oracle SOA Suite problems	Chapter 11, "Diagnosing Problems with SOA Composite Applications."
Managing large documents and metadata and for managing environments with large numbers of instances	Chapter "Managing Large Documents and Large Numbers of Instances" of *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.
Troubleshooting deployment and compilation errors	Section "Testing and Troubleshooting" in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.
Troubleshooting Oracle WSM Policy Manager errors	Section "Diagnosing Problems" chapter in the *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.
Troubleshooting Oracle Business Activity Monitoring (BAM)	Appendix "Troubleshooting" of *Oracle Fusion Middleware User's Guide for Oracle Business Activity Monitoring*.
Troubleshooting Oracle Business Rules	Appendix "Oracle Business Rules Troubleshooting" of *Oracle Fusion Middleware User's Guide for Oracle Business Rules*.
Oracle User Messaging Service	Section 26.7, "Troubleshooting Oracle User Messaging Service."

C

Oracle Enterprise Manager Roles

This appendix describes role-based access and the privileges that users with the administrator, operator, and monitor roles are authorized with when accessing pages in Oracle Enterprise Manager Fusion Middleware Control.

This appendix includes the following topic:

- Section C.1, "Roles and Privileges"

For information about how to create roles, add users to groups, and secure resources with roles and policies, see *Oracle Fusion Middleware Securing Resources Using Roles and Policies for Oracle WebLogic Server* and the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*. Click the **Contents** link in the Console Help to access procedures for performing the above-mentioned tasks.

C.1 Roles and Privileges

Oracle Enterprise Manager Fusion Middleware Control supports the notion of role-based access. Users are mapped to different roles; each role corresponds to a different set of privileges. Using this mechanism, you can provision certain users with simple monitoring privileges (for instance view-only access), while administrators can be granted full access, including the ability to update configurations, restart servers, and so on.

The following roles have been defined for Oracle WebLogic Server in Oracle Enterprise Manager Fusion Middleware Control:

- Administrator

  This role provides complete management and monitoring capabilities.

- Operator

  This role provides restricted management capabilities.

- Monitor

  This role provides read-only capabilities.

C.1.1 Overall Role Functionality Matrix

Table C–1 lists the actions that users with each role can perform.

Table C–1 Role Functionality Matrix

Actions	Monitor	Operator	Administrator
View monitoring metrics	Yes	Yes	Yes

Table C–1 (Cont.) Role Functionality Matrix

Actions	Monitor	Operator	Administrator
View configurations	Yes	Yes	Yes
Update configurations	No	No	Yes
Handle fault actions	No	Yes	Yes
Create instances using the Test Web Service page	Yes	Yes	Yes
Start, stop, retire, and activate a composite	No	Yes	Yes
Execute unit tests	No	Yes	Yes
Attach and detach policies	No	No	Yes
View instances, the flow trace, and the audit trail	Yes	Yes	Yes
View audit trail payloads	Yes	Yes	Yes
Delete instances	No	No	Yes
Start and stop the SOA Infrastructure	No	Yes	Yes
Perform deployment options (deploy, undeploy, and redeploy)	No	Yes	Yes
Modify composite properties (enable payload and audit level)	No	Yes	Yes
Create partitions	No	No	Yes
Delete partitions	No	No	Yes
Bulk composite lifecycle management (start all, stop all, retire all, and activate all)	No	Yes	Yes

C.1.2 SOA Infrastructure Page Access

Table C–2 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–2 SOA Infrastructure Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Dashboard** tab	Monitor	Yes	Yes	Yes
View **Deployed Composites** tab	Monitor	Yes	Yes	Yes
▪ Start/stop (SOA Infrastructure)	Operator	No	Yes	Yes
▪ Activate/retire	Operator	No	Yes	Yes
▪ Deployment options	Operator	No	Yes	Yes
▪ Set as default	Operator	No	Yes	Yes
View **Instances** tab	Monitor	Yes	Yes	Yes
▪ Delete options (all)	Administrator	No	No	Yes
▪ Abort	Administrator	No	No	Yes

Table C–2 (Cont.) SOA Infrastructure Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Faults and Rejected Messages** tab	Monitor	Yes	Yes	Yes
■ Recovery actions	Operator	No	Yes	Yes
■ Delete rejected messages	Administrator	No	No	Yes

C.1.3 SOA Infrastructure Menu Access

Table C–3 lists the lowest role that a user must have to access the options on this menu and the privileges that each role has on the menu options.

Table C–3 SOA Infrastructure Menu

Menu Items	Lowest Role for Accessing	Monitor	Operator	Administrator
Control	Operator	No	Yes	Yes
SOA Deployment	Operator	No	Yes	Yes
Logs >**Log Configuration**	Administrator	No	No	Yes
Other menu items	Monitor	Yes	Yes	Yes

C.1.4 SOA Composite Menu Access

Table C–4 lists the lowest role that a user must have to access the options on this menu and the privileges that each role has on the menu options.

Table C–4 SOA Composite Menu

Menu Items	Lowest Role for Accessing	Monitor	Operator	Administrator
SOA Deployment	Operator	No	Yes	Yes
Test Service	Monitor	Yes	Yes	Yes
Other menu items	Monitor	Yes	Yes	Yes

C.1.5 Composite Home Page Access

Table C–5 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–5 Composite Home Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Dashboard** tab	Monitor	Yes	Yes	Yes
■ Test composite service action	Operator	No	Yes	Yes
■ Activate/retire action	Operator	No	Yes	Yes
■ Start/stop action	Operator	No	Yes	Yes
■ Property changes (settings)	Operator	No	Yes	Yes
View **Instances** tab	Monitor	Yes	Yes	Yes
■ Delete/abort actions	Administrator	No	No	Yes

Table C–5 (Cont.) Composite Home Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Faults** tab	Monitor	Yes	Yes	Yes
■ Fault recovery actions	Operator	No	Yes	Yes
■ Delete rejected messages	Administrator	No	No	Yes
View **Unit Test** tab	Monitor	Yes	Yes	Yes
■ Execute test action	Operator	No	Yes	Yes
View **Policies** tab	Monitor	Yes	Yes	Yes
■ Attach/detach action	Administrator	No	No	Yes

C.1.6 BPEL Process Service Engine Access

Table C–6 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–6 BPEL Process Service Engine

Menu Items	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Dashboard** tab	Monitor	Yes	Yes	Yes
View **Statistics** tab	Monitor	Yes	Yes	Yes
View **Instances** tab	Monitor	Yes	Yes	Yes
View **Faults** tab	Monitor	Yes	Yes	Yes
■ Fault recovery actions (abort, retry, and so on)	Operator	No	Yes	Yes
View **Deployed Components** tab	Monitor	Yes	Yes	Yes
Message Recovery tab	Monitor	Yes	Yes	Yes
■ BPEL message recovery action	Operator	No	Yes	Yes
View Configuration (Properties page)	Monitor	Yes	Yes	Yes
■ **Apply** button	Administrator	No	No	Yes
■ **Add** button	Administrator	No	No	Yes

C.1.7 Oracle Mediator Service Engine Access

Table C–7 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–7 Mediator Service Engine

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Dashboard** tab	Monitor	Yes	Yes	Yes
View **Statistics** tab	Monitor	Yes	Yes	Yes
View **Instances** tab	Monitor	Yes	Yes	Yes

Table C–7 (Cont.) Mediator Service Engine

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Faults** tab	Monitor	Yes	Yes	Yes
▪ Fault recovery action (abort, retry, and so on)	Operator	No	Yes	Yes
View **Deployed Components** tab	Monitor	Yes	Yes	Yes
View Configuration (Properties page)	Monitor	Yes	Yes	Yes
Apply button	Administrator	No	No	Yes

C.1.8 Human Workflow Service Engine Access

Table C–8 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–8 Human Workflow Service Engine

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Dashboard** tab	Monitor	Yes	Yes	Yes
View **Statistics** tab	Monitor	Yes	Yes	Yes
View **Instances** tab	Monitor	Yes	Yes	Yes
View **Faults** tab	Monitor	Yes	Yes	Yes
▪ Fault recovery action	Operator	No	Yes	Yes
View **Deployed Components** tab	Monitor	Yes	Yes	Yes
View **Notification Management** tab	Monitor	Yes	Yes	Yes
▪ Send notifications	Administrator	No	No	Yes
Configuration (Properties page)	Monitor	Yes	Yes	Yes
▪ **Apply** button	Administrator	No	No	Yes

C.1.9 Business Rules Service Engine Access

Table C–9 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–9 Business Rules Service Engine

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Dashboard** tab	Monitor	Yes	Yes	Yes
View **Instances** tab	Monitor	Yes	Yes	Yes
View **Faults** tab	Monitor	Yes	Yes	Yes
View **Deployed Components** tab	Monitor	Yes	Yes	Yes

C.1.10 BPEL Process Service Component Home Page Access

Table C–10 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–10 BPEL Process Service Component Home Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Dashboard** tab	Monitor	Yes	Yes	Yes
View **Instances** tab	Monitor	Yes	Yes	Yes
View **Faults** tab	Monitor	Yes	Yes	Yes
▪ Fault recovery action	Operator	No	Yes	Yes
View **Policies** tab	Monitor	Yes	Yes	Yes
▪ Attach/detach action	Administrator	No	No	Yes

C.1.11 Oracle Mediator Service Component Home Page Access

Table C–11 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–11 Mediator Service Component Home Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Dashboard** tab	Monitor	Yes	Yes	Yes
View **Instances** tab	Monitor	Yes	Yes	Yes
View **Faults** tab	Monitor	Yes	Yes	Yes
▪ Fault recovery action	Operator	No	Yes	Yes
View **Policies** tab	Monitor	Yes	Yes	Yes
▪ Attach/detach action	Administrator	No	No	Yes

C.1.12 Human Task Service Component Home Page Access

Table C–12 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–12 Human Task Service Component Home Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Dashboard** tab	Monitor	Yes	Yes	Yes
View **Instances** tab	Monitor	Yes	Yes	Yes
View **Faults** tab	Monitor	Yes	Yes	Yes
▪ Fault recovery action	Operator	No	Yes	Yes
View **Policies** tab	Monitor	Yes	Yes	Yes
▪ Attach/detach action	Administrator	No	No	Yes
View **Administration** tab	Monitor	Yes	Yes	Yes
▪ Apply changes	Administrator	No	No	Yes

C.1.13 Decision Service Component Home Page Access

Table C–13 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–13 Decision Service Component Home Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Dashboard** tab	Monitor	Yes	Yes	Yes
View **Instances** tab	Monitor	Yes	Yes	Yes
View **Faults** tab	Monitor	Yes	Yes	Yes
▪ Fault recovery action	Operator	No	Yes	Yes
View **Policies** tab	Monitor	Yes	Yes	Yes
▪ Attach/detach action	Administrator	No	No	Yes

C.1.14 Flow Trace Page Access

Table C–14 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–14 Flow Trace Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View Flow trace	Monitor	Yes	Yes	Yes

C.1.15 Audit Trail Access

Table C–15 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–15 Audit Trail Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Audit Trail** tab	Monitor	Yes	Yes	Yes
▪ Audit trail payloads	Monitor	Yes	Yes	Yes
View **Flow Debug** tab	Monitor	Yes	Yes	Yes
View **Sensors** tab	Monitor	Yes	Yes	Yes
View **Fault Recovery** tab	Monitor	Yes	Yes	Yes
▪ Recovery action	Operator	No	Yes	Yes

C.1.16 Services Home Page Access

Table C–16 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–16 Services Home Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Dashboard** tab	Monitor	Yes	Yes	Yes

Table C–16 (Cont.) Services Home Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Policies** tab	Monitor	Yes	Yes	Yes
■ Attach/detach action	Administrator	No	No	Yes
View **Faults** tab	Monitor	Yes	Yes	No
■ Delete rejected messages	Administrator	No	No	Yes
View Properties	Monitor	Yes	Yes	Yes
■ Apply changes	Administrator	No	No	Yes
■ Add properties	Administrator	No	No	Yes

C.1.17 References Home Page Access

Table C–17 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–17 References Home Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Dashboard** tab	Monitor	Yes	Yes	Yes
View **Policies** tab	Monitor	Yes	Yes	Yes
■ Attach/detach action	Administrator	No	No	Yes
View **Faults** tab	Monitor	Yes	Yes	Yes
■ Delete rejected messages	Administrator	No	No	Yes
View Properties	Monitor	Yes	Yes	Yes
■ Apply changes	Administrator	No	No	Yes
■ Add properties	Administrator	No	No	Yes

C.1.18 Oracle B2B Pages Access

Table C–18 lists the lowest role that a user must have to access these pages and the privileges that each role has on these pages.

Table C–18 B2B Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **B2B Configuration** page	Monitor	Yes	Yes	Yes
■ Apply changes	Administrator	No	No	Yes
View **B2B Bindings** page	Monitor	Yes	Yes	Yes

C.1.19 Business Events Page Access

Table C–19 lists the lowest role that a user must have to access this page and the privileges that each role has on this page.

Table C–19 Business Events Page

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View **Events** tab	Monitor	Yes	Yes	Yes
■ Subscribe/test	Administrator	No	No	Yes
■ Show event definition	Monitor	Yes	Yes	Yes
View **Subscriptions** tab	Monitor	Yes	Yes	Yes
■ Add/edit/delete subscriptions	Administrator	No	No	Yes
■ Manage database agents	Administrator	No	No	Yes
View **Faults** tab	Monitor	Yes	Yes	Yes
■ Retry/abort	Operator	No	Yes	Yes

C.1.20 System MBean Browser Access

Table C–20 lists the lowest role that a user must have to access this browser and the privileges that each role has on this page.

Table C–20 System MBean Browser

Page Elements	Lowest Role for Accessing	Monitor	Operator	Administrator
View Configuration	Monitor	Yes	Yes	Yes
■ Add/apply changes	Administrator	No	No	Yes

Index

A

abort
  recovery action, 8-17, 8-28, 13-9, 14-3, 14-7, 17-3, 17-7
actionable email account, 20-6
  configuring, 20-4
activating
  a composite revision, 7-13, 7-17
activation specification
  definition, 34-4
active data cache
  monitoring, 24-2
activities
  configuring for automatic recovery, 12-4
  for BPEL 2.0 in process flows and audit trails, 13-10
activity message types
  Message State list is not applicable, 14-10
adapters
  See Oracle Adapters
addresses
  configuring multiple send addresses, 20-12
administration tasks
  for application developers, 1-22
administrator
  role, C-1
ant
  creating partitions, 7-34
  deployment, 7-5
application developer
  administration tasks, 1-22
application policies
  creating, 2-6
application roles
  creating, 2-6
  demo user community, A-4
  managing in Oracle Enterprise Manager, 20-28
  Oracle Business Activity Monitoring (BAM), 25-9
  seeding with WLST scripts, 20-28
assertions
  definition, 1-21, 7-22
AsynchAuditBatchSize property
  storing multiple audit trail messages in a single transaction, 12-3
audit level
  BPEL process service engine, 12-2
  BPMN process service engine, 37-2
  configuring to display adapter header properties, 34-3
  generating orphaned service component instances, 8-8
  impact of disabling audit tracking across multiple composite flows, 7-18
  order of precedence of execution, 1-13, 7-18
  overriding the audit level defined at the SOA Infrastructure level, 7-18
  relation to partitioning the COMPOSITE_INSTANCE table, 9-13
  setting at the BPEL process service component level, 12-8
  setting at the SOA composite application level, 7-18
  SOA composite application, 7-18
  SOA Infrastructure, 3-3
audit tracking
  order of precedence of execution, 1-13
audit trail
  configuring the storage in one transaction, 14-14
  for BPEL 2.0 projects, 13-10
  monitoring a BPEL process service component, 13-1
  not generated when composite audit level is off, 1-14
  not supported for spring service component, 1-8
  rolled back transaction error message, 13-6
  sensors, 13-2, 38-2
  threshold setting, 12-2
  viewing for a BPMN process service component, 38-1
audit trail threshold, 37-2
AuditConfig property, 3-8
  bpelRecoveryStatus, 4-3, 13-4
  excludeBpelMaxCreationTime key, 4-3, 8-10, 8-13
authentication providers
  adding, 20-14
automatic recovery
  configuration, 12-4
  in clustered environments, 12-6
  is not recovering a specific BPEL instance, B-8
automatic release timers
  globally disabling for Oracle BPM Worklist, 20-11

automating
the testing of SOA composite applications, 7-20

B

B2B *See* Oracle B2B
backup, 1-16
BAM
See Oracle Business Activity Monitoring
binding components
can only publish web services to the UDDI registry, 36-5
configuring, 34-1
configuring adapters, 34-4
configuring the publishing of web services to UDDI, 36-5
configuring web services, 34-3
definition, 1-9
direct binding, 35-2
endpoint properties, 34-27
faults, 35-1
impact when a composite is shut down, 7-13, 7-17
instances, 35-1
limitation on policy attachment errors for service binding components, 36-3
limitation on policy attachments, 36-1
limitation on publishing references to the UDDI registry, 36-5
managing, 36-1
monitoring, 4-6, 35-1
monitoring instances and faults, 35-1
monitoring rejected messages, 35-3
Oracle AQ Adapter properties, 34-4
Oracle AQ Socket properties, 34-26
Oracle Database Adapter properties, 34-5
Oracle File Adapter properties, 34-9
Oracle FTP Adapter properties, 34-14
Oracle JMS Adapter properties, 34-21
Oracle MQ Series Adapter properties, 34-23
policies, 36-1
properties, 34-1
publishing to the UDDI registry, 36-4
references, 1-9
rejected messages, 35-3
restrictions on specifying multiple bindings, 36-1
services, 1-9
BPEL 2.0
audit trail, 13-10
behavior of activity sensors in compensate and compensateScope activities, 13-17
process flow, 13-10
BPEL process service components
audit trail and process flow, 13-1
configuring, 12-1
defining a fault policy, 1-17
definition, 1-7
fault recovery, 14-1
faults, 13-13, 14-1
instance states remain as completed even if the composite instance state becomes stale, 7-6
instances, 13-13, 13-15
limitations on selecting the option to continue instances upon redeployment, 7-7
managing, 14-1
monitoring, 13-1, 13-15
policies, 14-4
recovery message information displayed in the Flow Trace page, 13-3
sensor data, 13-16
BPEL process service engines
audit level, 12-2
configuring, 12-1
deployed processes, 13-23, 16-15
fault recovery, 14-5
faults, 13-18, 14-5
instances, 13-18, 13-22
managing, 14-1
message failure and recovery, 14-7
monitoring, 13-1
monitoring faults, 13-18
monitoring instances, 13-18
payload validation, 12-3, 37-3
request and thread statistics, 13-21
BPEL processes
automatic recovery configuration, 12-4
automatic recovery in cluster environments, 12-6
setting the instance name during design time, 7-20
starting and stopping a managed server on which the SOA Infrastructure is deployed in the middle of BPEL processing, 7-19
BPEL sensors
disabling at the service engine level, 12-3
disabling at the SOA composite application level, 7-19
order of precedence for setting, 7-19
BPMN process service components
audit trail and process flow, 38-1
behavior of instances after redeployment, 7-7
fault recovery, 39-1
faults, 38-7, 39-1
instances, 38-7, 38-9
monitoring, 38-1
policies, 39-4
selecting the option to continue instances upon redeployment, 7-7
BPMN process service engines
audit level, 37-2
deployed processes, 38-14
fault recovery, 39-5
faults, 38-10, 39-5
instances, 38-10, 38-13
message failure and recovery, 39-7
monitoring, 38-1
request and thread statistics, 38-12
browsers
Copy details to clipboard link does not appear with Mozilla Firefox, 13-7
limitation on using the Safari Browser to view

WSDL file content, B-55
supported, 2-1
bulk recovery
cannot perform bulk fault recovery on the Faults tab of a flow trace, 13-9
of faults, 8-18, 8-29
business event messages
cannot be recovered, 8-18, 8-29
business events
BPEL component in a retired composite cannot receive a business event, 33-5
consistency levels, 33-4
database agent subscriptions, 33-2
definition, 33-1
events consumed by multiple revisions of the same composite, B-41
fault recovery, 33-6
faults, 33-6
managing, 33-1
messages are lost between EDN and composites or composites across clusters, B-42
pausing the delivery of events, B-41
picked up twice or more by the SOA server, B-42
policies on event subscriptions are not supported, 33-4
receiving events in a composite with multiple revisions, 33-5
rolled back OAOO event messages appear in the log files, B-41
service component subscriptions, 33-2
setting the number of retries, B-41
subscribers, 33-4
subscribing to, 33-1
troubleshooting, B-40
tuning event delivery, B-41
business faults, 1-17
business monitors
BPEL monitors, 7-36
BPEL sensors, 7-36
BPMN measurements, 7-36
enabling and disabling, 7-36
setting the status level for all sensors, 7-36
business rule execution tracing, 19-12
audit levels, 19-13
development level, 19-13
production level, 19-19
business rule messages
cannot be recovered, 8-18, 8-29
business rules
engine cache and operation statistics, 19-4
monitoring, 19-1
business rules service engines
deployed components, 19-9
faults, 19-7
instances, 19-5

C

callback messages
configuring automatic recovery attempts, 12-8
configuring unresolved messages for automatic recovery, 12-4
performing manual message recovery, 14-7
recovering in resolved and undelivered states, 14-10
stranded, 14-10
callback server URL
property, 3-5
property change requires a restart, 3-5
capture composite instance state
displaying created instances immediately even if a constraint is defined, 7-17
if disabled, the Recent Instances section does not show instances requiring fault recovery as running, 6-3
impact of creating a new instance when this property is disabled, 3-4
meaning of ? icon, 8-7
property, 3-4
case agnostic group names
enabling in human tasks, 20-29
chunking
enabling data checking for SOAP over HTTP deliveries, 3-7
specifying a chunk size, 3-7
clusterDbTimeRefresh property
definition, 12-7
clusters
automatic recovery of BPEL processes, 12-6
configuring master node recovery, 12-6
component tables
partitioning, 9-11
composite sensors
audit trail, 13-2, 38-2
searching for, 8-8
composite URLs
editing, 8-3
configuration plan
selecting in the Deploy SOA Composite wizard, 7-3
consistency levels
of business events, 33-4
continue
recovery action, 8-17, 8-29, 13-9, 14-3, 14-7, 17-3, 17-7
conversation ID
creating on the Test Web Service page, 1-6
definition, 1-6
generating, 1-6
not creating on the Test Web Service page, 1-6
Copy details to clipboard link
does not appear with Mozilla Firefox, 13-7
cross references
Oracle Mediator, 18-1
custom drivers
changing the driver name, 3-16

D

data sources

creating and managing for adapters in Oracle WebLogic Server Administration Console, 2-16
definition, 3-7
managing SOA Infrastructure data sources from Oracle WebLogic Server Administration Console, 2-16
properties, 3-7
database connections
resolving slow connections, B-6
database growth
delete before purging large amounts of data, then add the indexes back in, 9-7
developing a purging and partitioning methodology, 9-3
executing the purge scripts, 9-10
instance purging states, 9-8
looped purge in parallel script, 9-7
looped purge script, 9-5
management, 9-1
managing with the purge script, 9-4
moving long running instances to a different partition, 9-17
partial partitioning of components, 9-19
purge scripts are only supported on Oracle databases, 9-5
purge scripts supported on different Oracle SOA Suite releases, 9-5
purge scripts to run when upgrading, 9-5
purging the instances of a specific SOA composite application, 9-9
referential integrity and equipartitioning, 9-2
resequenced message purge states for Oracle Mediator, 9-8
resolving issues caused by high transaction volume, B-50
selecting the schema partition keys, 9-3
database schemas
components with, 9-2
purge scripts are only supported on Oracle databases, 9-5
reducing the size of, 9-3
database tables
partitioning, 9-2
decision service components
definition, 1-8
deployed components, 19-9
faults, 19-7
instances, 19-10
log files, 19-20
monitoring, 19-1
statistics, 19-4
decision tracing
See business rule execution tracing
default composite revision
distinguishing the default revision from other revisions, 1-20, 7-12, 7-14
impact of retiring, 7-13, 7-14
impact of undeploying, 7-14, 7-15
impact of undeploying when no other active, available revisions exist, 7-9, 7-15
retiring, 7-13
setting, 7-14
undeploying, 7-9
undeploying causes the next active, available revision to become the default, 7-9
Delete Rejected Messages option
when to use, 8-31, 8-33
demo user community, A-1
definition, A-1
groups, A-3
installing in the database, A-1
must run the installation script locally in a domain, A-1
organizational hierarchy, A-2
permissions defined in the WorkflowPermission class, A-6
roles granted to and owned by users, A-5
soa-infra application roles, A-4
SOATestDemoApp application roles, A-5
users, A-1
deployment
ant, 7-5
composites with task flows, 7-4
exporting a deployed SOA composite application, 7-27
from SSO-enabled Oracle Enterprise Manager Fusion Middleware Control, 7-2
guidelines, 7-2
issues with deploying the same composite with a human workflow into multiple partitions, 7-4
lifecycle states, 7-10
managing the state of deployed composites, 7-11, 7-15
of SOA composite applications, 7-14
restrictions on cross references between composites, 7-2
revisions of different composites, 7-2
selecting partitions, 7-3
SOA composite applications, 7-2
using the Redeploy SOA Composite wizard to redeploy an existing instance, 7-2
WLST, 7-5
Diagnostic Framework
definition, 11-3
diagnostics
frameworks, 11-1
getting dump descriptions, 11-6
incident processing rules, 11-4
listing the supported dumps, 11-6
manually executing dumps, 11-27
notification creation, 11-23
notifications, 11-2
querying problems and incidents, 11-29
supported database diagnostic dumps, 11-10
supported deployed composite metadata diagnostic dumps, 11-11
supported deployed composite WSDL/schema cache diagnostic dumps, 11-16

supported diagnostic scenarios and MBeans, 11-2
supported event diagnostic dumps, 11-14
supported instance audit trail diagnostic dumps, 11-13
supported runtime environment diagnostic dumps, 11-6
supported runtime platform configuration diagnostic dumps, 11-7
supported SOA dumps, 11-5
using Diagnostic Framework, 11-3
using WLDF, 11-2
viewing incident packages, 11-29
watch creation, 11-23
watches, 11-2
direct binding
cannot configure properties for, 34-1
definition, 35-2
monitoring, 35-2
no policies and properties tabs are available, 35-2
Direct Binding Invocation API
inbound invocations of composites and outbound invocations of Oracle Service Bus (OSB), 10-2
direct Java invocations
using, 3-17
Disable Alert link
definition, 4-3, 6-3
dispatcher engine threads, 12-3, 37-3
dispatcher invoke threads, 12-3, 37-3
dispatcher system threads, 12-3, 37-3
distributed cache
storing instance and callback message data, 14-11
distribution lists
Oracle Business Activity Monitoring (BAM), 23-11
downloading
the deployable JAR file, 7-27
drivers
configuring, 26-6
deploying Oracle User Messaging Service, 28-1
deploying with Oracle Enterprise Manager Fusion Middleware Control, 28-2
deploying with Oracle Fusion Middleware Configuration Wizard, 28-4
deploying with WLST, 28-2
deploying with wsadmin, 28-5
email driver, 26-15
messaging extension driver, 26-9
properties, 26-8
proxy driver, 26-32
securing driver properties with passwords, 26-9
SMPP driver, 26-19
undeploying, 28-9
unregistering, 28-9
VoiceXML driver, 26-27
worklist driver, 26-29
XMPP driver, 26-24
dumps
getting descriptions of, 11-6
listing the supported dumps, 11-6
manually executing, 11-27
supported database diagnostic dumps, 11-10
supported deployed composite metadata diagnostic dumps, 11-11
supported deployed composite WSDL/schema cache diagnostic dumps, 11-16
supported event diagnostic dumps, 11-14
supported instance audit trail diagnostic dumps, 11-13
supported runtime environment diagnostic dumps, 11-6
supported runtime platform configuration diagnostic dumps, 11-7
supported with Oracle SOA Suite, 11-5
durable BPEL processes
definition, 7-7
limitations on selecting the option to continue instances upon redeployment, 7-7
durable processes
definition, B-43
dynamic assignments, 20-6
functions, 20-5

E

ECIDs
adding an index on the DLV_MESSAGE.ECID column to improve SQL performance when searching messages for a specific ECID, 14-9
copying from the Warning dialog on the Flow Trace page, 13-4
definition, 13-2, 14-8, 38-2
restrictions, 13-4, 38-3
searching for, 8-7, 8-13
specifying when creating a message recovery search criteria, 14-8
EJB transaction timeouts
updating, B-6
email addresses
configuring the outgoing email address, 20-4
configuring the Reply To address, 20-4
correcting, 22-8
email driver
client API, 26-19
common properties, 26-16
configuring, 26-15
custom properties, 26-17
interoperability, 26-15
email notification messages
configuring the number of, 20-12
email server
adding the hostname and IP address to the /etc/hosts file, 20-3
IMAP, 20-3
SMTP, 20-3
emulations, 1-21
endpoint addresses
displaying the addresses of all external services, 7-19
endpoint properties
definition, 34-4

inbound adapters, 29-1
outbound adapters, 29-6
endpoint reference
changing for Oracle Service Registry, 34-29
endpoint URLs
editing, 8-3
specifying during instance creation, 8-2
engine cache and operation statistics
for business rules, 19-4
enterprise message sources, 24-7
equipartitioning
definition, 9-2
event definition language
definition, 33-2
Event Delivery Network
definition, 33-1
EDN-DB, 33-1
EDN-JMS, 33-1
event engine
monitoring, 24-4
exhausted messages
resetting to the undelivered state, 14-11
exhausted state
definition, 14-8
message recovery, 14-8
exporting
a deployed SOA composite application, 7-27

F

Facade API
examples, 10-4
finding composite and component instances, 10-4
oracle.soa.management.facade.Locator interface, 10-1
overview of interfaces, 10-2
programmatically managing SOA composite applications, 10-1
retrieving the state of a composite, 10-4
top level entry point, 10-1
Farm home page
navigating to Oracle SOA Suite pages, 2-16
fault policy, 8-17
creating, 1-17, 8-19, 8-21, 8-24, 8-27
defining, 8-15
definition, 1-16
required for BPEL process fault recovery, 1-17
Fault Time From field, 3-6
faults
automatic recovery configuration, 12-4
automatic recovery configuration for invoke and callback messages, 12-8
automatic recovery in clustered environments, 12-6
binding components, 35-1
BPEL process service component fault recovery, 14-1
BPEL process service engine fault recovery, 14-5
BPMN process service component fault recovery, 39-1
BPMN process service engine fault recovery, 39-5
bulk recovery, 8-17, 8-18, 8-29
bulk recovery at SOA Infrastructure level does not permit a check of composite states, 8-18
business events, 33-6
business events fault recovery, 33-6
business fault definition, 1-17
business rules service engines, 19-7
example of bulk-fault recovery for BPEL processes, 8-21
example of bulk-fault recovery for BPMN processes, 8-23
example of bulk-fault recovery for Oracle Mediator, 8-26
example of single-fault recovery for BPEL processes, 8-20
example of single-fault recovery for BPMN processes, 8-22
example of single-fault recovery for Oracle Mediator, 8-25
examples of BPEL process fault recovery, 8-19
examples of BPMN process fault recovery, 8-21
examples of nonrecoverable faults, 1-17
examples of Oracle Mediator fault recovery, 8-24
examples of recoverable faults, 1-17
human workflow fault recovery, 22-3, 22-6
human workflow service components, 21-1, 22-6
human workflow service engines, 21-6, 22-3
in audit trail, 13-2, 38-2
inbound adapters, 30-1
options for single-fault recovery, 8-16
Oracle Mediator, 17-1, 17-5
Oracle Web Service Manager fault definition, 1-17
outbound adapters, 30-7, 30-8
recovery, 1-16, 8-16
recovery actions, 8-17, 8-28
recovery of SOA composite application faults, 4-4, 6-4
SOA composite application fault recovery, 8-27
SOA Infrastructure, 8-15
SOA Infrastructure fault recovery, 8-15
system fault definition, 1-17
types, 1-17
File Adapter logs, 30-11
flow traces
canceled onMessage branches are not displayed, 13-6
for composite-to-composite invocations on multiple servers, 13-10
impact when instance tracking is disabled, 7-18
issues viewing on lower end hosts, B-55
viewing faults, 13-2
viewing sensors, 13-2
functions
restrictions on adding multiple properties to a function, 20-7

G

Go to BPEL Recovery Console link
  definition, 4-3, 6-3
governance
  definition, 1-20
Gridlink
  definition, 1-16
group names
  enabling case agnostic group names in human tasks, 20-29
groups
  demo user community, A-3
  partitioning, 9-13
  seeding with LDAP tools, 20-26
GUID attribute
  mapping requirements, 20-25

H

heartBeatInterval property
  definition, 12-7
HTTP binding
  cannot change httpBinding property, 34-1
human workflow
  configuring Oracle HTTP Server for task form attachments, 20-9
  deploying composites with task flows, 7-4
  faults not persisted in the dehydration store, 8-16, 8-28
  moving data from a test to a production environment, 22-9
  notification properties, 20-1
  notifications, 22-7
  prerequisites for configuring notification properties, 20-2
  task service properties, 20-5
  task status, 21-3
  troubleshooting, B-13
  troubleshooting AMX extension issues, B-29
  troubleshooting design time at runtime issues, B-25
  troubleshooting human workflow API usage issues, B-26
  troubleshooting identity service issues, B-38
  troubleshooting notification issues, B-15
  troubleshooting Oracle BPM Worklist/task region issues, B-30
  troubleshooting Oracle JDeveloper data control/form generation issues, B-26
  troubleshooting service/System MBean Browser issues, B-27
  troubleshooting task action issues, B-14
  troubleshooting task assignment, routing, and escalation issues, B-13
  troubleshooting task comments/attachment issues, B-24
  troubleshooting task form/action issues, B-22
  troubleshooting task history issues, B-20
  troubleshooting task mapping attribute issues, B-18
  troubleshooting task report issues, B-19
  troubleshooting task view issues, B-17
  troubleshooting test-to-production issues, B-34
  WSDL files security, 5-12
human workflow service components
  configuring, 20-1
  definition, 1-7
  fault recovery, 22-6
  faults, 21-1, 22-6
  instances, 21-1, 21-5
  managing, 22-1
  monitoring, 21-1
  performing fault recovery from Oracle BPM Worklist, 1-17, 8-15
  policies, 22-1
  recommendation not to use oracle/wss10_saml_token_service_policy in a production environment, 22-2
  URI of task details application, 22-5
human workflow service engines
  configuring, 20-1
  deployed workflows, 21-11
  fault recovery, 22-3
  faults, 21-6, 22-3, 22-6
  instances, 21-6, 21-10
  managing, 22-1
  monitoring, 21-1
  requests and operation statistics, 21-6, 21-8

I

identity service
  customizing the provider, 20-26
  GUID attribute mapping requirements, 20-25
  third-party providers, 20-13
IMAP email server
  configuring, 20-3
IMAP server
  restarting the SOA server after configuring, 20-3
incident packages
  creating, 11-4
  viewing, 11-29
indexes
  recommendation to delete before purging large amounts of data, then add the indexes back in, 9-7
indexing
  LDAP attributes, 20-21
instance and callback message data
  storing in Oracle Coherence distributed cache, 14-11
instance and fault metrics pages
  disabling in the System MBean Browser, 3-9
  disabling the retrieval of instance and fault count metrics, 3-5, 3-6, B-44
  optimizing the loading of, 3-5, 3-6, B-44
  setting a default search criteria to fetch recent instances and faults only within a recent duration, 3-6
instance ID

creating, 1-6, 8-7
displayed in the SOA composite application Instances page, 1-8
for BPEL process service components, 13-15
for service components, 1-8
for SOA composite applications, 1-6
instance names
setting during design time, 7-20
instance states, 3-4
definition, 3-4
filtering by execution state, 8-10, 8-13
filtering by fault state, 8-10, 8-13
filtering by recovery state, 8-10, 8-13
for composites in which one BPEL process is completed and another is faulted, 8-12
meaning of the ? icon, 8-7
of BPEL process service component remain as completed even if the composite instance state becomes stale, 7-6
of service components and SOA composite applications, 8-12
scenarios under which the state is not evaluated, 8-7
instance tracking
impact of disabling audit tracking across multiple composite flows, 7-18
impact on flow traces when tracking is disabled, 7-18
instances
behavior of service component instances after redeployment, 7-8
binding components, 35-1
business rules service engines, 19-5
capturing the composite instance state, 3-4
clicking the refresh icon to view the actual number of total instances, 7-17
decision service components, 19-10
deleting at the SOA Infrastructure level, 8-12, 8-14
deleting from a composite application home page, 8-7, 8-10
deleting large numbers of, 9-4
deleting running, rolled back instances, 8-10, 8-14
deleting with the looped purge in parallel script, 9-7
deleting with the looped purge script, 9-5
displaying created instances immediately even if a constraint is defined, 7-17
distinguishing unit test instances from other instances, 1-6, 7-22
executing the purge scripts, 9-10
finding composite and component instances programmatically, 10-4
human workflow service components, 21-1, 21-5
human workflow service engines, 21-6, 21-10
inbound adapters, 30-1
initiating a test instance, 7-17, 8-1
invoking multiple instances, 8-5
keeping current BPM composite instances running after redeployment, 7-6
limitations on selecting the option to continue instances upon redeployment, 7-7
mismatch between composite instances and service component instances, 8-11
monitoring BPEL process service component instances, 13-15
moving long running instances to a different partition, 9-17
orphaned service component instances, 8-8
outbound adapters, 30-7
passing security properties with messages during testing, 8-3
possible reasons for redeployment failure, 7-7, 7-8
purging the instances of a specific SOA composite application, 9-9
scenarios in which the Test button is disabled, 8-2
service components, 1-7
setting the instance name during design time, 7-20
SOA composite applications, 1-6
specific BPEL instance is not being recovered, B-8
specifying the XML payload data during testing, 8-2
states, 3-4
states for purging, 9-8
statistics
Oracle Mediator, 16-6
terminating, 8-10
Test button is disabled when a composite is stopped or retired, 7-17
throughput details, 13-15
WSDL URL that does not contain the revision number is processed by the default composite, 8-2
instant messaging
messages are not actionable, 20-5
interaction specification
definition, 34-4
invocations
direct binding, 3-17
invoke messages
configuring automatic recovery attempts, 12-8
configuring unresolved messages for automatic recovery, 12-4
performing manual message recovery, 14-7
situations in which the Last Modified Date column remains empty, 14-10
invoked decision function, 19-19
invoked decision service, 19-19
IPv4 support, B-55
IPv6 support, B-55

J

JAR files
downloading the deployable JAR file, 7-27
Java EE applications
navigating through, 2-12
JCA Adapters

See Oracle Adapters
JMS sensors
values are not displayed, 13-17

L

large document threshold, 37-2
setting a, 12-2
LDAP attributes
indexing recommendations, 20-21
LDAP tools
seeding users, groups, and application roles, 20-26
LDAP user profile
configuring for Oracle User Messaging Service (UMS), 26-34
lifecycle
of SOA composite applications, 1-19, 7-10
performing bulk lifecycle management tasks in partitions, 7-34
local optimization
calls when a composite is unreachable, 3-20
condition checks for using, 3-17
configuring, 3-17, 7-26
forcing the use of, 3-18
overriding, 3-18
viewing the local optimization setting for policies, 7-26
Locator
entry point for Facade API clients, 10-1, 10-3
log files
accessing, 2-5
configuring, 3-13
decision service components, 19-20
encoding property, 3-15
from all managed servers, B-4
Oracle Business Activity Monitoring (BAM), 24-14
Oracle User Messaging Service (UMS), 27-5
log levels
setting for troubleshooting, B-1
log tracing, B-1
loggers
deployment, B-2
Event Delivery Network (EDN), B-2
human workflow/approval management extensions (AMX)/rules, B-2
Oracle Adapters, B-4
Oracle B2B, B-4
Oracle BPEL Process Manager, B-4
Oracle BPM Suite, B-4
Oracle Business Activity Monitoring (BAM), 23-9
Oracle Mediator, B-3
Oracle User Messaging Service (UMS), 27-6
Oracle Web Services Manager (OWSM), B-4
SOA Infrastructure, B-2
logging in
to Oracle Enterprise Manager Fusion Middleware Control, 2-1
logging out
of Oracle Enterprise Manager Fusion Middleware Control, 2-17
looped purge in parallel script
deleting instances with, 9-7
looped purge script
deleting instances with, 9-5

M

managed servers
starting and stopping, B-53
managing database growth, 9-1
mark cancelled
message recovery action, 14-11
masteAliveThreshold property
definition, 12-7
master nodes
recovering, 12-6
maxMessageRaiseSize property
definition, 12-5, 12-6
MDS connections
managing, 2-7
Mediator
See Oracle Mediator
message recovery
adding an index on the DLV_MESSAGE.ECID column to improve SQL performance when searching messages for a specific ECID message, 14-9
exhausted state, 14-8
filtering the display of instances by whether or not they are recoverable, 8-10, 8-13
in BPEL process service engines, 14-7
in BPMN process service engines, 39-7
managing failed automatic recovery attempts, 14-8
mark cancelled action, 14-11
Message States list not applicable to activity message types, 14-10
obtaining the ECID value for message recovery, 13-4, 14-8
recover action, 14-10
recovering callback messages in resolved and undelivered states, 14-10
recovery message information displayed in the Flow Trace page, 13-3
recovery message information displayed in the SOA Infrastructure home page, 4-2
recovery message information displayed on the SOA composite application home page, 1-4, 6-2
recovery options, 14-10
reset action, 14-11
resynchronizing lost, in-memory, Quartz-scheduled jobs, 14-8
situations in which the Last Modified Date column remains empty, 14-10
stranded callback messages, 14-10
message sources, 24-7
messaging extension driver

client API, 26-12
common properties, 26-10
configuring, 26-9
custom properties, 26-11
driver application archive (EAR), 26-9
extension driver security, 26-11
using, 26-12
migration
of human workflow data from a test to a production environment, 22-9
monitor
role, C-1
Mozilla Firefox
Copy details to clipboard link does not appear, 13-7
MTOM attachments
attaching to outbound web services, 34-1
multiple instances
invoking, 8-5
multiple send addresses
configuring, 20-12

N

nodeReapInterval property
definition, 12-7
nodeReapThreshold property
definition, 12-8
nodes
recovering, 12-6
nonfatal connection
specifying a retry count, 3-7
notification service
pluggable, 20-9
notifications
actionable addresses, 20-4
configuration modes, 20-3
configuring the number of email notification messages, 20-12
configuring the outgoing email address, 20-4
configuring the Reply To address, 20-4
creating, 11-23
human workflow
incoming email, 22-7
outgoing, 22-7
properties, 20-1
prerequisites for configuring, 20-2
restarting the server after IMAP and SMTP email server configuration, 20-3
sender and default sender addresses must match values on the Workflow Notification Properties page, 20-2

O

operator
role, C-1
optSoapShortcut
SOAP optimization is automatically configured in 11g, 3-5
Oracle Adapters
configuring, 29-1
configuring binding components, 34-4
deployed resource adapters, 2-13
endpoint properties, 34-27
inbound adapters, 29-1
outbound adapters, 29-6
faults
inbound adapters, 30-1
outbound adapters, 30-7, 30-8
File Adapter logs, 30-11
in multiple revisions of SOA composite applications that include an inbound adapter are displayed as running, 8-11
instances
inbound adapters, 30-1
outbound adapters, 30-7
monitoring, 30-1
Oracle AQ Adapter properties, 34-4
Oracle Database Adapter properties, 34-5
Oracle File Adapter properties, 34-9
Oracle FTP Adapter properties, 34-14
Oracle JMS Adapter properties, 34-21
Oracle MQ Series Adapter properties, 34-23
Oracle Socket Adapter properties, 34-26
properties
inbound adapters, 30-6
outbound adapters, 30-10
rejected messages
inbound adapters, 30-2, 30-3
Oracle AQ Adapter
properties, 34-4
Oracle B2B
accessing from Oracle Enterprise Manager Fusion Middleware Control, 32-3
bindings, 32-1
configuring, 31-1
database tables are not deleted by the Oracle SOA Suite purge scripts, 9-4
foreign key constraints must be disabled before dropping the partition, 9-17
infrastructure monitoring, 32-1
metrics, 31-1
monitoring, 32-1
server properties, 31-1
Oracle BAM adapter
configuring, 23-13
configuring HTTPS, 23-19
configuring to retry sending messages, 23-16
Oracle BPM Worklist
configuring with Oracle HTTP Server, 5-2
globally disabling automatic release timers, 20-11
performing fault recovery from, 1-17
performing human workflow service component fault recovery, 8-15
Oracle Business Activity Monitoring (BAM)
adding members to application roles, 25-6
adding users in Oracle BAM Administrator, 25-11
application roles, 25-9

availability management, 25-1
calling web services, 23-27
configuration property files, 23-31
configuration property reference, 23-31
configuring, 23-1
configuring advanced properties, 23-29
configuring credential mapping, 23-21, 23-26
configuring distribution lists, 23-11
configuring Oracle BAM adapter, 23-13
configuring Oracle BAM batching properties, 23-23
configuring Oracle BAM connection factories, 23-16
configuring Oracle Data Integrator integration properties, 23-8
configuring Oracle User Messaging Service (UMS), 23-9
configuring server basic properties, 23-5
configuring SSL, 23-26
configuring the application URL, 23-4, 23-7
configuring the data source JNDI, 23-7
configuring the embedded LDAP server, 25-10
configuring the logger, 23-9
configuring the outbound email account, 23-8
configuring the report cache persistence manager, 23-8
configuring the report loading indicator, 23-4
configuring the server name, 23-4
configuring trusted domains, 23-21
configuring user permissions, 23-26
configuring viewset sharing, 23-7
configuring web applications properties, 23-3
configuring web basic properties, 23-2
defining users and groups, 25-4
managing, 25-1
managing object ownership, 25-14
monitoring, 24-1
monitoring logs, 24-14
monitoring open connections, 24-11
monitoring Oracle BAM active data cache, 24-2
monitoring performance, 24-12
monitoring the client requests in progress, 24-9
monitoring the enterprise message sources, 24-7
monitoring the event engine component, 24-4
monitoring the report cache component, 24-6
monitoring the report server, 24-10
monitoring web applications, 24-9
monitoring web services, 24-11
removing invalid users from Oracle BAM Administrator, 25-15
securing, 23-26
securing JMS resources, 23-27
user management, 25-2
using Oracle Internet Directory, 23-27
using previously seeded group members, 25-5
Oracle Business Process Management Suite
administration, 1-12
configuration, 1-13
definition, 1-11
management, 1-15
monitoring, 1-15
service components, 1-7
Oracle Coherence
definition, 14-12
Oracle Coherence distributed cache
architecture, 14-12
availability requires use of Oracle Exalogic, 14-12
configuring in the System MBean Browser, 14-14
configuring the storage of multiple audit trail messages in one transaction, 14-14
performance recommendations, 14-13
starting the data grid nodes, 14-15
storing instance and callback message data, 14-11
Oracle Data Integrator
configuring with Oracle Business Activity Monitoring (BAM), 23-8
Oracle Database Adapter
properties, 34-5
Oracle Enterprise Manager Fusion Middleware Control
accessing Oracle B2B, 32-3
deploying drivers, 28-2
logging in, 2-1
logging out, 2-17
Oracle Exalogic
definition, 14-12
Oracle Exalogic platforms
setting the AsynchAuditBatchSize property, 12-3
storing instance and callback message data in Oracle Coherence distributed cache, 14-11
Oracle File Adapter
properties, 34-9, 34-14
Oracle Fusion Middleware, 1-1
Oracle Fusion Middleware Configuration Wizard
deploying drivers, 28-4
Oracle HTTP Server
configuring Oracle SOA Suite and Oracle HTTP Server for SSL communication, 5-9
configuring with Oracle BPM Worklist, 5-2
Oracle Internet Directory
using with Oracle Business Activity Monitoring (BAM), 23-27
Oracle JMS Adapter
properties, 34-21
Oracle Mediator service components
configuring, 15-1
definition, 1-7
fault recovery, 1-17
fault recovery examples, 8-24
managing, 17-1
monitoring, 16-1
resequenced message purge states for, 9-8
setting the instance name during design time, 7-20
Oracle Mediator service engines
managing, 17-1
monitoring, 16-1
Oracle MQ Series Adapter
properties, 34-23
Oracle Service Bus (OSB)

invoking, 10-2
Oracle Service Registry
changing the endpoint reference and service key, 34-29
configuring caching of WSDL URLs, 34-30
configuring the publishing of web services to UDDI, 36-5
inquiry URL, 3-4
properties, 3-4
publishing to nondefault businesses, 36-7
publishing web services to the UDDI registry, 36-4
Oracle Single Sign-On (OSSO)
configuring, 5-2
Oracle SOA Composer
providing a nonadministrator with access, 20-28
Oracle SOA Suite
administration, 1-12, 2-1
configuration, 1-13
definition, 1-1
management, 1-15
monitoring, 1-15
Oracle Socket Adapter
properties, 34-26
Oracle SSL ciphers, 3-7
Oracle User Messaging Service (UMS)
accessing configuration pages, 26-5
adding or removing user messaging preferences business terms, 26-5
architecture, 26-3
client aliases, 26-38
components, 26-2
configuring, 26-1
configuring the email driver, 26-15
configuring the messaging extension driver, 26-9
configuring the proxy driver, 26-32
configuring the SMPP driver, 26-19
configuring the VoiceXML driver, 26-27
configuring the worklist driver, 26-29
configuring the XMPP driver, 26-24
configuring to send and receive notifications, 20-2
configuring to the LDAP user profile, 26-34
definition, 26-1
deregistering messaging client applications, 27-4
driver configuration, 26-6
driver properties, 26-8
enabling client security, 26-37
enabling security, 26-37
keystore configuration, 26-38
logging configuration, 27-6
managing, 28-1
message status, 27-3
metrics and statistics, 27-7
monitoring, 27-1
monitoring drivers using the All tab, 27-4
securing, 26-36
securing JMS resources, 26-39
securing passwords, 26-9
sender and default sender addresses must match values on the Workflow Notification Properties page, 20-2
setting the storage method, 26-5
troubleshooting, 26-40
UMS client applications, 26-2
UMS drivers, 26-2
UMS server, 26-2
viewing log files, 27-5
web service security on notification, 26-37
Oracle Virtual Assembly Builder
description, 1-16
Oracle SOA Suite support, 1-16
Oracle wallet password, 3-7
Oracle Web Service Manager faults, 1-17
Oracle WebLogic Server
starting and stopping a managed server on which the SOA Infrastructure is deployed in the middle of BPEL processing, 7-19
Oracle WebLogic Server Administration Console
accessing, 2-14
administering security of human workflow users, 2-16
configuring adapter connections for composite references, 2-16
creating and managing data sources of adapters, 2-16
creating and managing JMS resources of adapters, 2-16
deploying human task user interfaces, 2-16
managing data sources of the SOA Infrastructure, 2-16
managing JTA settings, 2-16
performing Oracle SOA Suite administrative tasks from, 2-16
oracle.soa.local.optimization.force property
using, 3-18
OracleSystemUser, 25-4
oracle.webservices.local.optimization property
using, 3-18
ora-human-intervention action
fault policy definitions, 8-17, 8-27, 8-29
orphaned service component instances
deleting, 8-8
for which no SOA composite application instance has been created, 8-12
generating, 8-8
OVAB
See Oracle Virtual Assembly Builder
OWSM
See Oracle Web Services Manager

P

partial partitioning
restricting partitioning to tables with high growth rates, 9-13
partitioning
all tables of a service component and service engine, 9-12
always partition the COMPOSITE_INSTANCE

table, 9-13
component tables, 9-2, 9-11, 9-12, 9-13
configuring partitions, 9-12
constraints, 9-12
developing a methodology, 9-3
executing the verification script, 9-16
groups, 9-13
limitations on, 1-21
moving long running instances to a different partition, 9-17
no tables of a service component and service engine, 9-12
Oracle B2B foreign key constraints must be disabled before dropping the partition, 9-17
partial partitioning of components, 9-19
range partition keys, 9-13
restricting partitioning to tables with high growth rates, 9-13
running the purge scripts before performing partitioning, 9-12
selecting the schema partition keys, 9-3
verification script, 9-12
verifying and dropping partitions, 9-16
partitions
accessing the Undeploy SOA Composite wizard, 7-10
activating all composites in a partition, 7-35
API management of, 10-1, 10-3
bulk lifecycle management tasks, 1-21, 7-29, 7-34
cannot rename an existing partition, 7-32
cannot restrict login access to, 7-29
cannot start, stop, activate, or retire, 7-31
cannot transfer deployed composites to a different partition, 7-32
composites using the same inbound resource deployed to different partitions, 7-30
creating, 7-31
creating with ant and WLST, 7-34
default partition, 1-21, 7-3, 7-29
definition, 1-2, 1-15, 1-21, 7-29
deleting, 7-30, 7-31
deleting and undeploying all composites in the partition, 2-12, 7-10
deploying a composite to a partition, 7-35
invalid names for, 7-32
issues with deploying the same composite with a human workflow into multiple partitions, 7-4
limitations on using the socket adapter, 7-30
managing, 7-29
minimum of one required for deployment, 7-3, 7-33
naming conventions, 7-32
not associated with a state, 1-22, 7-31
recreating deployed composites in another partition, 7-33
retiring all composites in a partition, 7-35
selecting during deployment, 7-3
shutting down all composites in a partition, 7-35
starting all composites in a partition, 7-35
top level entry point for partition management APIs, 10-1, 10-3
undeploying all composites in a partition, 7-9, 7-15, 7-36
valid names for, 7-32
passwords
changing the default password in the embedded LDAP server, 20-27
payload validation
BPEL process service engine, 12-3, 37-3
fault generation for invalid payloads, 7-19
for synchronous services, 7-19
SOA composite applications, 7-19
SOA Infrastructure, 3-4
payloads
resolving large payload errors, B-49
saving, 8-6
pluggable notification service, 20-9
registration, 20-10
policies
binding components, 36-1
BPEL process service components, 14-4
BPMN process service components, 39-4
configuring security policies for human workflow web services, 20-29
definition, 1-18
details about which policies to use, 7-23
execution of, 1-19
faults, 1-17
human workflow service components, 22-1
in SOA composite applications, 7-23
limitation on policy attachment errors for service binding components, 36-3
limitation on policy attachments for binding components, 36-1
local attachments, 7-26
location of errors in services and references, 36-3
multiple servers with the same policy, 36-4
no support for policies on event subscriptions, 33-4
Oracle Mediator, 17-4
overriding policy configuration property values, 36-4
policy sets, 36-2
recommendation not to use oracle/wss10_saml_token_service_policy in a production environment with human tasks, 22-2
supported categories, 1-18
policy sets
definition, 20-29, 36-2
popped ruleset, 19-18
ports
changing the SOA Infrastructure server URL port, 3-12
privileges
required for performing specific tasks, 1-12, C-1
process flow
for BPEL 2.0 projects, 13-10
monitoring a BPEL process service component, 13-1

programmatic management of SOA composite applications, 10-1
properties
actionable addresses, 20-4
actionable email account configuring, 20-6
audit level of BPEL process service engine, 12-2
audit level of BPMN process service engine, 37-2
audit level of SOA composite application, 7-18
audit level of SOA Infrastructure, 3-3
audit trail threshold, 12-2, 37-2
binding components, 34-1
callback server URL, 3-5
capture composite instance state of SOA Infrastructure, 3-4
changing of specific properties with green and red arrow icons requires a restart, 3-2
chunking, 3-7
clusterDbTimeRefresh, 12-7
data sources, 3-7
dispatcher engine threads, 12-3, 37-3
dispatcher invoke threads, 12-3, 37-3
dispatcher system threads, 12-3, 37-3
dynamic assignment and task escalation functions, 20-6
for local optimization, 7-26
heartBeatInterval, 12-7
inbound adapters, 30-6
large document threshold, 12-2, 37-2
masteAliveThreshold, 12-7
maxMessageRaiseSize, 12-5, 12-6
nodeReapInterval, 12-7
nodeReapThreshold, 12-8
notification properties, 20-3
Oracle Service Registry, 3-4
Oracle SSL ciphers, 3-7
Oracle wallet password, 3-7
order of precedence, 3-1
order of precedence of execution, 1-13, 7-18
outbound adapters, 30-10
payload validation in SOA composite applications, 7-19
payload validation in SOA Infrastructure, 3-4
payload validation in the BPEL process service engine, 12-3, 37-3
qualityOfService, 14-13
server URL, 3-5
SOA Infrastructure, 3-1
startupRecoveryDuration, 12-6
startWindowTime, 12-5
stopWindowTime, 12-5
subsequentTriggerDelay, 12-5, 12-6
threshHoldTimeInMinutes, 12-5
UDDI Registry, 3-4
web service binding, 3-7
proxy driver
client API, 26-34
common properties, 26-33
configuring, 26-32
custom properties, 26-33
proxy server
best practices for setting proxy properties, B-55
specifying, B-55
purging
deleting indexes before purging large amounts of data, then adding the indexes back in, 9-7
deleting instances with the looped purge in parallel script, 9-7
deleting instances with the looped purge script, 9-5
deleting large numbers of instances with the purge scripts, 9-4
deleting orphaned instances, 8-8
developing a methodology, 9-3
executing the purge scripts, 9-10
instances, 9-4
instances states for purging, 9-8
purge script capabilities, 9-4
purge scripts supported on different Oracle SOA Suite releases, 9-5
purge scripts to run when upgrading, 9-5
resequenced message purge states for Oracle Mediator, 9-8
scripts are only supported on Oracle databases, 9-5
the instances of a specific SOA composite application, 9-9
when to use the Delete With Options dialog or the purge scripts, 8-10, 8-14
pushed ruleset, 19-18

Q

QualityOfService property
enabling or disabling Oracle Coherence cache, 12-4

R

recover
message recovery action, 14-10
recoverable activities
definition, 14-8
recovery
of master nodes, 12-6
recovery actions
abort, 8-17, 8-28, 13-9, 14-3, 14-7, 17-3, 17-7
continue, 8-17, 8-29, 13-9, 14-3, 14-7, 17-3, 17-7
replay, 8-17, 8-28, 13-9, 14-3, 14-7, 17-3, 17-7
rethrow, 8-17, 8-29, 13-9, 14-3, 14-7, 17-3, 17-7
retry, 8-17, 8-28, 13-9, 14-3, 14-7, 17-3, 17-6
redeployment
behavior of BPMN instances after redeployment, 7-7
behavior of service component instances after redeployment, 7-8
from SSO-enabled Oracle Enterprise Manager Fusion Middleware Control, 7-2
guidelines, 7-5
keeping current BPM composite instances running

after redeployment, 7-6
limitations on selecting the option to continue instances upon redeployment, 7-7
possible reasons for failure, 7-7, 7-8
redeploying multiple composites at once is not supported, 7-5
SOA composite applications, 7-5
references
definition, 1-9, 4-6
displaying the endpoint addresses of, 7-19
displaying the WSDL files of, 7-19
location of policy errors, 36-3
monitoring, 4-7
Refresh Alarm Table button
resynchronizing lost jobs, 14-8
refresh icon
location, 7-12
rejected messages
binding components, 35-3
definition, 1-17
deleting at the SOA Infrastructure level, 8-30
deleting from the application home page, 8-32
inbound adapters, 30-2, 30-3
when to use the Delete Rejected Messages option, 8-31, 8-33
replay
recovery action, 8-17, 8-28, 13-9, 14-3, 14-7, 17-3, 17-7
reports engine
monitoring, 24-10
request breakdown
statistics in Oracle Mediator, 16-14
requests and operation statistics in human workflow service engines, 21-6, 21-8
resequencer tables
purging, 9-8
reset
message recovery action, 14-11
RESTful web service
security options in the Test Web Service page, 8-4
testing in the Test Web Service page, 8-3
rethrow
recovery action, 8-17, 8-29, 13-9, 14-3, 14-7, 17-3, 17-7
retiring
a default SOA composite application revision, 7-13
a SOA composite application revision, 7-13, 7-17
impact of retiring a default SOA composite application revision, 7-13, 7-14
impact of undeploying a default SOA composite application revision, 7-14, 7-15
retry
recovery action, 8-17, 8-28, 13-9, 14-3, 14-7, 17-3, 17-6
revisions
definition, 1-19
distinguishing the default revision from other revisions, 1-20
undeployment of default composite revision, 7-9
undeployment of default composite revision when no other active, available revisions exist, 7-9
version added to the application name, 1-20
roles
administrator, C-1
audit trail access, C-7
BPEL process service component home page access, C-6
BPEL process service engine access, C-4
business events page access, C-8
business rules service engine access, C-5
composite home page access, C-3
creating, C-1
decision service component home page access, C-7
flow trace page access, C-7
human task service component home page access, C-6
human workflow service engine access, C-5
matrix of role functionality, C-1
monitor, C-1
operator, C-1
Oracle B2B pages access, C-8
Oracle Mediator service component home page access, C-6
Oracle Mediator service engine access, C-4
references home page access, C-8
required for performing specific tasks, 1-12, C-1
services home page access, C-7
SOA Composite menu access, C-3
SOA Infrastructure menu access, C-3
SOA Infrastructure page access, C-2
System MBean Browser access, C-9
routing statistics in Oracle Mediator, 16-8
RPC/literal-style WSDL files
specifying, 8-7
rule execution tracing
See business rule execution tracing
rules
See business rules

S

SAML message-protected policy configuration
setting up for the SOA Infrastructure, 5-2
search criteria
setting a default search criteria to fetch recent instances and faults only within a recent duration, 3-6
Secure Socket Layer
See SSL
securing
Oracle Business Activity Monitoring (BAM), 23-26
Oracle User Messaging Service (UMS), 26-36
security
application policies, 2-6
application roles, 2-6
automatically authenticating Oracle BPM Worklist users in SAML SSO environments, 5-4

automatically authenticating Oracle BPM Worklist users in Windows Native authentication environments, 5-5
automatically authenticating Oracle Business Process Management Process Composer users in Windows Native authentication environments, 5-5
available documentation, 5-1
configuring for human workflow WSDL files, 5-12
configuring for two-way SSL communication, 5-6
configuring Oracle HTTP Server with Oracle BPM Worklist, 5-2
configuring Oracle Single Sign-On (OSSO), 5-2
configuring Oracle SOA Suite and Oracle HTTP Server for SSL communication, 5-9
configuring SSL between SOA composite application instances and Oracle WebCache, 5-11
developing new security providers for use with Oracle WebLogic Server, 5-2
enabling an asynchronous process deployed to an SSL-enabled, managed server to invoke another asynchronous process over HTTP, 5-11
invoking references in one-way SSL environments in Oracle JDeveloper, 5-9
listing Oracle Internet Directory as the first authentication provider, 5-5
passing security properties with messages during testing, 8-3
policies in SOA composite applications, 7-23
procedures unique to Oracle SOA Suite, 5-1
programming security for Oracle WebLogic Server, 5-2
recommendation to configure either all or no managed servers with SSL, 5-6
securing an Oracle WebLogic Server production environment, 5-2
securing and administering web services, 5-2
securing Oracle WebLogic Server, 5-2
securing web services for Oracle WebLogic Server, 5-2
setting up SAML message-protected policy configuration for the SOA Infrastructure, 5-2
switching from non-SSL to SSL configurations with Oracle BPM Worklist, 5-6
understanding Oracle WebLogic Server security, 5-2
using a custom trust store for one-way SSL during design time, 5-11
seeding
users, groups, and application roles with LDAP tools, 20-26
sensors, 13-2, 38-2
behavior of activity sensors in BPEL 2.0 compensate and compensateScope activities, 13-17
business monitors, 7-36
disabling BPEL sensors at the service engine level, 12-3
disabling BPEL sensors at the SOA composite application level, 7-19
displaying the current state, 7-19
JMS sensor values are not displayed, 13-17
monitoring, 13-16
server URL
property, 3-5
property change requires a restart, 3-5
ServerManager
top level entry point for partition management APIs, 10-1, 10-3
service and reference binding components
See binding components
service components
BPEL processes, 13-1, 14-1
BPMN processes, 38-1
definition, 1-7
human workflow, 20-1, 21-1, 22-1
instance IDs, 1-7
instances, 1-7
Oracle Mediator, 15-1
orphaned instances, 8-8
service engines
BPEL processes, 13-1, 14-1
BPMN processes, 38-1
definition, 1-9
human workflow, 20-1, 21-1, 22-1
monitoring at the SOA Infrastructure level, 4-4
monitoring the average request processing times, 4-5
no management pages for the spring service component, 1-8
Oracle Mediator, 15-1
service infrastructure
definition, 1-11
monitoring the average request processing times, 4-5
service key
changing for Oracle Service Registry, 34-29
services
definition, 1-9, 4-6
location of policy errors, 36-3
monitoring, 4-7, 36-7
Set Time Threshold link
definition, 4-3, 6-3
Show Details link
definition, 4-3, 6-3
shutting down
a composite revision, 7-13, 7-17
SMPP driver
client API, 26-23
common properties, 26-20
configuring, 26-19
custom properties, 26-21
interoperability, 26-19
SMTP email server
configuring, 20-3
SMTP server
restarting the SOA server after configuring, 20-3

SOA composite applications
activating a revision, 7-13, 7-17
application states after SOA Infrastructure shutdown, 3-10
audit level, 7-18
configuring for two-way SSL communication, 5-6
configuring Oracle SOA Suite and Oracle HTTP Server for SSL communication, 5-9
contents, 1-11
definition, 1-4
deleting instances, 8-7, 8-12
deployed states, 7-10
deploying, 7-1
deploying a revision, 7-14
fault recovery, 8-27
faults, 8-27
filtering by instance states, 8-10, 8-13
finding the composite and component programmatically, 10-4
impact of retiring a default revision, 7-13, 7-14
impact of undeploying a default revision, 7-14, 7-15
in which one BPEL process is completed and another is faulted, 8-12
instances, 1-6
lifecycle, 1-19
managing, 8-1
managing the application state at the SOA Infrastructure level, 7-11
managing the application state from the SOA composite application home page, 7-15
message recovery information displayed on the home page, 1-4, 6-2
monitoring, 6-1
monitoring faults, 6-1
monitoring instances, 6-1
packaging, 7-1
payload validation, 7-19
policies, 7-23
programmatically managing, 10-1
purging instances of, 9-9
redeploying, 7-5
redeploying a revision, 7-15
retiring a default revision, 7-13
retiring a revision, 7-13, 7-17
retrieving the composite state programmatically, 10-4
securing, 5-1
setting the default revision, 7-14
setting the instance name during design time, 7-20
showing the composite XML definition, 7-19
showing the WSDL and endpoint URI, 7-19
shutting down a revision, 7-13, 7-17
starting up a revision, 7-13, 7-17
Test button is disabled when a composite is stopped or retired, 7-17
test instances, 8-1
testing, 1-21, 7-17, 7-20
undeploying, 7-8
undeploying a revision, 7-15
undeploying the default composite revision, 7-9
SOA Composite menu
description of options, 2-10
navigating through, 2-8
SOA Governance
definition, 1-20
SOA Infrastructure
audit level, 3-3
cannot change the Infrastructure port from Oracle WebLogic Server Administration Console, 3-12
changing ports, 3-12
configuring, 3-1
during startup, may not be completely initialized to administer incoming requests, 4-2, 7-31
fault recovery, 8-15
faults, 8-15
managing the application state, 7-11
may not completely initialize to administer incoming requests until all deployed composites are loaded, 3-10
message recovery information displayed in the home page, 4-2
monitoring, 4-1
payload validation, 3-4
processing requests, 4-5
properties, 3-1
recent instances and faults, 4-1
restarting the SOA Infrastructure does not activate endpoints when a retired composite is activated, 3-11
starting and stopping, 3-9
startup failure when cwallet.sso includes the SOA map, 3-11
state of composites after SOA Infrastructure shutdown, 3-10
waiting for startup initialization to complete, 3-10, 7-17
SOA Infrastructure application
definition, 1-2
SOA Infrastructure menu
description of options, 2-4
navigating through, 2-2
SOA Partition menu
navigating through, 2-11
sources
message, 24-7
spring service components
definition, 1-7
no spring service engine management pages, 1-11
scope of support in Oracle Enterprise Manager, 1-8
SSL
configuring for two-way SSL communication, 5-6
configuring Oracle SOA Suite and Oracle HTTP Server for SSL communication, 5-9
configuring SOA composite applications for two-way SSL communication, 5-6
configuring SSL between SOA composite

application instances and Oracle WebCache, 5-11
configuring with Oracle Business Activity Monitoring (BAM), 23-26
enabling an asynchronous process deployed to an SSL-enabled, managed server to invoke another asynchronous process over HTTP, 5-11
invoking references in one-way SSL environments in Oracle JDeveloper, 5-9
key alias, 5-8
recommendation to configure either all or no managed servers with SSL, 5-6
switching from non-SSL to SSL configurations with Oracle BPM Worklist, 5-6
using a custom trust store for one-way SSL during design time, 5-11
Start Time From field
setting a default search criteria to fetch recent instances and faults only within a recent duration, 3-6
starting
a managed Oracle WebLogic Server on which the SOA Infrastructure is deployed in the middle of BPEL processing, 7-19
starting up
a composite revision, 7-13, 7-17
startupRecoveryDuration property
definition, 12-6
startWindowTime property
definition, 12-5
stopping
a managed Oracle WebLogic Server on which the SOA Infrastructure is deployed in the middle of BPEL processing, 7-19
stopWindowTime property
definition, 12-5
stress testing
invoking multiple instances, 8-5
subsequentTriggerDelay property
definition, 12-5, 12-6
synchronous services
payload validation, 7-19
system dispatcher messages
definition, 12-3
system faults, 1-17
System MBean Browser
ASNSDriverIMAddress property, 20-4
ASNSDrivers property, 20-13
AsynchAuditBatchSize, 12-3, 14-14
AuditConfig property, 3-8, 4-3, 6-2, 6-3, 8-10, 8-13, 13-4
BpelcClasspath property, 12-4
bpel.config.auditlevel property, 12-8
CustomNSDriverPropertyNames property, 20-4, 20-10
DisableAsserts, 12-4
DisableSensors, 12-4
disabling instance and fault count metrics retrieval, 3-9
ExpirationMaxRetry property, 12-4
ExpirationRetryDelay property, 12-4
FaxCoverPageCount property, 20-4
GlobalTxMaxRetry property, 3-8
GlobalTxRetryInterval property, 3-8
HttpProxyAuthRealm property, 3-8
HttpProxyAuthType property, 3-8
HttpProxyHost property, 3-8
HttpProxyPassword property, 3-8
HttpProxyPort property, 3-8
HttpProxyUsername property, 3-8
HttpServerURL property, 3-8
HttpsServerURL property, 3-9
InstanceKeyBlockSize property, 12-4
instructions on using, 2-7
invoking custom MBeans with WLST, 2-7
KeystoreLocation property, 3-9, 5-7
MaximumNumberOfInvokeMessagesInCache property, 12-4
MaxRecoveryAttempt property, 12-4
navigational path, 3-2
OneWayDeliveryPolicy property, 12-4
QualityOfService, 12-4
QualityofService, 14-14
RetryNotificationMessageThrottle property, 20-4
ServerURL property, 3-12
SOA Infrastructure advanced configuration properties, 3-8
StatsLastN property, 12-4
SyncMaxWaitTime property, 12-4
UddiCacheLifetime property, 3-9, 34-30

T

tablespaces
extending to avoid problems at runtime, B-49
task auto release
configuring, 20-8
portal realm mapping configuration, 20-8
pushback assignee, 20-8
task auto release configuration, 20-9
worklist application URL configuration, 20-8
task escalations, 20-6
task flows
deployment, 7-4
task forms
configuring Oracle HTTP Server for task form attachments, 20-9
task service
configuring the workflow custom classpath URL, 20-6
configuring the workflow service session timeout, 20-6
task service properties
human workflow, 20-5
test cases
contents of, 7-21
definition, 1-21
test runs
definition, 7-21

test suite
assertions, 1-21
definition, 1-21
emulations, 1-21
testing
automating the testing of SOA composite applications, 7-20
initiating a test instance, 8-1
passing security properties with messages, 8-3
restrictions on displaying payload data in the Test Web Service page, 8-6
restrictions on stress tool of Test Web Service page, 8-5
scenarios in which the Test button is disabled, 7-17, 8-2
specifying the XML payload data, 8-2
Test button is disabled when a composite is stopped or retired, 7-17
WSDL URL that does not contain the revision number is processed by the default composite, 8-2
third party application servers
instructions for using this guide, 1-13
thread dumps from all managed servers, B-4
threshHoldTimeInMinutes property
definition, 12-5
throughput
instance details, 13-15
timeouts
resolving, B-5
setting, 12-4
TopLink sessions
managing cache, 2-7
transaction timeouts
setting, 12-4
transient processes
definition, B-42
troubleshooting
additional documentation for, B-56
AMX extension issues, B-29
application transaction does not complete and the underlying composite is stuck in a running state, B-10
asynchronous transactions return with an error after a long time, B-7
automatic recovery of BPEL instances is not recovering a specific instance, B-8
best practices for starting and stopping a managed server, B-53
business events, B-40
design time at runtime issues, B-25
event delivery network, B-40
extending tablespaces, B-49
flow diagram display issues on lower end hosts, B-55
human workflow API usage issues, B-26
human workflow issues, B-13
identity service issues, B-38
increasing database connection values, B-6
limitations on Safari browser use, B-55
long running, synchronous calls to remote web services error out, B-7
notification issues, B-15
Oracle BPM Worklist/task region issues, B-30
Oracle JDeveloper data control/form generation issues, B-26
Oracle User Messaging Service (UMS), 26-40
resolving connection timeouts, B-5
resolving database growth caused by high transaction volume, B-50
resolving exception errors caused by large payloads, B-49
resolving message failure caused by too many open files, B-48
runtime diagnostics, B-8
runtime failure with a "Cannot read WSDL" error, B-8
service/System MBean Browser issues, B-27
setting log levels, B-1
slow application performance, B-51
SOA server startup problems, B-54
some composites are retried multiple tines on failure, B-10
task action issues, B-14
task assignment, routing, and escalation issues, B-13
task attribute mapping issues, B-18
task comments/attachment issues, B-24
task form/action issues, B-22
task history issues, B-20
task report issues, B-19
task view issues, B-17
test-to-production issues, B-34
updating EJB transaction timeouts, B-6

U

UDDI Registry
configuring the publishing of web services to, 36-5
inquiry URL, 3-4
properties, 3-4
publishing to, 36-4
publishing web services to, 36-7
UMS
See Oracle User Messaging Service
undeploying
a composite revision, 7-15
undeployment
deleting a partition and all its composites, 7-10
from SSO-enabled Oracle Enterprise Manager Fusion Middleware Control, 7-2
guidelines, 7-8
of all composites in a partition, 7-9
of an application instance does not cause the instance to complete, 13-23, 16-13
of default composite revision, 7-9
of default composite revision when no other active, available revisions exist, 7-9, 7-15
SOA composite applications, 7-8

unit tests
assertions, 1-21
distinguishing unit test instances from other instances, 1-6, 7-22
emulations, 1-21
of SOA composite applications, 1-21
running, 7-20
test cases, 1-21
test suites, 1-21
users
demo community, A-1
demo user community, A-1
seeding with LDAP tools, 20-26

V

verification script
executing, 9-16
for partitioning, 9-12
not provided for business rules, 9-16
VoiceXML driver
client API, 26-29
common properties, 26-27
configuring, 26-27
custom properties, 26-28
interoperability, 26-27

W

watches
creating, 11-23
web browsers
Copy details to clipboard link does not appear with Mozilla Firefox, 13-7
limitation on using the Safari Browser to view WSDL file content, B-55
supported, 2-1
web service binding properties, 3-7
web services
configuring, 34-3
WebLogic Diagnostic Framework
See WLDF
WLDF
definition, 11-2
WLST
creating partitions, 7-34
deploying drivers, 28-2
deployment, 7-5
invoking custom MBeans, 2-7
workflow custom classpath URL
configuring, 20-6
workflow service session timeout
configuring, 20-6
worklist driver
client API, 26-32
common properties, 26-31
configuring, 26-29
custom properties, 26-32
installing, 26-30
installing on the IBM WebSphere platform, 26-31
wsadmin
deploying drivers, 28-5
WSDL files
displaying the WSDL files of all external services, 7-19
human workflow, 5-12
specifying during instance creation, 8-2
specifying RPC/literal style WSDL files, 8-7
specifying the default revision, 8-3
WSDL URL that does not contain the revision number is processed by the default composite, 8-2
WS-Reliable Messaging
restrictions, 7-26

X

XML definition
of SOA composite applications, 7-19
XML payload data
specifying during instance creation, 8-2
XMPP driver
application archive (EAR), 26-25
client API, 26-27
common properties, 26-25
configuring, 26-24
custom properties, 26-26
definition, 26-24
interoperability, 26-24, 26-25