

Oracle® Fusion Middleware

Administering Oracle Enterprise Scheduler

12c (12.2.1)

E57088-02

November 2016

Documentation for administrators that describes how to define, schedule and run jobs with Oracle Enterprise Scheduler.

Copyright © 2011, 2016, Oracle and/or its affiliates. All rights reserved.

Primary Author: Oracle Corporation

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, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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 about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	xi
Audience	xi
Documentation Accessibility	xi
Related Documents.....	xi
Conventions.....	xii
 What's New in This Guide.....	 xiii
 1 Introduction to Oracle Enterprise Scheduler	
1.1 Introduction to Administering Oracle Enterprise Scheduler	1-1
1.2 Basic Tasks for Configuring and Managing Oracle Enterprise Scheduler	1-2
1.2.1 Setup and Configuration	1-2
1.2.2 Creating Jobs to Run	1-2
1.2.3 Bundling Jobs and Refining Their Work	1-3
1.2.4 Monitoring the System	1-3
 2 Getting Started Managing Oracle Enterprise Scheduler	
2.1 Logging into Oracle Enterprise Manager Fusion Middleware Control Console	2-1
2.2 Navigating to Oracle Enterprise Scheduler Administration Tasks	2-2
2.3 The Scheduling Service Home Page.....	2-5
2.4 Logging Out of Oracle Enterprise Manager Fusion Middleware Control	2-7
 3 Performing Basic Administration Tasks	
3.1 Introduction to Performing Basic Administration Tasks.....	3-1
3.2 Installing Oracle Enterprise Scheduler	3-2
3.2.1 Targeting Oracle Enterprise Scheduler During Domain Creation.....	3-2
3.2.2 OWSM-PM Targeting With Oracle Enterprise Scheduler.....	3-2
3.3 Configuring Perl to Support Process Jobs.....	3-3
3.4 Configuring Oracle Enterprise Scheduler	3-3
3.4.1 Expanding an Oracle Enterprise Scheduler Cluster.....	3-4
3.4.2 Configuring a Request Processor	3-5

3.4.3	Configuring a Request Dispatcher	3-6
3.5	Searching for Configuration Changes to Oracle Enterprise Scheduler in Cloud Control	3-7
3.6	Starting and Stopping Oracle Enterprise Scheduler Components	3-8
3.6.1	Starting and Stopping an Oracle Enterprise Scheduler Service Instance.....	3-9
3.6.2	Starting and Stopping a Request Processor or Dispatcher	3-9
3.7	Starting and Stopping a Request Processor Dispatcher	3-10
3.8	Managing Application Properties	3-11
3.8.1	Job Location Properties.....	3-12
3.9	Managing Application and Role Policies	3-14
3.9.1	Managing Application Roles	3-14
3.10	Managing Oracle Enterprise Scheduler Web Services	3-15
3.10.1	Securing Web Services	3-15

4 Managing Oracle Enterprise Scheduler Requests

4.1	Introduction to Managing Oracle Enterprise Scheduler Job Requests	4-1
4.2	Managing Oracle Enterprise Scheduler Requests	4-2
4.2.1	Submitting an Oracle Enterprise Scheduler Job Request	4-2
4.2.2	Searching for Oracle Enterprise Scheduler Job Requests	4-4
4.2.3	Creating or Editing Predefined Job Schedules.....	4-9
4.2.4	Viewing Job Request Details.....	4-10
4.2.5	Holding and Resuming Oracle Enterprise Scheduler Job Requests	4-11
4.2.6	Canceling Oracle Enterprise Scheduler Job Requests	4-11
4.3	Managing Purge Policies	4-15
4.3.1	Setting Up a New Purge Policy	4-15
4.3.2	Physically Purging Job Requests from the Database	4-17
4.3.3	Viewing Purge Policies.....	4-18
4.3.4	Updating a Purge Policy	4-18
4.3.5	Deleting a Purge Policy	4-18
4.4	Configuring Simple Data Security for Job Requests.....	4-18
4.4.1	Oracle Simple Data Security Actions.....	4-19
4.4.2	How to Create Data Security Policies for Oracle Enterprise Scheduler Components	4-20

5 Managing the Work of Oracle Enterprise Scheduler Jobs

5.1	Introduction to Managing the Work of Oracle Enterprise Scheduler Jobs	5-1
5.2	Managing Job Metadata.....	5-1
5.2.1	Managing Job Definitions	5-2
5.2.2	Managing Job Sets	5-17
5.2.3	Managing Incompatibilities	5-20
5.2.4	Bulk Customization of Job Metadata.....	5-23
5.3	Managing Work Assignments and Workshifts	5-30
5.3.1	Managing Work Assignments.....	5-30
5.3.2	Managing Workshifts	5-34

5.3.3	Managing Schedules	5-36
5.4	Managing Metadata Security for Jobs.....	5-38
5.4.1	Metadata Security Actions	5-38
5.4.2	How to Create Metadata Policies for Oracle Enterprise Scheduler Resources	5-39
6	Monitoring Oracle Enterprise Scheduler	
6.1	Overview of Monitoring Oracle Enterprise Scheduler	6-1
6.2	Monitoring Oracle Enterprise Scheduler Request Activity	6-1
6.2.1	Viewing Top Ten Long Running Requests and Top Ten Ready Requests	6-2
6.2.2	Viewing Completed Job Requests.....	6-3
6.2.3	Viewing Job Request Response and Load	6-3
6.2.4	Viewing Performance as Processing and Wait Times.....	6-4
6.3	Monitoring the Performance of a Single Oracle Enterprise Scheduler Instance	6-4
6.3.1	Viewing a Summary of Oracle Enterprise Scheduler Activity	6-5
6.3.2	Viewing Current Oracle Enterprise Scheduler Activity	6-5
6.3.3	Viewing Historical Oracle Enterprise Scheduler Reports	6-6
6.3.4	Managing Chart Sets.....	6-7
6.3.5	Selecting the Performance Time Period	6-8
6.4	Monitoring Key Performance Metrics for Oracle Enterprise Scheduler.....	6-9
6.5	Viewing General Oracle Enterprise Scheduler Instance Information.....	6-9
6.6	Viewing Scheduler Component Status.....	6-10
6.7	Managing Logging for Oracle Enterprise Scheduler.....	6-10
6.7.1	Viewing Log Information for the Domain.....	6-11
6.7.2	Viewing Job Request Logs	6-11
6.7.3	Viewing Log Messages for Scheduled Job Requests.....	6-12
6.7.4	Setting Oracle Enterprise Scheduler Log Levels.....	6-12
6.7.5	Tracing Oracle Enterprise Scheduler Jobs	6-13
6.7.6	Saving Job Request Logs	6-15
6.8	Managing Alerts for Oracle Enterprise Scheduler in Cloud Control.....	6-15
6.8.1	Configuring Metric Alerts for Oracle Enterprise Scheduler in Cloud Control	6-17
6.8.2	Viewing Alerts for Oracle Enterprise Scheduler in Cloud Control	6-18
6.9	Auditing Administrative Actions.....	6-19
6.9.1	Getting the List of Auditable Events	6-19
6.9.2	Specifying Which Events Get Logged	6-20
6.9.3	Viewing Event Audit Data.....	6-21
A	High Availability for Oracle Enterprise Scheduler	
A.1	Introduction to High Availability for Oracle Enterprise Scheduler.....	A-1
A.2	Oracle Enterprise Scheduler Concepts	A-2
A.2.1	Oracle Enterprise Scheduler Architecture	A-2
A.2.2	Oracle Enterprise Scheduler Components.....	A-4
A.2.3	Oracle Enterprise Scheduler Life Cycle	A-4
A.2.4	Oracle Enterprise Scheduler Life Cycle Tools.....	A-6

A.3	Configuring High Availability for Oracle Enterprise Scheduler	A-6
A.3.1	Oracle Enterprise Scheduler Configuration and Deployment Artifacts	A-7
A.3.2	Oracle Enterprise Scheduler Logging	A-7
A.3.3	Oracle Enterprise Scheduler Cluster Architecture	A-7
A.3.4	Configuring the Oracle Enterprise Scheduler Front End Host and Port	A-9
A.3.5	Failover Requirements.....	A-10
A.3.6	Scalability.....	A-11
A.3.7	Backup and Recovery	A-11
A.3.8	Load Balancing	A-11
A.4	Managing an Oracle Enterprise Scheduler Cluster	A-12
A.4.1	Starting and Stopping the Cluster	A-12
A.4.2	Propagating Configuration Changes to the Cluster.....	A-12
A.4.3	Deploying Applications to the Cluster	A-12
A.4.4	Failures and Expected Behavior.....	A-13

B Troubleshooting Oracle Enterprise Scheduler

B.1	Introduction to Troubleshooting Oracle Enterprise Scheduler	B-1
B.2	Getting Started with Troubleshooting Oracle Enterprise Scheduler Jobs	B-1
B.2.1	Troubleshooting Asynchronous Scheduled Jobs	B-2
B.2.2	Troubleshooting SOA Composite Jobs.....	B-5
B.2.3	Troubleshooting Process Jobs	B-5
B.2.4	Steps for Manual Recovery.....	B-5
B.2.5	Job Diagnostics	B-7
B.3	Getting Started with Troubleshooting an Oracle Enterprise Scheduler Cluster.....	B-13
B.3.1	Finding Performance and Scalability Issues	B-14
B.3.2	Using a Shared Database	B-14
B.3.3	Tuning Oracle Enterprise Scheduler System Performance.....	B-14
B.4	Problems and Solutions.....	B-17
B.4.1	Job Remains in WAIT State	B-18
B.4.2	Synchronous Job Continues in RUNNING State for Too Long	B-19
B.4.3	Asynchronous Jobs Remain in RUNNING State and Do Not Complete	B-20
B.4.4	Asynchronous Java SOA Job Remains In RUNNING State	B-20
B.4.5	Asynchronous Java Oracle ADF Business Components Job Remains In RUNNING State.....	B-21
B.4.6	Asynchronous PL/SQL Job Remains in RUNNING State	B-22
B.4.7	Job Does Not Execute at Scheduled Time	B-22
B.4.8	Asynchronous Java Job Requires Manual-Error Recovery.....	B-23
B.4.9	Spawned (Process Type) Job Requires Manual Error Recovery	B-23
B.4.10	Job Remains in CANCELLING State	B-24
B.4.11	Performance and Scalability Goes Down When Two Very Database Intensive Jobs Run at the Same Time	B-25
B.4.12	Newly Added Server Is Not Being Utilized or Running Inappropriate Jobs	B-25
B.4.13	Oracle Enterprise Scheduler Run-Time System Is Throwing Errors.....	B-26

B.4.14	Oracle Enterprise Scheduler Is Running Out Of Database Connections.....	B-27
B.4.15	Job Queue Full Due to a Hanging Job.....	B-27
B.5	Using My Oracle Support for Additional Troubleshooting Information	B-28
C	Audit Reference for Oracle Enterprise Scheduler	
C.1	About Custom and Standard Audit Reports	C-1
C.2	Audit Events in Oracle Enterprise Scheduler	C-1

List of Tables

3-1	Job Location Properties.....	3-13
4-1	Executable Requests and Cancellation States.....	4-11
4-2	Pre-processor States and States Following Cancellation Initiation.....	4-12
4-3	Returned Synchronous Java Job States and States Following Cancellation Initiation....	4-13
4-4	Returned Asynchronous Java Job States and States Following Cancellation Initiation.	4-13
4-5	Returned PL/SQL Job States and States Following Cancellation Initiation.....	4-14
4-6	Returned Spawned Job States and States Following Cancellation Initiation.....	4-14
4-7	Grant Actions for Data Security.....	4-19
5-1	System Properties.....	5-6
5-2	exportESSJobMetadataBulk Parameters.....	5-28
5-3	exportESSJobMetadataBulk Exceptions.....	5-28
5-4	customizeJobDefinitionBulk Parameters.....	5-29
5-5	customizeJobDefinitionBulk Exceptions.....	5-29
5-6	Specialization Properties Available for Specialization Rules.....	5-32
5-7	Grant Actions for Metadata Security.....	5-38
5-8	Sample Permission Grants for Security.....	5-39
6-1	Response and Load Statistics.....	6-4
6-2	Performance Statistics.....	6-4
6-3	Completed Request Statistics by Job Name.....	6-6
6-4	Completed Request Statistics by User.....	6-6
6-5	Request Metrics By Work Assignment.....	6-7
6-6	Completed Request Statistics.....	6-7
6-7	Oracle Enterprise Scheduler Key Performance Metrics.....	6-9
6-8	Loggers and log levels for Oracle Enterprise Scheduler.....	6-13
6-9	Oracle Enterprise Scheduler Alerts.....	6-16
6-10	Oracle Enterprise Scheduler Audit Event Categories	6-20
6-11	FilterPreset Levels for Event Auditing.....	6-20
A-1	Oracle Enterprise Scheduler External Component Failover.....	A-11
B-1	Job Request States and Associated Diagnostic Codes.....	B-8
C-1	Oracle Enterprise Scheduler Audit Events.....	C-2

Preface

This document describes how to configure and manage Oracle Enterprise Scheduler.

Oracle Enterprise Scheduler provides the ability to run different job types, including: Java, PL/SQL, and binary scripts, distributed across the nodes in an Oracle WebLogic Server cluster. Oracle Enterprise Scheduler runs these jobs securely, with high availability and scalability, with load balancing and provides monitoring and management through Oracle Enterprise Manager Fusion Middleware Control.

Audience

This document is intended for administrators of Oracle Enterprise Scheduler.

Provide a thorough explanation of the intended audience. You can incorporate brief sentence or two on the tasks, if that helps you define the audience. Generally, the preface should be the central place to define the audience, not in the content of the chapters.

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

Related Documents

For more information, see the following documents in the Oracle 12c Fusion Middleware documentation set:

- *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Scheduling Service*
- *WLST Command Reference for SOA Suite*
- *Installing and Configuring Oracle SOA Suite and Business Process Management*

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.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	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

There are no changes in this guide for Release 12c(12.2.1).

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>.

Note: Screens shown in this guide may differ from your implementation, depending on the skin used. Any differences are superficial.

Introduction to Oracle Enterprise Scheduler

This chapter provides a general introduction to Oracle Enterprise Scheduler and its administration tasks. With Oracle Enterprise Scheduler, you can define, schedule and run jobs, or units of work done on an application's behalf.

Oracle Enterprise Scheduler is installed with a product that embeds it to provide scheduling services. You typically won't have to install Oracle Enterprise Scheduler.

This chapter includes the following sections:

- [Introduction to Administering Oracle Enterprise Scheduler](#)
- [Basic Tasks for Configuring and Managing Oracle Enterprise Scheduler](#)

1.1 Introduction to Administering Oracle Enterprise Scheduler

With Oracle Enterprise Scheduler, you can define, schedule and run jobs. A job is a unit of work done on an application's behalf.

For example, you might define a job that runs a particular PL/SQL function or command-line process. You can define a job that executes Java code. Those job types are included with Oracle Enterprise Scheduler:

- Java job type for executing Java code.
- Web Service job type.
- EJB job type.
- Process job type for executing a command-line command.
- PL/SQL job type for executing functions in Oracle's extension language for SQL.

When you create jobs from these types you associate metadata as a job definition. Through this context-specific metadata, the job does meaningful work on the application's behalf. That metadata can include parameter values the user specifies and values for properties defined by Oracle Enterprise Scheduler. By defining schedules, you can have jobs run when you want them to (such as at off-peak times). You can also specify constraints that prevent jobs from running when you *don't* want them to. By creating job sets, you can group multiple jobs into batches whose jobs run either simultaneously or in a sequence.

Oracle Enterprise Scheduler provides tools with which you can monitor the system. For example, you might want to get a snapshot of current and pending job requests, or see how the system is performing under load. You can also get logs and the status of service components.

You can perform each of these tasks, from submitting jobs to monitoring the system, with the Scheduling Services pages of Oracle Enterprise Manager Fusion Middleware

Control Console. For a more detailed list of tasks, see [Basic Tasks for Configuring and Managing Oracle Enterprise Scheduler](#).

1.2 Basic Tasks for Configuring and Managing Oracle Enterprise Scheduler

You administer Oracle Enterprise Scheduler with the Oracle Enterprise Manager Fusion Middleware Control Console (Fusion Middleware Control).

There, the Scheduling Service home page provides an overview of scheduler component status, top running and completed scheduled job requests, as well as a performance summary of scheduled job requests. Other Scheduling Service pages provide a user interface for performing the tasks described in this section (and many others).

Oracle Enterprise Manager Fusion Middleware Control Console provides accessibility options for the pages on which you monitor and manage Oracle Enterprise Scheduler applications. Fusion Middleware Control supports screen readers and provides standard shortcut keys to support keyboard navigation. You can also view the console pages in high contrast or with large fonts for better readability. For information and instructions on configuring accessibility in Fusion Middleware Control, see "Using Oracle Fusion Middleware Accessibility Options" in *Oracle Fusion Middleware Administrator's Guide*.

The following sections provide a summary of steps you might take to set up the Oracle Enterprise Scheduler.

1.2.1 Setup and Configuration

1. Create a domain and configure a cluster. Oracle Enterprise Scheduler is installed by a product that includes it as a service to manage schedules. However, it is possible that the embedding product didn't configure domains that support Oracle Enterprise Scheduler for deploying. See [Installing Oracle Enterprise Scheduler](#).
2. Configure a request processor and request dispatcher. A job request processor is bound to a particular Oracle Enterprise Scheduler server, and is responsible for allocating threads for job requests. A job request dispatcher polls for job requests. See [Configuring Oracle Enterprise Scheduler](#).
3. Start the instance. See [Starting and Stopping Oracle Enterprise Scheduler Components](#).

1.2.2 Creating Jobs to Run

1. Create job definitions. You can define and manage job definitions that specify values specific to the request at hand. See [Managing Job Definitions](#).
2. Create schedules. You can define a schedule for managing the execution of job requests, work assignments, work shifts and purge policies. You can also create schedules on the fly when creating a request. See [Specifying a Job Request Schedule](#), [Creating or Editing Predefined Job Schedules](#) and [Managing Schedules](#).
3. Create and submit job requests. A job request pulls together a job definition with a schedule on which the job runs. See [Submitting an Oracle Enterprise Scheduler Job Request](#).

4. Create a purge policy. Purge policies allow you to define the conditions for retaining and purging job requests associated with a request processor. See [Managing Purge Policies](#).

1.2.3 Bundling Jobs and Refining Their Work

1. Create job sets. With job sets, you can bundle jobs into groups in which the jobs run at the same time or in a sequence. See [Managing Job Sets](#).
2. Define restrictions with incompatibilities. Incompatibilities specify which jobs shouldn't run at the same time, along with the conditions under which they are incompatible. See [Managing Incompatibilities](#).
3. Define restrictions with work assignments. Work assignments define rules under which jobs can be processed, along with windows of time in which they can run. See [Managing Work Assignments](#).

1.2.4 Monitoring the System

- Monitor the system. You can monitor job requests, system performance, and system status. You can also get logs for job requests and the system. See [Monitoring Oracle Enterprise Scheduler](#).

Getting Started Managing Oracle Enterprise Scheduler

This chapter describes how to access Oracle Enterprise Manager Fusion Middleware Control and Oracle Enterprise Scheduler configuration, monitoring, and management tasks.

Note:

You can also use Oracle WebLogic Scripting Tool to perform server management tasks. For more information, see “Enterprise Scheduler Custom WLST Commands” in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

This chapter includes the following sections:

- [Logging into Oracle Enterprise Manager Fusion Middleware Control Console](#)
- [Navigating to Oracle Enterprise Scheduler Administration Tasks](#)
- [The Scheduling Service Home Page](#)
- [Logging Out of Oracle Enterprise Manager Fusion Middleware Control](#)

2.1 Logging into Oracle Enterprise Manager Fusion Middleware Control Console

You can administer Oracle Enterprise Scheduler by using the Oracle Enterprise Manager Fusion Middleware Control. This section describes how to start and log in to the console.

To log in to Oracle Enterprise Manager Fusion Middleware Control Console:

1. Use Internet Explorer 7, Mozilla Firefox 2.0.0.2, or 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 Console 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 the 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. You can also add new administrator users using the WebLogic console.
- *password* is the password you entered during installation.

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

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

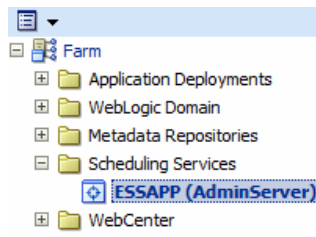
The farm home page is displayed. From there, you can navigate to the Scheduling Service.

2.2 Navigating to Oracle Enterprise Scheduler Administration Tasks

The Scheduling Service menu of Fusion Middleware Control or Cloud Control provides access to administrative tasks for Oracle Enterprise Scheduler.

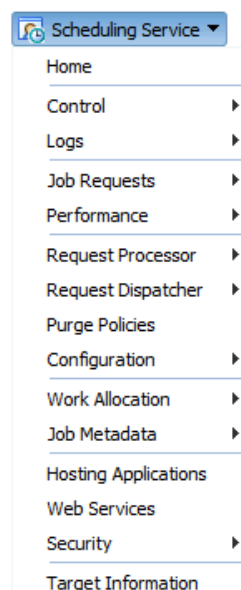
To navigate to Oracle Enterprise Scheduler pages in Fusion Middleware Control or Cloud Control:

1. From the navigation pane, expand the farm and then **Scheduling Services**, and select the Oracle Enterprise Scheduler component.



The Scheduling Service home page displays. See [The Scheduling Service Home Page](#) for further information about the contents of the home page.

2. Select the **Scheduling Service** menu.



The **Scheduling Service** menu displays the following options.

Menu Option	Description
Home	Displays the Scheduling Service home page. For more information about the contents of this page, see The Scheduling Service Home Page .
Control	Displays the following options: • Start Up: This option runs the Oracle Enterprise Scheduler server. For more information, see Starting and Stopping Oracle Enterprise Scheduler Components. • Shut Down: This option shuts down the Oracle Enterprise Scheduler server. For more information, see Starting and Stopping Oracle Enterprise Scheduler Components.
Logs	The View Log Messages option displays the Log Messages page for viewing the contents of event log files. See "Managing Log Files and Diagnostic Data" chapter in <i>Oracle Fusion Middleware Administrator's Guide</i> for more information. The log files are stored in the following directories: • <code>DOMAIN_HOME/servers/SERVER_HOME/logs</code> • <code>DOMAIN_HOME/servers/SERVER_HOME/logs/apps</code> • <code>DOMAIN_HOME/servers/SERVER_HOME/logs/owsm/msglogging/diagnostic.log</code> • <code>DOMAIN_HOME/servers/SERVER_HOME/sysman/log/emoms.log</code>
Job Requests	Displays the following options: • Search Job Requests: Allows you to search for job requests according to particular parameters. For more information, see Searching for Oracle Enterprise Scheduler Job Requests. • Submit Job Requests: Allows you to submit an execution request for a particular job. For more information, see Submitting an Oracle Enterprise Scheduler Job Request. • Define Schedules: This option displays the Schedules page which allows you to create, edit and delete schedules for job requests. For more information, see Creating or Editing Predefined Job Schedules.
Performance	Displays the following options: • Service Summary: This option displays a brief overview of the service performance. For more information, see Viewing a Summary of Oracle Enterprise Scheduler Activity. • Current Activity: This option displays current Oracle Enterprise Scheduler activity. For more information, see Viewing Current Oracle Enterprise Scheduler Activity. • Historical Reports: This option displays a chart or table depicting previous Oracle Enterprise Scheduler data. For more information, see Viewing Historical Oracle Enterprise Scheduler Reports.

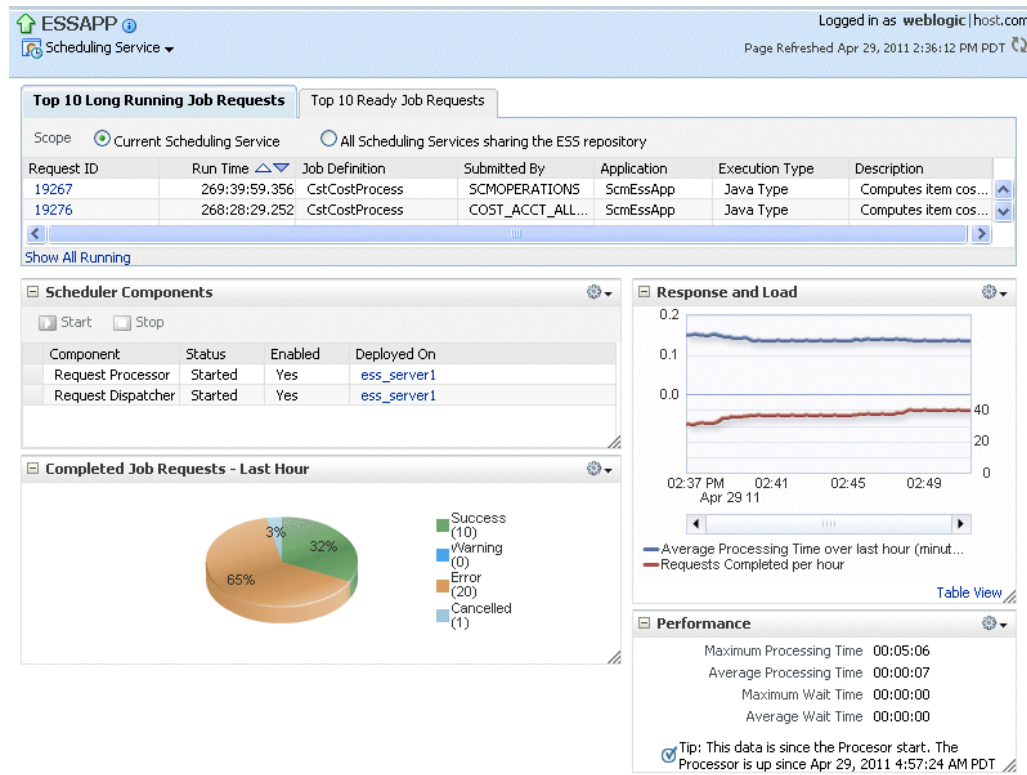
Menu Option	Description
Request Processor	Displays the following options: • Start/Stop: This option causes the request processor to start or stop running. For more information, see Starting and Stopping Oracle Enterprise Scheduler Components. • Configure: This option displays the Configure Request Processor page, which allows you to enable the request processor, create work assignment bindings and so on. For more information, see Configuring Oracle Enterprise Scheduler.
Request Dispatcher	Displays the following options: • Start/Stop: This option causes the request dispatcher to start or stop running. For more information, see Starting and Stopping Oracle Enterprise Scheduler Components. • Configure: This option displays the Configure Request Dispatcher page, which allows you to enable the request dispatcher, manage polling intervals, and so on. For more information, see Configuring Oracle Enterprise Scheduler.
Purge Policies	Allows you to configure and manage job request purge policies including job request purge criteria, retention criteria and purge policy schedules. For more information, see Managing Purge Policies.
Configuration	Opens the Application Properties page that allows you to view, add, modify or delete application properties. For more information, see Managing Application Properties.
Work Allocation	Displays the following options: • Work Assignments: This option displays the Work Assignments page which allows you to create, edit and delete work assignments. For more information, see Managing Work Assignments. • Workshifts: This option displays the Workshifts page which allows you to create, edit and delete workshifts. For more information, see Managing Workshifts. • Schedules: This option displays the Schedules page which allows you to create, edit and delete schedules for work assignments, workshifts and job request purge policies. For more information, see Managing Schedules.
Job Metadata	Displays the following options: • Job Definitions: This option displays the Job Definitions page, which allows you to manage job definitions. See Managing Job Definitions for further information. • Job Sets: This option displays the Job Sets page, which allows you to create and edit job sets for a given application. For more information, see Managing Job Sets. • Incompatibilities: This option displays the Incompatibilities page, which allows you to create and edit incompatibilities for a given application. For more information, see Managing Incompatibilities.
Hosting Applications	Lists hosting applications deployed across all Scheduling Services sharing the Oracle Enterprise Scheduler repository.

Menu Option	Description
Web Services	Displays a page that provides an overview of the web services in the application including web service endpoints and application-level metrics. For more information, see Managing Oracle Enterprise Scheduler Web Services .
Security	Displays the following options: • Application Policies: This option displays the authorization policies that an application relies upon for controlling access to its resources. See Managing Application and Role Policies for further information. • Application Roles: This option displays the roles used by security aware applications specific to the application. See Managing Application and Role Policies for further information.
General Information Target Information	Displays general details about the Oracle Enterprise Scheduler instance. Known as Target Information in Cloud Control. For more information, see Viewing General Oracle Enterprise Scheduler Instance Information .

2.3 The Scheduling Service Home Page

The Scheduling Service home page provides an overview of the performance of Oracle Enterprise Scheduler components and jobs, including component status, the number of completed job requests in the last hour, as well as the processing times for running jobs.

You can use this page as a starting point for monitoring and administering Oracle Enterprise Scheduler. [Figure 2-1](#) shows a portion of the Scheduling Service home page.

Figure 2-1 Scheduling Service Home Page

These pages contain the following regions:

- Top Ten Long Running Requests and Top Ten Ready Job Requests Regions

The Top Ten Long Running Requests region displays the top ten long running scheduled job requests, including request ID, job runtime, job definition used, executing application, job execution type and description. You can set the scope of the top ten long running requests displayed to the current scheduling service only, or all scheduling services sharing the Oracle Enterprise Scheduler repository. For more information, see "Viewing Top Ten Long Running Oracle Enterprise Scheduler Requests" in [Monitoring Oracle Enterprise Scheduler](#).

- Scheduler Components Region

The Scheduler Components region displays the components of Oracle Enterprise Scheduler, including the job request processor and dispatcher. The tab displays the status of each component, the name of the server to which it is deployed and whether or not the component is enabled. You can start and stop each component as required. For more information, see "Viewing Scheduler Component Status" in [Monitoring Oracle Enterprise Scheduler](#).

- Completed Job Requests Region

The Completed Job Requests region displays the status of scheduled jobs completed within the last hour. For more information, see "Viewing Completed Job Requests" in [Monitoring Oracle Enterprise Scheduler](#).

- Response and Load Region

The Response and Load region displays performance monitoring statistics regarding the time required to process job requests. For more information, see "Viewing Job Request and Load" in [Monitoring Oracle Enterprise Scheduler](#).

- Performance Region

The Performance region displays performance data for job requests, such as processing times and wait times. For more information, see "Viewing Performance as Processing and Wait Times" in [Monitoring Oracle Enterprise Scheduler](#).

- Monitoring and Diagnostics Region (Cloud Control only)

The Monitoring and Diagnostics sub-section displays major incidents and changes that have occurred in the instance, including incidents at the instance level, incidents that have occurred in descendant targets, any configuration changes that have occurred over the last seven days and any support workbench problems. Incidents are listed by severity, namely fatal, critical, warning and escalation levels.

2.4 Logging Out of Oracle Enterprise Manager Fusion Middleware Control

This topic 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, 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 **Log Out** item in the **Weblogic** dropdown list.

Performing Basic Administration Tasks

This chapter describes how to create and configure the infrastructure on which Oracle Enterprise Scheduler runs. Oracle Enterprise Scheduler is installed with applications for which it provides scheduling services. However, before you begin using Oracle Enterprise Scheduler, you might have to configure its infrastructure.

This chapter includes the following sections:

- [Introduction to Performing Basic Administration Tasks](#)
- [Installing Oracle Enterprise Scheduler](#)
- [Configuring Perl to Support Process Jobs](#)
- [Configuring Oracle Enterprise Scheduler](#)
- [Searching for Configuration Changes to Oracle Enterprise Scheduler in Cloud Control](#)
- [Starting and Stopping Oracle Enterprise Scheduler Components](#)
- [Starting and Stopping a Request Processor Dispatcher](#)
- [Managing Application Properties](#)
- [Managing Application and Role Policies](#)
- [Managing Oracle Enterprise Scheduler Web Services](#)

3.1 Introduction to Performing Basic Administration Tasks

Oracle Enterprise Scheduler is installed by the product that includes it in order to provide job scheduling services for the product. However, it might be necessary for you to select deployment of Oracle Enterprise Scheduler explicitly during domain creation, or to configure aspects of Oracle Enterprise Scheduler before you can get the most out of the service.

Among the basic administration tasks you *might* need to perform are:

- Installing Oracle Enterprise Scheduler and creating a domain that includes Oracle Enterprise Scheduler components.
- Configuring Oracle Enterprise Scheduler by configuring a cluster, request processor, and request dispatcher. While a cluster is optional (Oracle Enterprise Scheduler can run on a single node), a processor and dispatcher are required in order for Oracle Enterprise Scheduler to do its work.
- Starting and stopping the service instance, its request processors or request dispatchers.

- Manage application and role policies or service web services.

3.2 Installing Oracle Enterprise Scheduler

Oracle Enterprise Scheduler does not have its own installer, but is installed by the installer of the embedding product such as Oracle SOA Suite. Refer to the embedding product's installation documentation for details.

The IDE is installed by the design time installer of the embedding product (for example, the Oracle SOA Suite design time installer). The installer installs the IDE and automatically configures it to Oracle JDeveloper. Before you run JDeveloper, make sure to set the variable `MW_HOME` to the middleware home location as required by the IDE.

The Oracle Enterprise Scheduler runtime component is installed by the design time or production installer of the embedding product (for example, Oracle SOA suite). The embedding product may automatically deploy Oracle Enterprise Scheduler, but if it does not, then it can be deployed using the "Oracle Enterprise Scheduler Service Basic" template to a server or cluster. The "Oracle Enterprise Manager Fusion Middleware Control Plugin for ESS" template must be deployed and selected to enable Oracle Enterprise Scheduler functionality in Oracle Enterprise Manager Fusion Middleware Control.

3.2.1 Targeting Oracle Enterprise Scheduler During Domain Creation

If the Oracle Enterprise Scheduler is not automatically targeted by the embedding product, you must select the Oracle Enterprise Scheduler basic template when the domain is created. By default, the Oracle Enterprise Scheduler is targeted to a default server called `ess_server1`. You can use the following procedure to target it to a different server (for example a SOA server):

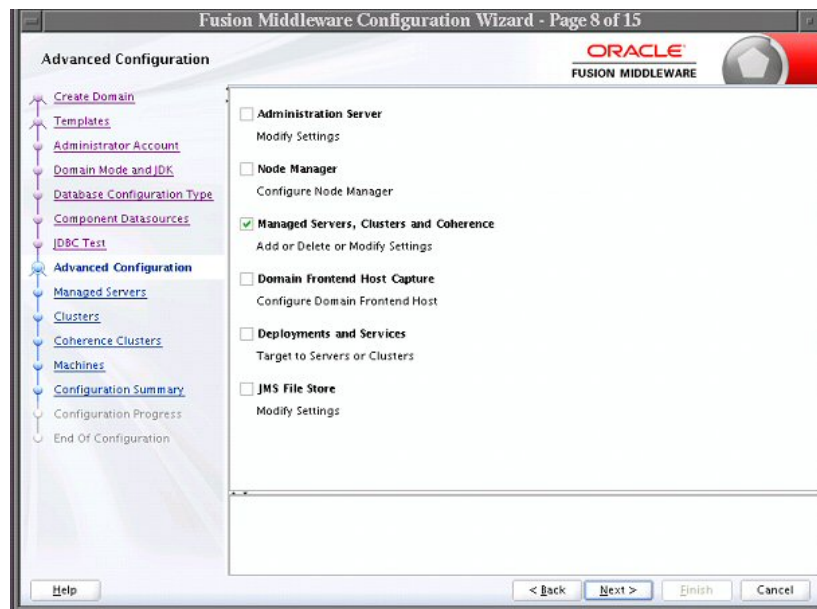
1. When you extend SOA with Oracle Enterprise Scheduler, `ess-server1` is created and the `ESS-MGD-SVRS` server group is targeted to `ess_server1` by default. Use the following steps to target Oracle Enterprise Scheduler to `soa_server1`:
 - a. Check **ESS-MGD-SVRS** for `soa_server1`.
 - b. Uncheck **ESS-MGD-SVRS** for `ess-server1`.
 - c. Delete `ess_server1`.

3.2.2 OWSM-PM Targeting With Oracle Enterprise Scheduler

OWSM-PM is intended to be targeted to just one server in a domain. To facilitate this requirement, Oracle Enterprise Scheduler templates no longer target OWSM-PM. If another product in the domain automatically targets OWSM-PM, then there is nothing to do. However, if there are no managed servers in the domain except for Oracle Enterprise Scheduler, or none of these servers has OWSM-PM, then OWSM-PM must be targeted manually.

3.2.2.1 Targeting OWSM-PM Manually

In the Fusion Middleware Configuration Wizard, select the **Managed Servers, Clusters and Coherence** check box as shown in [Figure 3-1](#).

Figure 3-1 Fusion Middleware Configuration Wizard

On the Managed Server screen, for `ess_server1`, select the `WSMPM-MAN-SVR` server group. `ESS-MGD-SVRS` should already be selected.

3.3 Configuring Perl to Support Process Jobs

To complete domain setup, you *may* need to configure Perl to support process jobs. Oracle Enterprise Scheduler optionally uses Perl for supporting process jobs.

For more information see "How to Create and Store a Process Type Job Definition" in *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Scheduler*.

You do this in the Enterprise Manager.

1. In Enterprise Manager, from the **Scheduling Service** menu, choose **Configuration > Application Properties**.
2. Configure the ESS level property in ESSAPP called `PerlCommand` to the perl executable location. For example, `PerlCommand = c:\Perl\bin\perl.exe`.
3. Perl version is 5.10 or higher.
4. ESS locates perl in the following order:
 - a. From the configuration above.
 - b. If `WL_HOME` is defined, it looks here: `$WL_HOME.../../webtier_mwhome/webtier/perl/bin/perl`.
 - c. Otherwise, it uses it from the container's path.

3.4 Configuring Oracle Enterprise Scheduler

You can run Oracle Enterprise Scheduler as a single instance or as a cluster of servers. Each Oracle Enterprise Scheduler server includes a request processor and dispatcher, both of which must be configured.

Configuring Oracle Enterprise Scheduler involves the following main steps:

1. **Configure a cluster.** Optionally, configure a cluster of Oracle Enterprise Scheduler servers.
2. **Configure the request processor.** Configure the Oracle Enterprise Scheduler component which receives and manages job requests.
3. **Configure the request dispatcher.** Configure the Oracle Enterprise Scheduler component that polls the request processor for job requests and dispatches jobs.

This section contains the following topics:

- [Expanding an Oracle Enterprise Scheduler Cluster](#)
- [Configuring a Request Processor](#)
- [Configuring a Request Dispatcher](#)

3.4.1 Expanding an Oracle Enterprise Scheduler Cluster

An Oracle Enterprise Scheduler cluster is created when the WebLogic domain is created. You may want to expand the cluster to handle a larger load. You can add new cluster nodes to an Oracle Enterprise Scheduler cluster from the Oracle WebLogic Server Console.

When scaling out an Oracle Enterprise Scheduler cluster with an additional managed server, it may be undesirable for the new server to immediately begin processing requests in the default work assignment. This could be the case if all other servers have work assignments bound in standard mode. If one or more of the running servers is using the default work assignment, however, then the present work allocation is compatible with a server using the default work assignment.

When you add a new server to an existing cluster, it may be undesirable for the new server to immediately start processing requests in the default work assignment. This could be the case if all other servers have work assignments bound in standard mode. If at least one of the running servers uses the default work assignment, this implies that the present work allocation is compatible with a server using the default work assignment.

Oracle Enterprise Scheduler provides protection for a new server so the user has an opportunity to perform work allocation before the server starts processing jobs. When a newly created server starts for the first time, Oracle Enterprise Scheduler determines whether using the default work assignment could be problematic and if so, binds a pre-seeded internal work assignment (ESSInternalWA) which includes a health check job. The user can use the health check job to check the server, and then unbind the internal work assignment and bind a user-created work assignment as needed.

Note that when determining if the default work assignment can be used, Oracle Enterprise Scheduler considers all running servers in the group and it is irrelevant what applications are deployed to the servers. Servers that are down are not taken into account.

For more information about work assignments, see [Managing Work Assignments](#).

For more information about expanding a cluster using Oracle WebLogic Server Console, see *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

3.4.2 Configuring a Request Processor

If no work assignment bound to a request processor, the default work assignment processes job requests. The binding of work assignments provides control over what jobs run at what times and with what resources. A binding can be made in one of two modes: standard (the default) or exclusive.

Standard binding mode means the request processor can process job requests as defined by the specialization rules when an active workshift is defined. If a job request is specialized to two different work assignments, either of those work assignments or the default work assignment can process the job request.

When using exclusive binding mode, job requests specialized to the work assignment are processed exclusively by that work assignment when it is active. These job requests are excluded from all other work assignments, including the default work assignment. If the work assignment does not have an active workshift, then the job request can be processed by another work assignment.

As an example, consider the following work assignments:

- LongWA, specialization is (definition = 'JobDefinition://mypackage/LongRunningJob')
- SamWA, specialization is (definition = 'JobDefinition://mypackage/LongRunningJob' AND user = 'sam')

Suppose both LongWA and SamWA are bound in standard mode. When Sam submits LongRunningJob, either LongWA or SamWA can process the request.

Suppose LongWA is bound in standard mode and SamWA is bound in exclusive mode. When Sam submits LongRunningJob, only SamWA can process the request. The exclusive binding causes the specialization for LongWA to act as if it is:

```
(definition = 'JobDefinition://mypackage/LongRunningJob') AND NOT
(definition = 'JobDefinition://mypackage/LongRunningJob' AND user
= 'sam')
```

which is effectively:

```
(definition = 'JobDefinition://mypackage/LongRunningJob') AND
NOT (user = 'sam')
```

Note:

Be careful about using the exclusive binding mode when your specializations overlap. For example, if you bind both LongWA and SamWA in exclusive mode, then SamWA cannot run LongRunningJob at all. In this case, the specialization for SamWA becomes:

```
(definition = 'JobDefinition://mypackage/LongRunningJob'
AND user = 'sam') AND NOT (definition = 'JobDefinition://
mypackage/LongRunningJob')
```

Requirements for binding a work assignment are as follows:

1. The work assignment must be enabled, meaning its active flag must be set.
2. The work assignment must have at least one workshift.

3. Each workshift in the work assignment must have a thread allocation of at least one.
4. If the workshift includes a schedule, the following must be true:
 - The schedule must be active, meaning it is not expired.
 - The workshift duration must be at least one.
5. A work assignment can be bound to a particular server at most one time.
6. A work assignment can be bound to any number of servers in the group, but all bindings must use the same mode. Within a group, a work assignment cannot be bound in standard mode on one server and exclusive mode on another server.

For more information about work assignments, see [Managing Work Assignments](#).

To configure a request processor:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Request Processor** and then select **Configure**.
3. Under the Enable Request Processor field, enable or disable the request processor. Select the **Enable Request Processor** check box to turn on request processing.
4. Optionally, under the Work Assignment Bindings section, bind a work assignment to the request processor.
 - **Add:** To bind a work assignment to the request processor, click **Add**.
From the Select Work Assignment popup window, select a work assignment from the list and click **OK**.
 - **Remove:** To delete a work assignment binding from the job request processor, select a work assignment from the list of bindings and click **Remove**.
 - **Exclusive Binding:** Select to bind a work assignment exclusively to the request processor. The binding applies even if the work assignment is not enabled in the work assignment definition, or if the server is down. If a work assignment is bound to multiple servers, all the servers to which it is bound must use the same binding mode.
5. Optionally, under the Advanced Configuration section, configure the number of threads and the starvation threshold in minutes.
 - **Thread Count:** In the text field, enter the number of threads you want the processor to handle.
 - **Starvation threshold (min):** In the text field, enter the starvation threshold in minutes. The starvation threshold specifies the wait time for job requests.
6. Click **Apply** to save your changes.

3.4.3 Configuring a Request Dispatcher

Use the Configure Request Dispatcher page to enable or disable the job request dispatcher. You can also configure the polling interval for the request dispatcher.

Requests remain in waiting state in the Oracle Enterprise Scheduler repository, until the request dispatcher polls the repository for requests that are ready to run. After polling the repository, the dispatcher sets all requests to ready state. The request processor takes over control of the job requests after they have been placed in ready state.

The default maximum polling interval is 15 seconds.

To configure a request dispatcher:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Request Dispatcher** and then select **Configure**.
3. Under the Enable Request Dispatcher field, enable or disable the request processor.

Select the **Enable Request Dispatcher** check box to turn on request dispatching.
4. Optionally, under the Advanced Configuration section, configure the polling interval for the request dispatcher. In the **Maximum Poll Interval** field, enter the minimum interval at which the dispatcher polls for requests.
5. Click **Apply** to save your changes.

3.5 Searching for Configuration Changes to Oracle Enterprise Scheduler in Cloud Control

You can search for changes made to configurations to Oracle Enterprise Scheduler using the search page in Cloud Control.

To search for changes to configurations to Oracle Enterprise Scheduler in Cloud Control:

1. Display the alerts for Oracle Enterprise Scheduler as described in [Viewing Alerts for Oracle Enterprise Scheduler in Cloud Control](#).
2. In the Monitoring and Diagnostics pane, click the number of configuration changes to display the search page for configuration changes to Oracle Enterprise Scheduler alerts.

ESSAPP Scheduling Service Group

Enterprise Manager automatically collects configuration information for targets such as hosts and databases. Changes to these configurations are recorded and may be viewed from this page.

Search

Target Type: Scheduler Service Group

Target Name: is /domain_name

Configuration Item: Clear

Type of Change: All

Show History Records: Grouped

☒ Include Member Target Changes

☒ Changes Discovered In Last: 7 Days

☐ Changes Discovered

After: Sep 30, 2011 2:11:04 PM

Before: Oct 7, 2011 2:11:04 PM

Search Schedule and Notify History Job Activity Add Filters

Configuration Changes(0) Relationship Changes(0)

View Export Detach

Change Discovered	Target Name	History Records
No data to display		

3. Configure your search as desired.

- **Target Type:** Select the **Scheduler Service** or **Scheduler Service Group**.
- **Target Name:** Search for or enter the name of the target you want to search. From the dropdown list, select the criterion by which you want to search for the target name: **contains**, **is**, **is not**, **does not contain**, **starts with**, or **ends with**.
- **Type of Change:** From the dropdown list, select the type of configuration change for which you want to search.
- **Show History Records:** From the dropdown list, select the way you want the records to display, **Grouped** or **Show All**.
- **Include Member Target Changes:** Select to display configuration changes made to members of the target.
- **Changes Discovered in Last:** Select to search for changes discovered within a given number of days, weeks or months.
- **Changes Discovered:** Select to search for changes discovered within a particular time period. Use the **After** and **Before** fields to enter the relevant time period.

4. Click **Search** to search for the changes.

3.6 Starting and Stopping Oracle Enterprise Scheduler Components

You can start up and shut down an instance of Oracle Enterprise Scheduler from Fusion Middleware Control.

The following components can be started and stopped:

- Oracle Enterprise Scheduler instance.
- Job request processor and dispatcher.

Note:

Stopping an Oracle Enterprise Scheduler instance or component is not recommended. Shutting down an Oracle Enterprise Scheduler component does not stop job requests from accumulating in the queue.

This section contains the following topics:

- [Starting and Stopping an Oracle Enterprise Scheduler Service Instance](#)
- [Starting and Stopping a Request Processor or Dispatcher](#)

3.6.1 Starting and Stopping an Oracle Enterprise Scheduler Service Instance

The Control menu for the scheduling service allows you to start an Oracle Enterprise Scheduler instance and consequently begin managing scheduled job requests.

Note:

Stopping an Oracle Enterprise Scheduler instance or component is not recommended.

To start up an instance of Oracle Enterprise Scheduler:

1. From the Scheduling Service menu, select **Control**.
2. Select **Start Up**.

To shut down an instance of Oracle Enterprise Scheduler:

1. From the Scheduling Service menu, select **Control**.
2. Select **Shut Down**.

3.6.2 Starting and Stopping a Request Processor or Dispatcher

You can start or stop a configured request processor or dispatcher from the Scheduling Service menu.

Note:

Stopping an Oracle Enterprise Scheduler instance or component is not recommended.

To start or stop a request processor or dispatcher:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. Start the request processor or request dispatcher as follows.
 - From the Scheduling Services menu, select **Request Processor** and then select **Start** OR on the Home page, select a request processor in the Scheduler Components area and click on the **Start** button.

- From the Scheduling Services menu, select **Request Dispatcher** and then select **Start** OR on the Home page, select a request dispatcher in the Scheduler Components area and click on the **Start** button.
3. Stop the request processor or request dispatcher as follows.
 - From the Scheduling Services menu, select **Request Processor** and then select **Stop** OR on the Home page, select a request processor in the Scheduler Components area and click on the **Stop** button.
 - From the Scheduling Services menu, select **Request Dispatcher** and then select **Stop** OR on the Home page, select a request dispatcher in the Scheduler Components area and click on the **Stop** button.
 4. When prompted as to whether or not to stop the processor or dispatcher, click **OK**.

3.7 Starting and Stopping a Request Processor Dispatcher

You can start or stop a configured request processor or dispatcher from the Scheduling Service menu.

Note:

Stopping an Oracle Enterprise Scheduler instance or component is not recommended.

To start or stop a request processor or dispatcher:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. Start the request processor or request dispatcher as follows.
 - From the Scheduling Services menu, select **Request Processor** and then select **Start** OR on the Home page, select a request processor in the Scheduler Components area and click on the **Start** button.
 - From the Scheduling Services menu, select **Request Dispatcher** and then select **Start** OR on the Home page, select a request dispatcher in the Scheduler Components area and click on the **Start** button.
3. Stop the request processor or request dispatcher as follows.
 - From the Scheduling Services menu, select **Request Processor** and then select **Stop** OR on the Home page, select a request processor in the Scheduler Components area and click on the **Stop** button.
 - From the Scheduling Services menu, select **Request Dispatcher** and then select **Stop** OR on the Home page, select a request dispatcher in the Scheduler Components area and click on the **Stop** button.
4. When prompted as to whether or not to stop the processor or dispatcher, click **OK**.

3.8 Managing Application Properties

You can configure Oracle Enterprise Scheduler interaction with applications by setting application properties. On the Application Properties page in Fusion Middleware Control, you can set values for properties that are defined by Oracle Enterprise Scheduler or properties specified for the application in its deployed configuration.

Oracle Enterprise Scheduler defines the following properties that you configure on the Application Properties page:

- **RequestFileDirectory:** Specifies the directory for request and log output. The default is "{ESS_ENV:jrfServerLogPath}/ess_request/".
- **RequestFileDirectoryShared:** Specifies a flag indicating whether the request file directory is shared. The default is "false".
- **PerlCommand:** By default, Oracle Enterprise Scheduler uses the Java agent handler for requests in standard and extended mode. It always uses the Perl agent handler for requests in Fusion mode. To use the Perl agent handler in standard and extended request modes, you must add the `PerlCommand` property to the `ess-config.xml` file associated with the hosting application running the process job as shown in the following example.

```
<EssProperties>
  <EssProperty key="RequestFileDirectory" value="/tmp/ess/requestFileDirectory"/>
  <EssProperty key="RequestFileDirectoryShared" value="false"/>
  ...
  <EssProperty key="PerlCommand" value="/usr/bin/perl"/>
</EssProperties>
```

You can use token substitution to specify environment dependent values like directory names.

The Oracle Enterprise Scheduler Perl agent handler requires Oracle Perl version 5.10 or later. Instructions for installing Perl to support process jobs can be found in the chapter "Configuring Perl to Support Process Jobs" in Oracle Fusion Middleware Administrator's Guide for Oracle Enterprise Scheduling Service.

- **EssCallbackClientSecurityPolicyURI:** Specifies the security policy URI used in the `WS-Security` headers for web service invocations from Oracle Enterprise Scheduler for web service callbacks. The default is null.
- **ClusterMode:** Specifies whether the server instance is in standalone or extended mode. This is a read-only and immutable property.
- **HostingAppPolicyStripe:** An Oracle Enterprise Scheduler configuration property that is applicable only for the pre-deployed native hosting application. This property is pre-configured in the pre-deployed native hosting application to support multiple security stripes that can be leveraged by SOA and OSB applications. There is no static definition of policy stripes in the pre-deployed native hosting application's `ejb-jar.xml` file. The list of stripes (for additional components) can be extended dynamically using a runtime configuration MBean, Fusion Middleware Control or WLST scripts.
- **ServerURL:** The value of this property value is a string with the following format:
`http://host:port`

It is used at runtime to determine the `ESSWebService` end point address and is published as part of the `ESSWebService` concrete WSDL.

- **CallbackServerURL:** The value of this property is a string with the following format:

`http://host:port`

It is used at runtime to determine the end point address of Oracle Enterprise Scheduler callback web services (including `EssAsyncCallbackService`, and `EsswsJobAsyncCallbackService`). These end point addresses are published as part of the respective WSDLs.

To edit application properties:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Configuration** and then select **Application Properties**.

The Application Properties page displays.

3. In the **Application** dropdown field, select the application for which you want to set properties.
4. In the **Configuration Properties** section, locate the property whose value you want to change, then enter a new value in the **Value** column for that property.
5. To add a property, click the **Add** button.

The Add Property dialog box appears.

6. To delete a property, click the **Delete** button.
7. From the **Type** dropdown, select the new property's type.

Select **ESS** to choose a property defined by Oracle Enterprise Scheduler, then select the property's name from the **Name** dropdown. Enter the property's value in the **Value** text field.

8. Select another type to enter the name and value for another property.
9. Click **Add** to save the new property setting.
10. On the Application Properties page, click **Apply** to save application properties changes.

3.8.1 Job Location Properties

Oracle Enterprise Scheduler provides the means by which EJB and web service jobs can define a named abstract job location. The job location is specified by the Oracle Enterprise Scheduler `SYS_logicalClusterName` system property and specifies a logical cluster name (LCN). If the job definition for an EJB or web service job specifies a value for an LCN, certain environment-specific properties are specified using Oracle Enterprise Manager Fusion Middleware Control at the hosting application level rather than in the job definition. All job definitions with the same LCN share the value of the properties entered in the hosting application configuration properties using Oracle Enterprise Manager Fusion Middleware Control.

Note:

The terms "logical cluster" and "job location" can be used interchangeably.

To use the job location feature:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Job Metadata** and then select **Job Definitions**. Select a job definition from the **Results** table to open the Job Definitions Details page.
3. In the **System Properties** section, set the SYS_logicalClusterName system property to a name that you use to define the set of location related properties in the Application Properties page.
4. From the Scheduling Services menu, select **Configuration** and then select **Application Properties**.
5. Select the hosting application in the **Application** dropdown menu.
6. In the **Job Location Properties** section of the Applications Properties page, click the **Add Job Location** icon.

The Add Job Location dialog box appears.

7. In the **Job Location** field, enter the name you assigned to the SYS_logicalClusterName system property in step 3.
8. Fill in the values for the three fields that correspond to the properties shown in [Table 3-1](#).

Table 3-1 Job Location Properties

Field	Property Name	Description
JNDI Provider URL	JndiProviderUrl	The JNDI provider for the cluster. Used with the EJB job type.
Web service endpoint base URL	WsEndpointBaseUrl	The host and port. Used with the web service job type.
Web service WSDL base URL	WswsdlBaseUrl	The host and port of the WSDL. Used with the web service job type.

9. Click **Add** to save the new property setting.
10. On the Application Properties page, click **Apply** to save any application properties changes.

Note:

You can also delete job locations and properties by selecting them in the table and then clicking **Delete Job Location** or **Delete Property**.

3.9 Managing Application and Role Policies

To control access to Oracle Enterprise Scheduler data and metadata, you can create application roles to represent users and user groups, then grant them access to specific application resources.

This section describes how to create application roles. You might also be interested in the following sections:

- For more on granting access to metadata, see [Managing Metadata Security for Jobs](#) in [Managing the Work of Oracle Enterprise Scheduler Jobs](#).
- For more on granting access to data, see [Configuring Simple Data Security for Job Requests](#) in [Managing Oracle Enterprise Scheduler Requests](#).

3.9.1 Managing Application Roles

Using Fusion Middleware Control, you can create application roles with which you later associate people and access privileges.

1. From the navigation pane, expand the **WebLogic Domain** folder and select the domain for which you're creating policies.
2. From the WebLogic Domain menu, select **Security** and then select **Application Roles**.

The Application Roles page displays.

3. In the Search section, from the Application Stripe dropdown, select the application stripe with which you want to work.
4. Click **Create**.

The Create Application Role page displays.

5. In the General section, in the **Role Name** field, enter the name for the application role you're creating.
6. Optionally, enter a display name and description in the fields provided.
7. In the Members section, click **Add** to search for members to add to the role.
8. In the Add Principal window, in the Search section, from the **Type** dropdown select a principal type. You can also add name information to narrow your search.

Note that the Type values are as follows:

- An Application Role type is another application role.
 - A Group is a WebLogic Server user group.
 - A User is a WebLogic Server user.
9. Click the **Search** button to search for a principal based on the type and name you specified.
 10. In the **Searched Principals** section, select the name of a principal you want to add to the role.

11. Click **OK**.

12. In Create Application Role page, click **OK**.

3.10 Managing Oracle Enterprise Scheduler Web Services

Oracle Enterprise Scheduler web services are Oracle Infrastructure Web Services.

For information about managing the web services, see the chapter "Understanding Oracle WSM Policy Framework" in the *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

Note:

Oracle Enterprise Scheduler web services cannot be invoked from a web browser.

3.10.1 Securing Web Services

You can use either globally attached policies or directly attached policies to configure OWSM policies for Oracle Enterprise Scheduler web services and Oracle Enterprise Scheduler asynchronous callback web services.

Note:

Both the Oracle Enterprise Scheduler web service and asynchronous callback web service are Oracle Infrastructure Web Services.

Instructions for attaching globally and directly attached OWSM policies is documented in the "Attaching Policies" chapter in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

The following sections of *Securing Web Services and Managing Policies with Oracle Web Services Manager* can be especially useful when configuring globally attached or directly attached policies for Oracle Enterprise Scheduler web services:

- 4.3.1 Attaching Policies Directly Using Fusion Middleware Control
- 4.3.2 Attaching Policies Globally Using Fusion Middleware Control
- 4.4.3 Attaching Policies Directly to RESTful and Oracle Infrastructure Web Services and Clients Using WLST
- 4.4.4 Attaching Policies Globally Using WLST

Managing Oracle Enterprise Scheduler Requests

This chapter describes how to manage Oracle Enterprise Scheduler job requests, creating and submitting them, as well as holding and canceling them as required.

Job requests do work on behalf of an application. A job request consists of the job execution type, such as Java job or PL/SQL job, a job definition that associates metadata for specifics about the request, and a schedule that guides when the job runs. Managing job requests can mean creating and submitting them, but also holding and canceling them, as well as doing clean-up by purging requests from the database.

Note:

You can also use Oracle WebLogic Scripting Tool to perform request management tasks. For more information, see "Enterprise Scheduler Custom WLST Commands" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

This chapter includes the following sections:

- [Introduction to Managing Oracle Enterprise Scheduler Job Requests](#)
- [Managing Oracle Enterprise Scheduler Requests](#)
- [Managing Purge Policies](#)
- [Configuring Simple Data Security for Job Requests](#)

4.1 Introduction to Managing Oracle Enterprise Scheduler Job Requests

Using Fusion Middleware Control Console, you can manage the job requests that do work on behalf of applications for which Oracle Enterprise Scheduler is providing services.

Tasks related to managing job requests include creating job requests and specifying the schedules on which they run. You can submit the requests, as well as hold and cancel them.

Job request administration tasks you might need to perform include:

- Creating job requests by associating request-specific metadata as job definitions, then defining schedules that guide when the job runs.
- Holding, resuming, and canceling requests.
- Purging job requests from the database.

4.2 Managing Oracle Enterprise Scheduler Requests

An application uses Oracle Enterprise Scheduler to execute jobs on the application's behalf. When the application wants to execute a job, it sends a job request to Oracle Enterprise Scheduler. You configure those jobs by using job definitions that specify the job's parameters and associate them with the application. When you want to group together a collection of job requests, you create a job set.

The work of administering job requests includes submitting requests, searching for existing requests and viewing information about them, and pausing and canceling requests. You can do these tasks with Fusion Middleware Control Console.

This section contains the following topics:

- [Submitting an Oracle Enterprise Scheduler Job Request](#)
- [Searching for Oracle Enterprise Scheduler Job Requests](#)
- [Creating or Editing Predefined Job Schedules](#)
- [Viewing Job Request Details](#)
- [Holding and Resuming Oracle Enterprise Scheduler Job Requests](#)
- [Canceling Oracle Enterprise Scheduler Job Requests](#)

4.2.1 Submitting an Oracle Enterprise Scheduler Job Request

You can submit a job request by choosing a job definition that defines the metadata you want and by defining a schedule that guides when the job runs.

This section contains the following topics:

- [Creating a Job Request](#)
- [Specifying a Job Request Schedule](#)

4.2.1.1 Creating a Job Request

You create a job request by selecting a job definition for the job request, then selecting or creating a schedule.

You may want to configure system properties for the job request, such as the number of retries to attempt in the event of an execution error and a time out value for the job.

To submit a job request:

1. Use a pre-existing job definition. You can also first create a new job definition, as described in [Creating or Editing a Job Definition](#).
2. Navigate to the Submit Job Request page by clicking the Scheduling Service menu selecting **Job Requests**, then selecting **Submit Job Request**.
3. Select the application for which you want to submit the job request.
From the **Application** dropdown list, select the application deployment name.
4. Under Job Request Details, enter a description of the job request in the **Submission Notes** text field.
5. Select a job definition.

- a. Under Job Definition, click the search button next to the **Job Definition** text field.
 - b. Search for the required job definition.
In the **Name** text field, enter the name of the job definition you want to find. You can also enter the job definition's package in the **Package** field.
Click **Go** to search for the job definition.
 - c. From the search results table, select the job definition name.
 - d. Click **OK**.
6. In the Parameters region, optionally define any parameters you want to use with the scheduled job request.
 - a. In the Job Parameters region, click **Add** to add a parameter.
Fill in a name and value for the parameter.
 - b. You can set a value for any of the system properties available.
For example, you may want to configure the number of permissible retries for the job (**SYS_retries**), or a time out value (**SYS_request_timeout**).
7. Select a schedule by which the job is to run.
 - **Once:** Select a single date and time by clicking the **Calendar** button.
 - **Use existing schedule:** Click the search button to search for and select an existing schedule. For more on defining saved schedules, see [Creating or Editing Predefined Job Schedules](#).
 - **Specify schedule:** Create your own schedule on the fly. Follow the instructions in [Specifying a Job Request Schedule](#)
8. Click **OK**.

4.2.1.2 Specifying a Job Request Schedule

In the Submit Job Request page, selecting Specify schedule enables configuring your own schedule for a job request. You can configure a recurring job request using a job request schedule. Alternatively, you can configure a job request to run immediately or before a specified end date. Use a pre-existing job definition, or create a new job definition, as described in [Creating or Editing a Job Definition](#).

To specify a schedule:

1. In the Schedule region, select **Specify schedule**.
2. From the **Frequency** dropdown, select how often you want job requests using this schedule to run. Each frequency option has its own way to specify details for the option. As you specify details, note that you can also use the **Time Zone** dropdown to specify time zones.
3. Select the **Save Schedule** check box to save this schedule for later use by this or other job requests, then enter the following:
 - **Name:** Enter a name for the schedule. This is the name that appears when you assign the schedule to a job request.

- **Display Name:** Enter a value that is played to users for this schedule.
- **Package:** Optionally enter the name of a Java package related to this schedule if it helps you find or group schedules.
- **Description:** Optionally enter a description.

4.2.2 Searching for Oracle Enterprise Scheduler Job Requests

The Request Search page enables searching for job requests. Using the Request Search page, you can:

- Conduct a quick search by selecting a pre-configured search option from the **Quick Search** dropdown menu.
- Conduct a basic search that returns a list of job request details, including job request ID, executing application, job request status, and so on.
- Conduct an advanced search based on Run Time/Wait Time, Retry of Failed Runs, Date Range, Error Type and Request ECID/Correlation Flow ID along with the fields available as part of Basic Search.
- Modify the column display in the search results table.

This section contains the following topics:

- [Searching for a Job Request Using Simple Search](#)
- [Searching for a Job Request Using the Advanced Search Feature](#)
- [Configuring the Display of Columns in the Search Results Table](#)

4.2.2.1 Searching for a Job Request Using Simple Search

Basic search allows you to find a job request according to particular criteria such as job request ID, related application, job request status, and so on, or by any one of a number of pre-configured quick searches.

To search for a job request:

1. Navigate to the Request Search page by clicking the Scheduling Service menu and selecting **Job Requests**, then selecting **Search Job Requests**.
2. Select the scope of the job request search by selecting one of the following options:
 - **Current Scheduling Service:** Select this option to search for job requests submitted only to the scheduling service with which you are currently working.
 - **All Scheduling Services sharing the ESS repository:** Select this option to search for job requests submitted to all scheduling services sharing the repository -- for example, all scheduling services in a cluster of scheduling services.

The repository -- the Oracle Enterprise Scheduler run-time database that holds jobs requests -- can be shared across multiple domains with Oracle Enterprise Scheduler. This might be useful to see jobs running across domains that might be stressing a shared resource, such as a database.

3. To run a fast search, from the **Quick Search** dropdown list, select a pre-configured search option as shown in the following list.

- Requests submitted in the last hour
- Pending requests submitted in the last 24 hours
- Errored requests submitted in the last 24 hours
- All running requests
- All pending requests
- Requests currently being retried
- Requests retried in the last 24 hours
- Requests that resulted in System Error in the last 24 hours
- Requests that resulted in Business Error in the last 24 hours
- Ready requests for selected Work Assignment.
- Asynchronous requests that require manual recovery.

To run a regular job request search, skip this step.

4. Enter or select the criteria by which to search for job requests.

- **Request ID:** Enter the ID of the job request for which you want to search.
- **Application:** From the Application dropdown list, select the name of the application related to the job request for which you want to search. Alternatively, select **All** to search for job requests in all applications.
- **Status:** Select the status of the job request for which you want to search. Alternatively, select **All** to search for job requests with all statuses.

The job request statuses are:

- Blocked
- Cancelled
- Cancelling
- Completed
- Error
- Error Auto Retry
- Error Manual Recovery
- Expired
- Finished
- Hold
- Paused
- Pending Validation
- Ready

- Running
- Schedule Ended
- Succeeded
- Unknown
- Validation Failed
- Wait
- Warning
- **Execution Type:** Options include **Java Type**, **SQL Type**, and **Process Type**.
- **Submitted:** From the dropdown list, select the time period in which the job request to be searched has been submitted. Options include **Last Hour**, **Last 24 Hours**, **Last 7 Days**, and **Last 31 Days**.
- **Submitted By:** In the text field, enter the name of the user who submitted the job request you want to find.
- **Submission Notes:** In the text field, enter any notes associated with the submission.
- **Job Definition:** Click the search button next to the text field and select the relevant job definition name.
- **Job Location:** Specify an abstract location for EJB and web services jobs. See [Job Location Properties](#) for more information.
- **Work Assignment:** Click the search button next to the text field and select a work assignment from the list.
- **Product:** In the text field, enter the name of the product using the job request.
- 5. Optionally, conduct an advanced search by clicking the **Advanced** button. For more on advanced search, see [Searching for a Job Request Using the Advanced Search Feature](#).
- 6. Click **Search** to submit the job request search.

4.2.2.2 Searching for a Job Request Using the Advanced Search Feature

An advanced search is available in the Request Search page by clicking the **Advanced** button.

To search for a job request:

1. Enter a basic search for a job request. Navigate to the Request Search page by clicking the Scheduling Service menu and selecting **Job Requests**, then selecting **Search Job Requests**.
2. Select the scope of the job request search by selecting one of the following options:
 - **Current Scheduling Service:** Select this option to search for job requests submitted only to the scheduling service with which you are currently working.
 - **All Scheduling Services sharing the ESS repository:** Select this option to search for job requests submitted to all scheduling services sharing the

repository, for example all scheduling services in a cluster of scheduling services.

The repository -- the Oracle Enterprise Scheduler run-time database that holds jobs requests -- can be shared across multiple domains with Oracle Enterprise Scheduler. This might be useful to see jobs running across domains that might be stressing a shared resource, such as a database

3. Select your basic search criteria. For more information, see [Searching for a Job Request Using Simple Search](#).
4. Click **Advanced** to display the fields for the advanced search.
5. In the Date Range section, configure the date range in which to search for job requests. The date on the left is the beginning date and the date on the right is the end date. For each date, click the **Calendar** button to the right of the text field to select the date and time.
 - **Submitted between:** Enter the start and end dates during which the job request was submitted.
 - **Scheduled between:** Enter the start and end dates during which the job request is scheduled to run.
 - **Completed between:** Enter the start and end dates during which the job request finished running.
6. In the Run Time/Wait Time section, select the run or wait time of the job request for which you are searching, such as long or short running requests.
 - **None:** Select if no run or wait time is to be specified.
 - **Long running requests:** Select to search for requests running longer than a specified number of seconds, minutes, hours or days.

In the **Minimum Run Time** text field, enter the lower limit of the time period for which the job request runs. From the dropdown list, select the required unit of time.

- **Short running requests that waited:** Select to search for job requests running longer than a specified period of time and waiting less than a specified period.

In the **Maximum Run Time** text field, enter the upper limit of the time period for which the job request runs. From the dropdown list, select **Seconds**, **Minutes**, **Hours** or **Days**.

In the **Minimum Wait Time** text field, enter the lower limit of the time period during which the job request waits to run. From the dropdown list, select **Seconds**, **Minutes**, **Hours** or **Days**.

- **Waiting requests by time:** Select to search for job requests waiting to run for a specified time period.

In the **Minimum Wait Time** text field, enter the lower limit of the time period during which the job request waits to run. From the dropdown list, select **Seconds**, **Minutes**, **Hours** or **Days**.

In the **Maximum Wait Time** text field, enter the upper limit of the time period during which the job request waits to run. From the dropdown list, select **Seconds**, **Minutes**, **Hours** or **Days**.

7. In the Retry of Failed Runs section, use the **Number of Retries** dropdown list to select an operator such as equal to, greater than, greater than or equal to, and so on.

In the text field, enter the number of retries.

8. In the Error Type section, use the dropdown list to select the type of error:
 - **Business:** A job ends in a business error when it must abort prematurely due to unforeseen conditions, but is otherwise able to exit with its data in a consistent state. A job request might end in a business error as a result of an application setup or configuration condition, a functional conflict that requires an early exit or corrupt or inconsistent data. You cannot retry running a job request that ends in a business error.
 - **System:** A job ends in a system error when a technical problem occurs from which the job cannot recover, but otherwise exits on its own. Alternatively, the computer running the job crashes. Examples include table space issues and unhandled run-time exceptions. You can retry running a job request that ends in a system error.
9. In the Request ECID/Correlation Flow ID section, use the **Request ECID** and **Correlation Flow ID** fields to specify the values for your request.
10. Click **Search** to submit the job request search.

4.2.2.3 Configuring the Display of Columns in the Search Results Table

After running a search for job requests, you can configure the display of columns in the search results table.

To configure job request search results table display columns:

1. Display the main Oracle Enterprise Scheduler Request search page and display the search interface. [topicid:ess_searchlogmsgs_proc](#)
2. Show or hide columns.

Click the **View** dropdown list and select **Columns**. then click column names to add check marks for the ones you want and remove check marks for those you do not.

You can also instead click **Manage Columns** to view the Manage Columns dialog box, through which you can show or hide columns. Use the arrows between the columns to move column names from the **Visible** to the **Hidden** column, or vice versa.

3. Sort columns.

You can sort selected columns in ascending or descending order by clicking on the **View** dropdown list and selecting **Sort**. Click on **Advanced** to sort columns using multiple criteria.

4. Reorder columns.

You can reorder the columns in the table two ways:

- Select **View > Columns > Manage Columns**
- Select **View > Reorder Columns**

5. Alternatively, display all columns by clicking the **View** dropdown list and select **Columns**, then **Show All**.

4.2.3 Creating or Editing Predefined Job Schedules

You can define and save schedules for later use in submitting job requests.

1. Navigate to the Schedules page by clicking the Scheduling Service menu, selecting **Job Requests**, then selecting **Define Schedules**.
2. On the Schedules page, you can choose one of the following actions:
 - To create a new schedule:
 - a. From the **Application** dropdown list, select the name of the application for which you want to create a schedule.
 - b. Click **Create**.
 - c. Enter the following information:

Name: Enter a name for the schedule. This is the name that appears when you assign the schedule to a job request.

Package: Optionally enter the name of a Java package related to this schedule if it helps you find or group schedules.

Description: Optionally enter a description.
 - d. From the **Frequency** dropdown, select how often you want job requests using this schedule to run. Each frequency option has its own way to specify details for the option. As you specify details, note that you can click **Select date and time** icons to specify time zones as well.
 - e. Click **OK**.
 - To view details for an existing schedule:
 - a. In the **Results** area, locate the schedule whose details you want to view.
 - b. From the **Application** dropdown list, select the name of the application whose schedules you want to view.
 - c. In the **Name** column, click the schedule's name.
 - d. From the **Schedule Details** page, you can remove schedule times or click the **Edit** button to edit schedule details.
 - To edit an existing schedule:
 - a. In the **Name** and **Package** fields, enter values to search for.
 - b. Click **Go** to search.
 - c. In the **Results** area, locate the schedule you want to edit.
 - d. Select the schedule you want to edit, then click **Edit**.
 - e. On the Edit Schedule page, edit details about the schedule.
 - f. Click **OK**.

4.2.4 Viewing Job Request Details

You can view detailed information about an individual job request by clicking the job request ID or the request parent ID in job request search results. If the job is in an error state, an information box displays details regarding the error that occurred. For more on searching for job requests, see [Searching for Oracle Enterprise Scheduler Job Requests](#).

You can take the following actions on an individual job request:

- Display log information for the job request.
- Display the job set and all child job requests.
- Submit a job or job set request with parameters similar to those of a given job request.
- Recover an incomplete job request.

To view job request details:

1. Search for the relevant job requests as described in [Searching for a Job Request Using Simple Search](#)
2. In the table displaying the job request search results, select the job request whose details you want to view.
3. To view job request details, click the job request ID. Alternatively, click the parent ID associated with the job request to view the details of the job set with which the job is associated.

In the job request details page, you can take any of the following actions.

- **Request Log:** Select **Action** and then select Request Log to display log information for the job request.
- **SOA Composite Flow Trace:** Select **Action** and then select SOA Composite Flow Trace to display the user interface for the flow of messages through composite and component instances. Use this command in cases when request status has not been updated from the SOA composite. The request's flow trace provides information related to activity in the composite.

This action is available for asynchronous Java jobs where everything is in the same domain. If the job happens not to be a SOA composite web service invocation, then the flow trace user interface displays a "Trace not found" message.

- **Request Tree:** Select **Action** and then select Request Tree to display the parent job set along with all related child jobs.
- **Cancel:** Select **Action** and then select Cancel to cancel the job request.
- **Hold:** Select **Action** and then select Hold to temporarily suspend the job request.
- **Resume:** Select **Action** and then select Resume to resume a suspended job request.

- **Recover Stuck Request:** Select **Action** and then select Recover Stuck Request to recover an incomplete job request that cannot continue.
- **Change Priority:** Select **Action** and then select Change Priority to raise or lower the priority of the job request. Job requests with a higher priority are dispatched prior to job requests with lower priority.
- **Change Schedule:** Select **Action** and then select Change Schedule to assign a different schedule to the job request.
- **Submit Like:** Select **Action** and then select Submit Like to submit a job or job set with parameters similar to this one.

4.2.5 Holding and Resuming Oracle Enterprise Scheduler Job Requests

You can hold or resume a job request after it has been submitted.

To hold and resume a job request:

1. Search for a job request. [topicid:ess_searchjobrqst_proc](#) For more information, see [Searching for a Job Request Using Simple Search](#).
2. In the table of job request search results, select the job request you want to hold.
3. Click the **Hold** button.

You can also hold a job request from the Request Details page by selecting **Select Action > Hold**.

4. To resume the paused job request, select the job request and click **Resume**.

You can also resume a job request from the Request Details page by selecting **Select Action > Resume**.

4.2.6 Canceling Oracle Enterprise Scheduler Job Requests

You can cancel an executable job request after it has been submitted. Executable job requests include job sets and job requests submitted without schedules. A job request can be canceled when it is not in a state of execution.

When initiating a job request cancellation, the resulting cancellation state depends on the current state of the job request. A job request that is not executing is set directly to CANCELLED state. A job request in RUNNING, PAUSED or COMPLETED states are placed in CANCELLING state following the initiation of request cancellation.

The final state of a request depends on the processing stage the job request at the point when cancellation has been initiated.

[Table 4-1](#) displays the cancellation states applied to each executable request depending on its state when initiating cancellation.

Table 4-1 Executable Requests and Cancellation States

Job State When Initiating Cancellation	New Cancellation State
WAIT	CANCELLED
READY	CANCELLED
HOLD	CANCELLED

Table 4-1 (Cont.) Executable Requests and Cancellation States

Job State When Initiating Cancellation	New Cancellation State
BLOCKED	CANCELLED
PENDING_VALIDATION	CANCELLED
ERROR_AUTO_RETRY	CANCELLED
PAUSED	CANCELLING
RUNNING	CANCELLING
COMPLETED	CANCELLING
Any terminal state	No state change
CANCELLING	No state change
ERROR_MANUAL_RECOVERY	No state change

An executable request is processed in three major stages: pre-processor, executable and post-processor. Requests can be canceled at any of these stages. If the job request cannot complete all three stages, it transitions to CANCELLED.

This section contains the following topics:

- [Initiating Cancellation During Pre-Process Handler Execution](#)
- [Initiating Cancellation During Synchronous Java Job Execution](#)
- [Initiating Cancellation During Asynchronous Java Job Execution](#)
- [Initiating Cancellation During PL/SQL Job Execution](#)
- [Initiating Cancellation During Spawned Job Execution](#)
- [Canceling Parent Job Requests](#)
- [Canceling a Child Job Request](#)

4.2.6.1 Initiating Cancellation During Pre-Process Handler Execution

When initiating cancellation during the pre-processor stage, the state to which the job request transitions depends on the status returned by the pre-processor.

[Table 4-2](#) displays the job state following cancellation initiation.

Table 4-2 Pre-processor States and States Following Cancellation Initiation

State Returned by the Pre-Processor	State Following Cancellation Initiation
PROCEED	CANCELLED
DELAY	CANCELLED
WARN	CANCELLED
SYSTEM_ERROR	ERROR

Table 4-2 (Cont.) Pre-processor States and States Following Cancellation Initiation

State Returned by the Pre-Processor	State Following Cancellation Initiation
BIZ_ERROR	ERROR

4.2.6.2 Initiating Cancellation During Synchronous Java Job Execution

When initiating cancellation during the job execution stage, the state to which the request transitions depends on the manner of completion.

[Table 4-3](#) displays the job state following cancellation initiation.

Table 4-3 Returned Synchronous Java Job States and States Following Cancellation Initiation

State Returned by the Job	State Following Cancellation Initiation
Normal return (success)	CANCELLED
ExecutionWarningException	CANCELLED
ExecutionPausedException	CANCELLED
ExecutionBizErrorException	ERROR
ExecutionErrorException	ERROR

4.2.6.3 Initiating Cancellation During Asynchronous Java Job Execution

When initiating cancellation during the job execution stage, the state to which the request transitions depends on the manner of completion.

[Table 4-4](#) displays the job state following cancellation initiation.

Table 4-4 Returned Asynchronous Java Job States and States Following Cancellation Initiation

State Returned by the Job	State Following Cancellation Initiation
SUCCESS	CANCELLED
PAUSE	CANCELLED
WARNING	CANCELLED
ERROR	ERROR
CANCEL	CANCELLED
UPDATE	No state change. Wait for further updates.
BIZ_ERROR	ERROR
ERROR_MANUAL_RECOVERY	ERROR_MANUAL_RECOVERY

4.2.6.4 Initiating Cancellation During PL/SQL Job Execution

If the job executable is executing, Oracle Enterprise Scheduler attempts to kill the running RDBMS Scheduler job. If the job is successfully killed, the request transitions

to CANCELLED state. If the RDBMS Scheduler job completes before it can be killed, the state to which the job request transitions depends on the result of the job execution.

Table 4-5 displays the job state following cancellation initiation.

Table 4-5 Returned PL/SQL Job States and States Following Cancellation Initiation

State Returned by the Job	State Following Cancellation Initiation
SUCCESS	CANCELLED
WARNING	CANCELLED
PAUSE	CANCELLED
BIZ_ERROR	ERROR
SYSTEM_ERROR	ERROR

4.2.6.5 Initiating Cancellation During Spawned Job Execution

If the spawned job is executing, Oracle Enterprise Scheduler attempts to kill the running process. If the process is successfully killed, the request transitions to a CANCELLED state. If the process completes before it can be killed, the state to which the request transitions depends on the result of the process execution.

Table 4-6 displays the job state following cancellation initiation.

Table 4-6 Returned Spawned Job States and States Following Cancellation Initiation

State Returned by the Job	State Following Cancellation
SUCCESS	CANCELLED
WARNING	CANCELLED
PAUSE	CANCELLED
BIZ_ERROR	CANCELLED
SYSTEM_ERROR	CANCELLED

4.2.6.6 Canceling Parent Job Requests

A parent job request that is canceled transitions to a CANCELLING state until all child requests complete or have been canceled. The cancellation operation propagates to all eligible child job requests. For example, job set steps are canceled when the job set is canceled, sub-requests are canceled when the parent request is canceled and recurring executable job requests are canceled when the recurring parent is canceled. If a child job request is executable, it is subject to the rules described in the preceding sections. When all the child requests have completed, the parent request transitions to CANCELLED state.

4.2.6.7 Canceling a Child Job Request

A child job request may be an executable or a parent job request. When canceling a sub-request that completes in CANCELLED or another terminal state, the parent job request resumes as usual as long as the parent job request has not been canceled as well. The state of the sub-request does not affect the state of the parent request.

When canceling a step within a job set, the job set transitions to a **CANCELLED** state when the job set step transitions to a **CANCELLED** state. However, the job set may not revert to **CANCELLED** state if another job set step results in an error.

To cancel a job request:

1. Search for a job request.
2. In the table of job request search results, select the job request you want to cancel.
3. Click the **Cancel** button.

4.3 Managing Purge Policies

Purge policies allow the scheduling service to remove completed jobs according to specified criteria.

For example, a purge policy might specify the retention of all Java type job requests using a particular job definition submitted executed by a given application for three days. Another purge policy might retain a particular type of job request, say, all SQL job requests in a successful state, for only one day. You can also specify the frequency at which the purge policy is to run.

This section contains the following topics:

- [Setting Up a New Purge Policy](#)
- [Physically Purging Job Requests from the Database](#)
- [Viewing Purge Policies](#)
- [Updating a Purge Policy](#)
- [Deleting a Purge Policy](#)

4.3.1 Setting Up a New Purge Policy

A purge policy determines which job requests are to be purged and which retained. Defining a purge policy involves:

- Selecting the jobs to be purged: Selection criteria include the related application or product, a particular job definition or job type, the job submitter or a maximum number of requests.
- Specifying retention criteria: Decide how long job requests are to be retained depending on their status.
- Specifying purge frequency: Decide how often you want the purge policy to run.

Note:

After a purge policy has run, it is necessary to physically delete purged job requests from the database. For more information, see [Physically Purging Job Requests from the Database](#).

To set up a new purge policy:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.

2. From the Scheduling Services menu, select **Purge Policies**.
3. To configure a new policy, click **Setup New**.
In the Setup Purge Policy page, configure the purge policy.
4. In the **Description** text field, enter a description for the purge policy.
5. In the Request Criteria for Purge section, configure the characteristics of the job requests to be purged.
 - **Application:** From the dropdown list, select the application for which you want to create a purge policy.
 - **Product:** Enter the name of the product.
 - **Job Definition:** Click the browse button next to the **Job Definition** text field.
In the Select Job Definition window, enter the name of the job definition in the **Name** text field.
Click **Go** to search, then select the relevant job definition from those that display. Click **OK**.
 - **Execution Type:** From the dropdown list, select the job type required: All, Java Type, SQL Type, or Process Type.
 - **Submitted by:** Enter the name of the job submitter.
6. In the Retention Criteria for Purge section, configure the characteristics of the job requests to be retained.
 - **Default Retention Period (in days):** Enter the default period, in days, during which requests are to be retained.
 - **Retention Period - Success (in days):** Enter the period, in days, during which successful job requests are retained.
 - **Retention Period - Error (in days):** Enter the period, in days, during which errored job requests are retained.
 - **Retention Period - Warning (in days):** Enter the period, in days, during which job requests ending in a warning status are retained.
 - **Retention Period - Cancelled (in days):** Enter the period, in days, during which canceled job requests are retained.
7. In the Schedule section, set a schedule for the job request purge policy.
 - To run the purge policy only one time, select **Once**. Click the **Calendar** button to select the date and time you want the purge policy to run.
 - To set a schedule for the purge policy, select **Use Existing Schedule**. Click the browse button to search for a schedule.
In the Select Schedule window, select a schedule and click **OK**.
8. In the Advanced section, in the **Maximum requests to be processed** text field, enter the maximum number of job requests to be processed.
9. Click **OK** to save the purge policy.

4.3.2 Physically Purging Job Requests from the Database

Job request data is kept in the Oracle Enterprise Scheduler run-time store as records in the run-time store tables. These job requests records are kept in the run-time store until they are physically purged by a database administrator using SQL purge scripts. You must logically delete a job request before physically purging it.

Use the method `esspurge.purge_requests` on the Oracle Enterprise Scheduler schema to delete purged job requests from the database. In an Oracle Fusion Applications environment, the schema is typically called `FUSION_ORA_ESS`.

The `ESSPURGE` package contains a stored procedure which you can use to purge completed job requests. The package is normally loaded when the Oracle Enterprise Scheduler schema is created or updated. The stored procedure is shown in [Example 4-1](#).

The basic syntax of `esspurge.purge_requests` is shown in [Example 4-2](#), where `FUSION_ORA_ESS` is the name of the Oracle Enterprise Scheduler schema and `password` is the password.

Additional examples are shown in the following list.

- To purge job requests completed earlier than the current time, at a maximum of 50000 job requests, execute the following command:

```
execute esspurge.purge_requests(systimestamp, 50000);
```
- To purge job requests completed 30 days earlier, at a maximum of 50000 job requests, execute the following command:

```
execute esspurge.purge_requests(systimestamp - 30, 50000);
```
- To purge job requests that completed before June 01, 2010 at 15:00:00 (UTC), at a maximum of 50000 job requests, execute the following command:

```
execute esspurge.purge_requests(TIMESTAMP '2010-06-01 15:00:00 -00:00', 50000);
```

Example 4-1 *ESSPURGE Stored Procedure for Purging Completed Job Requests*

```
--- Purges job requests that have completed.
---
--- p_older_than : Purge only job requests that have completed after this time.
--- p_max_count : The maximum number of job requests to purge.
--- p_max_runtime : The maximum time, in minutes, the purge should run.
---   If null (default), this defaults to one day which effectively means
---   there is no time limit.
---
procedure purge_requests
( p_older_than in timestamp,
  p_max_count in number,
  p_max_runtime integer default null );
```

Example 4-2 *Purging Job Requests from the Database*

```
sqlplus FUSION_ORA_ESS/password
set serveroutput on size unlimited
set timing on
execute esspurge.purge_requests(systimestamp, 1000000);
```

4.3.3 Viewing Purge Policies

You can view all purge policies defined for the scheduling service.

To display purge policies for the scheduling service:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Purge Policies**.

4.3.4 Updating a Purge Policy

You can edit the request or retention criteria of an existing purge policy, as well as the purge policy schedule as long as the purge policy is not in the "Succeeded" or "Cancelled" state.

To update a purge policy:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Purge Policies**.
3. From the Purge Policies table, select the policy you want to update and click **Update**.
4. Edit the purge policy accordingly and click **OK**.

4.3.5 Deleting a Purge Policy

You can delete a given purge policy as long as it is not in the "Wait" or "Hold" state.

To delete a purge policy:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Purge Policies**.
3. From the Purge Policies table, select the policy you want to delete and click **Delete**.

4.4 Configuring Simple Data Security for Job Requests

Oracle Simple Data Security enforces security authorizations for access and modification of specific data records. Oracle Simple Data Security integrates with Oracle Platform Security Services (OPSS) by granting actions to OPSS principals.

The grant defines who (the principals) can do what (the actions) on a given resource. A grant in Oracle Simple Data Security can use any enterprise user or enterprise group as principals.

In the context of Oracle Enterprise Scheduler, a job request access control data security policy comprises a grant, a grantee and a set of `oracle.as.scheduler.security.RuntimeDataPermission` privileges for a set of job requests as follows:

- A grantee, represented by grantee ID such as a user or application role. The ID should match the user GUID or application role GUID retrieved from Oracle Fusion Middleware.

You can manage the job request access control data security policy using Oracle Enterprise Manager Fusion Middleware Control.

The following behaviors are in place by default:

- A user by default can see only the requests they submitted.
- If a user can see a request, then they can see the request logs.
- If a user can see a request, and if the request does not run as elevated privilege, then the user sees the request output.
- If a user can see a request, and if the request run as elevated user, the user is not able to see the request output.
- A request run-as user (elevated user, if specified) is able to see the request, request log, and request output.
- An Administrators group user (for example, "weblogic") is able to see all the requests and request logs.
- Administrators user is not able to see request output unless the requests were submitted and run as himself.

4.4.1 Oracle Simple Data Security Actions

You can use Enterprise Manager to create functional and data security policies for Oracle Enterprise Scheduler. There, you can associate actions with roles to create a policy.

[Table 4-7](#) lists available Simple Data Security actions.

Table 4-7 Grant Actions for Data Security

Action	Effect
ESS_REQUEST_READ	Read the request, get request state, and get details.
ESS_REQUEST_UPDATE	Update the request.
ESS_REQUEST_HOLD	Hold request execution.
ESS_REQUEST_CANCEL	Cancel a request execution.
ESS_REQUEST_LOCK	Lock a request.
ESS_REQUEST_RELEASE	Release the lock on a request.
ESS_REQUEST_DELETE	Delete a request.
ESS_REQUEST_PURGE	Purge a request.
ESS_REQUEST_OUTPUT_CREATE	Create the output of a request.
ESS_REQUEST_OUTPUT_READ	View the output of a request.
ESS_REQUEST_OUTPUT_DELETE	Delete the output of a request.
ESS_REQUEST_OUTPUT_UPDATE	Update the output of a request.

4.4.2 How to Create Data Security Policies for Oracle Enterprise Scheduler Components

You can use Enterprise Manager to create functional and data security policies for Oracle Enterprise Scheduler.

1. From the navigation pane, expand the **WebLogic Domain** folder and select the domain for which you're creating policies.
2. From the WebLogic Domain menu, select **Security** and then select **Application Policies**.

The Application Policies page displays.

3. In the Search section, from the **Application Stripe** dropdown, select the application stripe with which you want to work.
4. Click **Create** to begin granting permissions to certain users, groups, or application roles.

The Create Application Grant page appears.

5. In the Create Application Grant page, in the Grantee section, click **Add**.
6. In the Add Principal window, from the **Type** dropdown, select a type of principal, then enter a principal name or display name and click the search button to find the principal you want to add.
7. Under Search Principals, click the principal you want, then click **OK**.
8. In the Permissions section, click **Add**.
9. In the Search section, click **Permissions** or **Resource Types** depending on which you wish to search.
10. If you searched permissions, select **oracle.as.scheduler.security.RuntimeDataPermission** from the Permission Class dropdown. If you searched resource types, select **ESSRequestSimpleResourceType** from the Resource Type dropdown.
11. Click the **Search** button.
12. Under Search Results, select a resource, then click **Continue**.
13. In the Add Permission dialog, select the permission actions you want to grant.
14. Click **Select**.

Managing the Work of Oracle Enterprise Scheduler Jobs

This chapter describes how you can add metadata to refine Oracle Enterprise Scheduler job logic, including how to add work specifics, group jobs into meaningful sets, assign schedules that guide when the work occurs, and create exceptions, such as when jobs should *not* run simultaneously.

This chapter includes the following sections:

- [Introduction to Managing the Work of Oracle Enterprise Scheduler Jobs](#)
- [Managing Job Metadata](#)
- [Managing Work Assignments and Workshifts](#)
- [Managing Metadata Security for Jobs](#)

5.1 Introduction to Managing the Work of Oracle Enterprise Scheduler Jobs

You can define meaningful work by associating metadata with job types and by assigning schedules to the work.

When managing the work of Oracle Enterprise Scheduler jobs, you can:

- Associate request instance-specific data through properties -- whether your own or the system's -- that tie jobs to the context in which they are running
- Determine what level of access a group of people have to data related to the work done by jobs.
- Organize jobs into job sets so that they can be run as a unit, whether in sequence or in parallel.
- Specify that certain job definitions may not run at the same time, and the circumstances under which they cannot.
- Define the constraints within which jobs may run, based on factors such as system resources.

5.2 Managing Job Metadata

Oracle Enterprise Scheduler job metadata refers to the components that form the basis of scheduled job requests.

These include the following:

- **Job definition:** A job definition is the smallest unit of work performed in the context of the application that executes the job. A job definition is defined by a job type, such as a Java or SQL job type.
- **Job set:** A job set is a sequential or parallel set of job steps, where a job step can be a single job or another job set. A job set and each of its job set steps can have additional parameters, the values for which are provided when the job or job set is submitted as a job request.
- **Incompatibility:** An incompatibility allows you to specify job definitions and job sets that cannot run concurrently.

Note:

When you use Oracle Enterprise Manager Fusion Middleware Control or WLST commands to search for metadata, be sure to search for the actual name of the job definition, job set, or incompatibility rather than the display name. Display names are not found during searches.

This section contains the following topics:

- [Managing Job Definitions](#)
- [Managing Job Sets](#)
- [Managing Incompatibilities](#)

5.2.1 Managing Job Definitions

The Job Definitions page in Fusion Middleware Control allows you to view, create, edit, delete and search for job definitions.

This section contains the following topics:

- [Viewing Job Definitions](#)
- [Creating or Editing a Job Definition](#)
- [Deleting a Job Definition](#)

Note:

For a detailed description about how to use Oracle JDeveloper and Oracle Enterprise Manager Fusion Middleware Control to create job definitions for web service jobs, see *Developer's Guide for Oracle Enterprise Scheduler*, "Using Oracle Enterprise Manager Fusion Middleware Control or Oracle JDeveloper to Create a Job Definition."

5.2.1.1 Viewing Job Definitions

You can view the job definitions created for a given application. The table of job definitions displays details about the jobs related to an application such as the job definition name, the full path to which it is saved, the job type, and so on.

To display job definitions:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Job Metadata** and then select **Job Definitions**.

The Job Definitions page displays.

3. From the **Application** dropdown list, select the name of the application whose job definitions you want to view.

Note:

The application name is shown with its version number in parenthesis. The version number is not actually part of the application's name.

4. In the **Name** and **Package** fields, enter values to search for.
5. Click **Go** to search.

The job definitions for that application display in a table below the application dropdown list.

Column Name	Description
Name	This column displays the name of the job definition.
Display Name	This column contains a value that is displayed to users for this definition. This value may contain spaces and should be memorable for users who might later view or search for the definition in the user interface
Package	This column displays the name of the Java package associated with the job definition.
Job Type	This column displays the job type associated with the application.
Description	This column displays a description of the job definition.

6. To view the details of a specific job definition, click the name of the relevant job definition.

5.2.1.2 Creating or Editing a Job Definition

You can create or edit a job definition. With a job definition created, you can create a job request for a particular application. The job definition includes the directory path of the job to be run, the name of the application with which the job definition is associated and the job type used for the job definition. If the job is one of the web service types, the WSDL URL can also be specified.

Additional properties can be defined for a job definition as follows:

- **Parameters.** You can configure editable or read-only parameters to be submitted to the job request.
- **User properties.** You can configure properties to be filled in by end users at runtime, such as boolean, number and string values.

This section contains the following topics:

- [Configuring Application Defined Properties](#)
- [Configuring System Properties](#)
- [Configuring Access Control](#)

To create or edit a job definition:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Job Metadata** and then select **Job Definitions**.

The Job Definitions page displays.

3. From the **Applications** dropdown list, select the name of the application for which you want to create a job definition.
4. Click **Create** to create a new job definition, or **Edit** to modify an existing job definition.
5. For the job definition, configure the following properties:

- **Name** (required): Enter a name that uniquely identifies this definition. This value may not contain spaces.
- **Display Name**: Enter a value that is displayed to users for this definition. This value may contain spaces and should be memorable for users who might later view or search for the definition in the user interface.

This property supports internationalization for adapting to differing locales.

- **Package**: Enter the package of the job to be run.
- **Description**: Enter a description for the job.

This property supports internationalization for adapting to differing locales.

- **Job Type** (required): From the dropdown list, select the job type you want to use for the job definition. Select the **Override** check box to override the existing value when editing an existing job definition.

The Job Type you choose determines which one of the following fields is displayed beneath the Job Type. Note that you won't be able to edit the field's value if the corresponding value in this definition's job type was specified to be read only.

- **Class Name** (for JavaJobType, SyncEjbJobType, AsyncEjbJobType)

The class name that is associated with the job definition's job type. If you can edit the field, select **Override** and enter the class path and name or click Browse to specify a path name for the Java class.

- **Procedure Name** (for PlsqlJobType)

The procedure name for the PL/SQL procedure associated with this job definition's job type. If you can edit this field, select **Override** and enter the procedure name to be used by this job definition. You can also click the

Browse button to use an already defined database connection to browse for a stored procedure.

- **Process Name** (ProcessJobType)

The string that is the command associated with the job type. If you can edit this field, select **Override** and enter the command to be used by this job definition.

- **Select Web Service** button (for SyncWebserviceJobType, AsyncWebserviceJobType, OnewayWebserviceJobType)

The **Select Web Service** button invokes the Select Web Service dialog. Use the dialog to specify a WSDL file. After you select a valid WSDL, you can select the type of WSDL, and then the service, port and operation.

Note:

If you are editing an existing web service job definition, this button is named **Change Web Service** and you can use it to change the WSDL specification.

6. Optionally, configure application defined properties, system properties, and access control. For more information, see [Configuring Application Defined Properties](#), [Configuring System Properties](#), and [Configuring Access Control](#).

If you are creating a job definition for a secured web service job type, you must select the security policies for invocation and callback. See [Configuring Access Control](#) for more information.

7. Click **OK**.

5.2.1.2.1 Configuring Application Defined Properties

You use a parameter to pass data to the job request. In the job submission user interface, parameters can be passed using a number of display controls, such as a text box, date picker, choice list or list of values.

To configure parameters for a job definition:

1. On the Create Job Definition page, expand the **Application Defined Properties** section.

2. Click the **Add** button.

The Add Application Defined Property dialog box displays.

3. In the Add Application Defined Property dialog, enter the following information:

- **Type:** From the dropdown list, select a data type for the parameter: String, Long, Integer, DateTime, or Boolean. Required.
- **Name:** Enter a name for the parameter.
- **Initial Value:** Enter the value that should be used for this parameter until the user changes it.
- **Read Only:** Select the check box to have this parameter be read only to the user.

4. Click **OK**.

5.2.1.2.2 Configuring System Properties

System properties are parameter names that are known to and used by the system. You can specify that certain system properties should be used in your metadata.

Note:

If you modify the value of a system property, be sure that the modified value of the system property is correct. An incorrect system property value might cause the job definition to not work as expected.

To configure system properties for a job definition:

1. On the Create Job Definition page, expand the **System Properties** section.
2. Click the **Add** button.
The Add System Property dialog box displays.
3. From the **Name** dropdown list, select the system property you want to specify. Possible system properties are shown in [Table 5-1](#).
4. In the **Initial Value** text field, enter the value you want to assign to the system property.
5. Select the **Read Only** check box to have this property be read-only to the user.
6. Click **OK**.

Table 5-1 System Properties

System Property	Description
SYS_EXT_cmdLine.Unix	For a process job type, this specifies a string representing the command line command on Unix. If the SYS_cmdLine property is defined, this property is ignored.
SYS_EXT_cmdLine.Windows	For a process job type, this specifies a string representing the command line command on Windows. If the SYS_cmdLine property is defined, this property is ignored.
SYS_EXT_ejbOperationName [EJB_OPERATION_NAME]	Optional for EJB jobs. Specifies a pass-through parameter used by the EJB implementation to branch to the appropriate business methods. Example: manageFileAdapter
SYS_EXT_executableDir.Unix	For a process job, this specifies the directory where the process executable is located on Unix.
SYS_EXT_executableDir.Windows	For a process job, this specifies the directory where the process executable is located on Windows.
SYS_EXT_executableName	For a process job, this specifies the name of the process executable.
SYS_EXT_executableSuffix.Unix	Specifies the suffix for SYS_EXT_executableName on Unix. The default is ".sh".

Table 5-1 (Cont.) System Properties

System Property	Description
<code>SYS_EXT_executableSuffix.Windows</code>	Specifies the suffix for <code>SYS_EXT_executableName</code> on Windows. The default is ".cmd".
<code>SYS_EXT_executeAutoExport</code>	For process and Java job types, this specifies a value that prompts Oracle Enterprise Scheduler to automatically export any previously imported output files to the request's output directory at the start of the execute stage. Using this property assumes that the request sets the <code>SYS_EXT_supportOutputFiles</code> property to "output" and the <code>SYS_EXT_executeAutoExport</code> property to "true". This can be useful in the case that <code>RequestFileDirectory</code> is a local directory and the execute stage requires access to output files that were created in the pre-processor or during a previous execution attempt, in case of retry.
<code>SYS_EXT_fusionJob</code>	Set this to "true" to specify that the current job is Fusion Applications CP job type.
<code>SYS_EXT_invokeMessage</code>	The XML submit message used to invoke the web service.
<code>SYS_EXT_essAsyncRequestJndiMappedName</code> [ESS_ASYNC_REQUEST_JNDI_MAPPED_NAME]	For EJB jobs, specifies the JNDI mapped name of Oracle Enterprise Scheduler's <code>AsyncRequest</code> bean defined in the hosting application and bound to Oracle Enterprise Scheduler server's JNDI tree. This property is required only if: • The EJB invocation is asynchronous • You use a hosting application other than <code>EssNativeHostingApp</code> is used • The remote bean has to call back to Oracle Enterprise Scheduler beans (for example, an asynchronous callback).
<code>SYS_EXT_essJndiCsfKey</code> [ESS_JNDI_CSF_KEY_NAME]	Optional for EJB jobs. Specifies the CSF key alias of the authenticated Oracle Enterprise Scheduler server. This property is required only if the Oracle Enterprise Scheduler JNDI tree is authenticated. Example: <code>ess-jndi-csf-key</code>
<code>SYS_EXT_essMetadataJndiMappedName</code> [ESS_METADATA_JNDI_MAPPED_NAME]	For EJB jobs, specifies the JNDI mapped name of Oracle Enterprise Scheduler's <code>MetadataService</code> bean defined in the hosting application and bound to the Oracle Enterprise Scheduler server's JNDI tree. This property is required only if you use a hosting application other than <code>EssNativeHostingApp</code> and the remote bean requires access to Oracle Enterprise Scheduler's metadata bean
<code>SYS_EXT_essRuntimeJndiMappedName</code> [ESS_RUNTIME_JNDI_MAPPED_NAME]	For EJB jobs, specifies the JNDI mapped name of Oracle Enterprise Scheduler's <code>RuntimeService</code> bean that is defined in the hosting application and bound to the Oracle Enterprise Scheduler server's JNDI tree. This property is required only if you use a hosting application other than <code>EssNativeHostingApp</code> and the remote bean has to call the Oracle Enterprise Scheduler runtime bean (for example, to write output or log information, submit requests or operate on requests).

Table 5-1 (Cont.) System Properties

System Property	Description
SYS_EXT_jndiCSFKey [JNDI_CSF_KEY]	Required for EJB jobs only if the JNDI tree of the EJB server is secured. Points to the CSF alias that is mapped to the user name and password in the keystore. This specific user name/password pair is the credential required to access the secured JNDI for JndiMappedName lookup. This property can be added to either the RequestParameters object or to the Oracle Enterprise Scheduler configuration of the hosting application.
SYS_EXT_jndiProviderUrl [JNDI_PROVIDER_URL]	Optional for EJB jobs. Specifies the URL of the remote server. Required only if the EJB is remotely located.
SYS_EXT_jndiMappedName [JNDI_MAPPED_NAME]	Required for EJB jobs. Specifies the JNDI lookup name of a remote EJB implementation. Example: ejb/fileAdapter
SYS_EXT_processArguments	For a process job, this specifies the arguments.
SYS_EXT_requestLogLevel	Specifies the log level for Java and PL/SQL request loggers. The default is "INFO".
SYS_EXT_resolvedCmdLine	For a process job type, after the job is successfully started, this property provides a place for Oracle Enterprise Scheduler to save the actual command line (with substitution performed). This property is for diagnostic purposes only. It is for use by Oracle Enterprise Scheduler and other values might be overwritten by the service.
SYS_EXT_supportOutputFiles	Specifies a string indicating whether the job creates files in the file system. The following values are supported: • "work" to create a temporary file rather than an output file. • "output" to use the file system to create output files. • "none" (or no value) to not create any files in the file system.
SYS_EXT_uploadContentToRepository	Set this to "copy" to have Oracle Enterprise Scheduler upload request log and output files from its content store to a separate repository (UCM). The request log and output content remain in the content store, but are also available to user interfaces and tools that expect log/output in UCM.
SYS_EXT_useAlternateEnvironment	Set this to "true" to get environment properties from Fusion Applications-style files. The default is "false".
SYS_EXT_useExtendedSetup	Set this to "true" to have Oracle Enterprise Scheduler set up extended functionality, such as ApplSession, for each stage of the request if that functionality is available.

Table 5-1 (Cont.) System Properties

System Property	Description
SYS_EXT_userFileDirShared	For process and Java job types that may create files in the file system, this prompts Oracle Enterprise Scheduler to save in the request the configured RequestFileDirectory and RequestFileDirectoryShared from the ess-config.xml file. These values are saved as the properties SYS_userFileDir and SYS_EXT_userFileDirShared, respectively. Oracle Enterprise Scheduler uses these properties to create the request work and output directories, depending on SYS_EXT_supportOutputFiles. This for use by Oracle Enterprise Scheduler and other values might be overwritten by the service.
SYS_EXT_wsCancelMessage	Optional. The XML message for the web service cancel operation.
SYS_EXT_wsCancelOperation Name	Optional. The cancel operation name.
SYS_EXT_wsWsdLBaseUr l	The base URL part of WSDL URL.
SYS_EXT_wsWsdLUr l	The relative part of the web service WSDL URL (must be a concrete WSDL URL). Either the SYS_EXT_wsEndpointUr l property or the SYS_EXT_wsWsdLUr l property must be completely specified. For example, either wsWsdLBaseUr l and wsWsdLUr l are both configured, or wsEndpointBaseUr l and wsEndpointUr l are both configured.
SYS_EXT_wsEndpointBaseUr l	The base URL part of endpoint URL.
SYS_EXT_wsEndpointUr l	The endpoint URL for the web service. Either the SYS_EXT_wsEndpointUr l property or the SYS_EXT_wsWsdLUr l property must also be specified.
SYS_EXT_wsOperationName	The operation name.
SYS_EXT_wsPortName	The port name.
SYS_EXT_wsServiceName	The service name.
SYS_EXT_wsTargetNamespace	The target name space.
SYS_allowMultPending	Specifies whether multiple pending requests for the same job definition is allowed. This property has no meaning for a job set step. True or false.
SYS_application	Specifies the logical name of the J2EE application used for request processing. This property is automatically set by Oracle Enterprise Scheduler during request submission.
SYS_bizErrorExitCode	Specifies the process exit code for a process job request that denotes an execution business error. If this property is not specified, the system treats a process exit code of 4 as an execution business error. This property is optional for a process job type. It is not used for other job types. For more information about business errors, see Searching for a Job Request Using the Advanced Search Feature

Table 5-1 (Cont.) System Properties

System Property	Description
<code>SYS_className</code>	Specifies the Java executable for a Java job request. This should be the name of a Java class that implements the <code>oracle.as.scheduler.Executable</code> interface. This property is required for a Java job type. It is not used for other job types.
<code>SYS_cmdLine</code>	Specifies the command line used to invoke an external program for a process job request. This property is required for a Process job type. It is not used for other job types.
<code>SYS_effectiveApplication</code>	Specifies the logical name of the J2EE application that is the effective application used to process the request. A job definition, job type, or a job set step can be associated with a different application by defining the <code>EFFECTIVE_APPLICATION</code> system property. This property can only be specified using metadata and cannot be specified as a submission parameter.
<code>SYS_environmentVariables</code>	Specifies the environment variables to be set for the spawned process of a process job request. The property value should be a comma separated list of name value pairs (name=value) representing the environment variables to be set. This property is optional for a Process job type. It is not used for other job types.
<code>SYS_executePast</code>	Specifies whether instances of a repeating request with an execution time in the past should be generated. Instances are never generated before the requested start time or after the requested end time. To cause past instances to be generated, you must set this property to <code>TRUE</code> and specify the requested start time as the initial time from which instances should be generated. Note that a null requested start time defaults to the current time. Values for this property are: <code>TRUE</code>: All instances specified by a schedule are generated regardless of the time of generation. <code>FALSE</code>: Instances with a scheduled execution time in the past (that is, before the time of generation) are not generated. If this property is not specified, the system defaults to <code>TRUE</code> .
<code>SYS_extensionListener</code>	For internal use only.
<code>SYS_external_id</code>	Specifies an identifier for an external portion of an asynchronous Java job. For example, an asynchronous Java job usually invokes some remote process and then returns control to Oracle Enterprise Scheduler. This property can be used to identify the remote process. This property should be set by the job implementation of asynchronous Java jobs when the identifier is known. It is never set by Oracle Enterprise Scheduler.
<code>SYS_external_JobType</code>	Optional. The supported values are "ADFCB", "OSB" or "SOA". Any other value is invalid.

Table 5-1 (Cont.) System Properties

System Property	Description
<code>SYS_groupName</code>	Specifies the name of the Oracle Enterprise Scheduler isolation group to which this request is bound. This property is automatically set by Oracle Enterprise Scheduler during request submission.
<code>SYS_inputList</code>	Specifies input to a job request. The input to a serial job set is forwarded as input to the first step only. The input to a parallel job set is forwarded as input to all the parallel steps. Oracle Enterprise Scheduler imposes no format on the value of this property.
<code>SYS_inputWorkDirectory</code>	Specifies the working directory used during job request processing for input files. Oracle Enterprise Scheduler sets the value of this property during job request processing.
<code>SYS_listener</code>	Specifies the event listener class associated with the request. This should be the name of a Java class that implements the <code>oracle.as.scheduler.EventListener</code> interface.
<code>SYS_locale</code>	Specifies a language code to apply to the job.
<code>SYS_logicalClusterName</code>	Oracle Enterprise Scheduler provides the means by which EJB and web service jobs can define an abstract job location. The job location is specified by the Oracle Enterprise Scheduler <code>SYS_logicalClusterName</code> system property and specifies a logical cluster name (LCN). If the job definition for an EJB or web service job specifies a value for an LCN, certain environment-specific properties are specified using Oracle Enterprise Manager Fusion Middleware Control at the hosting application level rather than in the job definition. All job definitions with the same LCN share the value of the properties entered in the hosting application configuration properties using Oracle Enterprise Manager Fusion Middleware Control. In the Job Definitions > Job Definition Details page, set the <code>SYS_logicalClusterName</code> system property to a name that you use to define the set of location related properties (LCN) in the Configuration > Application Properties page. For more information, see Job Location Properties. Note that the terms "logical cluster" and "job location" can be used interchangeably.
<code>SYS_logWorkDirectory</code>	Specifies the logging working directory. This is for use by Oracle Enterprise Scheduler and other values might be overwritten by the service.
<code>SYS_outputList</code>	Specifies output from a request. The output of a serial job set is the <code>OUTPUT_LIST</code> of the last step. The output of a parallel job set is the concatenation of the <code>OUTPUT_LIST</code> of all the steps, in no guaranteed order, with <code>oracle.as.scheduler.SystemProperty.OUTPUT_LIST_DELIMITER</code> as a separator.

Table 5-1 (Cont.) System Properties

System Property	Description
<code>SYS_outputWorkDirectory</code>	Specifies the working directory used during job request processing for output files. Oracle Enterprise Scheduler sets the value of this property during job request processing. This is for use by Oracle Enterprise Scheduler and other values might be overwritten by the service.
<code>SYS_postProcess</code>	Specifies the post-process callout handler class. This should be the name of a Java class that implements the <code>oracle.as.scheduler.PostProcessHandler</code> interface.
<code>SYS_preProcess</code>	Specifies the pre-process callout handler class. This should be the name of a Java class that implements the <code>oracle.as.scheduler.PreProcessHandler</code> interface.
<code>SYS_priority</code>	Specifies the request processing priority. The priority interval is [0..9] with 0 as the lowest priority and 9 as the highest. Default: If this property is not specified, the system default value used is 4.
<code>SYS_procedureName</code>	Specifies the name of the PL/SQL stored procedure to be called for a SQL job request. The stored procedure should be specified using schema.name format. The property is required for a SQL job type. It is not used for other job types.
<code>SYS_product</code>	Specifies the product within the application that submitted the request.
<code>SYS_redirectedOutputFile</code>	Specifies the file where standard output and error streams are redirected for a process job request. This represents the full path of the log file where the standard output and error streams are redirected for the process when the request is executed. This is for use by Oracle Enterprise Scheduler and other values might be overwritten by the service.
<code>SYS_reprocessDelay</code>	Specifies the callout handler processing delay time. This represents the time, in minutes, to delay request processing when a delay is requested by a callback handler. Default: If this property is not specified, the system default used is 5. Integer type.

Table 5-1 (Cont.) System Properties

System Property	Description
<code>SYS_EXT_requestEffectiveEncoding</code>	Specifies the effective encoding associated with a process job request. <code>SpawnLauncher</code> determines the locale setting for a spawned job request in the following precedence order: 1. <code>LC_ALL/LANG</code> specified in the environment properties (<code>SystemProperty.ENVIRONMENT_VARIABLES</code>) for the request 2. <code>LC_ALL/LANG</code> specified in the hosting application's <code>ess-config.xml</code> file 3. WebLogic server <code>LC_ALL/LANG</code> The effective encoding is computed before the process is spawned and is stored in this property. This is later used to determine the encoding for the request log and output.
<code>SYS_request_timeout</code>	Specifies that job request can time out.
<code>SYS_requestCategory</code>	Specifies an application-specific label for a request. The label, defined by an application or system administrator, allows administrators to group job requests according to their own specific requirements.
<code>SYS_requestedProcessor</code>	Specifies the request processor node on which the request should be processed. This allows processor affinity to be specified for a job request. If this property is not specified, the request can run on any available request processor node. In general, this property should not be specified. If this property is specified for a request, the request processor's work assignments <code>oracle.as.scheduler.WorkAssignment</code> (specialization) must allow the execution of such requests, otherwise the request never executes. If the specified node is not running, the request remains in the <code>READY</code> state and does not execute until the node is restarted.
<code>SYS_requestExpiration</code>	Specifies the expiration time for a request. This represents the time, in minutes, that a request expires after its scheduled execution time. An expiration value of zero (0) means that the request never expires. If this property is not specified, the system default value used is 0. Request expiration only applies to requests that are waiting to run. If a request waits longer than the specified expiration period, it does not run. After a request starts running the request expiration no longer applies.
<code>SYS_retries</code>	Specifies the retry limit for a failed request. If request execution fails, the request retries up to the number of times specified by this property until the request succeeds. If retry limit is zero (0), a failed request is not retried. If this property is not specified, the system default used is 0.

Table 5-1 (Cont.) System Properties

System Property	Description
SYS_runasApplicationID	This property enables elevating access privileges for completing a scheduled job. Normally, a request runs as the submitting user. However, if this property is set in the metadata of the job associated with the request, then the request executes under the user identified by this property. This property can only be specified using metadata and cannot be specified as a submission parameter.
SYS_selectState	Specifies whether the result state of a job set step affects the eventual state of its parent job set. In order for the state of a job set step to be considered when determining the state of the job set, the <code>SELECT_STATE</code> must be set to <code>true</code>. If <code>SELECT_STATE</code> is not specified on a job set step, the state of the step is included in the determination of the state of the job set. Boolean type.
SYS_sqlJobClass	Specifies an Oracle Enterprise Scheduler job class to be assigned to the Oracle Enterprise Scheduler job used to execute a SQL job request. This property need not be specified unless the job used for a job request is associated with a particular Oracle Database resource consumer group or has affinity to a database service. If this property is not specified, a default Oracle Enterprise Scheduler job class is used for the job that executes the SQL request. That job class is associated with the default resource consumer group. It belongs to the default service, such that it has no service affinity and, in an Oracle RAC environment, any one of the database instances within the cluster might run the job. No additional privilege or grant is required for an Oracle Enterprise Scheduler SQL job request to use that default job class. This property is optional for a SQL job type. It is not used for other job types.
SYS_submittingApplication	Specifies the logical name of the J2EE application for the submitted (absolute parent) request. This property is automatically set by Oracle Enterprise Scheduler during request submission.
SYS_substitutionHandlers	This property has been deprecated.
SYS_successExitCode	Specifies the process exit code for a process job request that denotes an execution success. If this property is not specified the system treats a process exit code of 0 as execution success. This property is optional for a process job type. It is not used for other job types.

Table 5-1 (Cont.) System Properties

System Property	Description
SYS_userFileDir	Specifies a base directory in the file system where files, such as input and output files, may be stored for use by the request executable. Oracle Enterprise Scheduler supports a configuration parameter that specifies a file directory where requests may store files. At request submission, a USER_FILE_DIR property is automatically added for the request if the configuration parameter is currently set and USER_FILE_DIR property was not specified for the request. If the property is added, it is initialized to the value of the configuration parameter. The property is not added if the configuration parameter is not set at the time of request submission. This is for use by Oracle Enterprise Scheduler and other values might be overwritten by the service. Note that if this property is deleted, its value defaults to a system-defined value of <server log path>/ess_request/ and the value of the SYS_EXT_userFileDirShared property is assumed to be false.
SYS_userName	Specifies the name of the user used to execute the request. Normally, this is the submitting user unless the RUNAS_APPLICATIONID property was set in the job metadata. This property is automatically set by Oracle Enterprise Scheduler during request submission.
SYS_warningExitCode	Specifies the process exit code for a process job request that denotes an execution warning. If this property is not specified, the system treats a process exit code of 3 as execution warning. This property is optional for a process job type. It is not used for other job types.
SYS_workDirectoryRoot	Specifies the working directory for the process of a process job request. This is for use by Oracle Enterprise Scheduler and other values might be overwritten by the service.

5.2.1.2.3 Configuring Access Control

You can specify the level of access to this metadata that groups of users have. The roles visible here are defined in a `jazn-data.xml` file used by the application.

1. On the Create Job Definition page, expand the **Access Control** section.
2. Click the **Add** button.
The Add Access Control dialog box displays.
3. From the **Role** dropdown list, select the name of the role you want to apply to the job set. Users grouped by that role have the level of access you specify here.
4. Select the actions you want to allow role members to take: **Read, Execute, Update** and **Delete**.
5. Click **OK**.

5.2.1.2.4 Granting Permission to Job Roles in Enterprise Manager

This topic describes how to grant permission to job roles in Enterprise Manager

You can create a job definition in Enterprise Manager. When creating the job definition, you can grant permissions to a job and attach it to a role.

Before you grant permission to a job, you need to create an application role in *essnativehostingappstripe* in Enterprise Manager and associate a user or a group to this role in Enterprise Manager. You can do this using the security link in Weblogic Enterprise Manager.

Note: You can associate groups or individual users using the security link in Weblogic Enterprise Manager and not in ESS Enterprise Manager.

When you grant permissions to a job in Enterprise Manager and then create the job in ESS Enterprise Manager, the roles you created show up in Enterprise Manager. You can then attach the permissions to these roles.

To grant permission to job roles in Enterprise Manager:

1. In the Enterprise Manager window, navigate to **compact_domain** in the navigation pane.
2. In the Application Stripe dropdown, select **ESSNativeHostingApp**.
3. Click **Create** to specify the role for ESSNativeHostingApp.
4. In the **Role Name** field, enter **ESSJobRole** as the role name.
5. Click the **Add** icon to add the principal.
6. Search for the principal, test and click OK.
7. To select a role and specify the actions it can perform, go to the navigation panel and click **Scheduling Services > ESSAPP (AdminServer)**
8. Select the application for which you want to add the job definition.
9. You can use the filter criteria to look up the application.
10. To define a job type, open the job definition and click **Select Web Services**.
11. Add the access controls and click OK.

5.2.1.3 Deleting a Job Definition

You can delete a job definition.

To delete a job definition:

1. Search for the relevant job definition, as described in [Viewing Job Definitions](#).
2. In the Results table, select the job definition you want to delete and click the **Delete** button.

5.2.2 Managing Job Sets

You can view, create, edit, delete and search for job sets, which are collections of job requests that can be grouped together to run as a single unit. You can nest a job set so that it may contain a collection of job requests or one or more child job sets. Each job request or job set included within a job set is called a job set step.

Note:

To avoid creating a cyclical dependency, do not add a job set to a job set that contains it. For example, if JobSet2 contains JobSet1, be sure not to add JobSet2 to JobSet1.

You manage job sets with the Job Sets page in Fusion Middleware Control.

This section contains the following topics:

- [Viewing Job Sets](#)
- [Creating or Editing a Job Set](#)
- [Deleting a Job Set](#)

5.2.2.1 Viewing Job Sets

You can view the job sets created for a given application.

To display job sets:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Job Metadata** and then select **Job Sets**.

The Job Sets page displays.

3. From the **Applications** dropdown list, select the name of the application whose job sets you want to view.
4. Enter the name and package name of the job set you want to find, and click **Go**.

The job sets for the application you selected display in the **Results** table below the **Filter Criteria** section of the page.

Column Name	Description
Name	This column displays the name of the job set.
Display Name	The name displayed to users for this job set.
Package	This column displays the name of the Java package associated with the job set.
Execution Mode	This column displays the execution mode of the job set, Serial or Parallel.
Description	This column displays a description of the job set.

- To view the details of a specific job set, click the name of the job set you're interested in.

Figure 5-1 shows the Results table in the Job Sets page. The Results table displays job sets and their attributes, including name, package, execution mode and description.

Figure 5-1 The Job Sets Page and Results Table

Scheduling Service Home > Job Sets

Job Sets

Filter Criteria

Application: EssDemoApp

Name:

Package:

Go

Results

Create... Edit... Delete...

Name	Package	Execution Mode	Description
ShortAndLongJob	/mypackage	Serial	

Total Rows : 1

5.2.2.2 Creating or Editing a Job Set

You define a job set as either serial or parallel. At runtime, Oracle Enterprise Scheduler runs parallel job set steps together, in parallel. When a serial job set runs, on the other hand, Oracle Enterprise Scheduler runs the steps one after another in a specified sequence. With a serial job set, Oracle Enterprise Scheduler supports conditional branching between steps based on the execution status of a previous step.

For each step in a job set, you can configure an action to be taken on completion of the step, depending on the state of the step. You can also configure parameters and system properties for the job set, such as elevating access privileges to complete a particular job request or specifying the number of times a job can be retried in the event of failure.

To create or edit a job set:

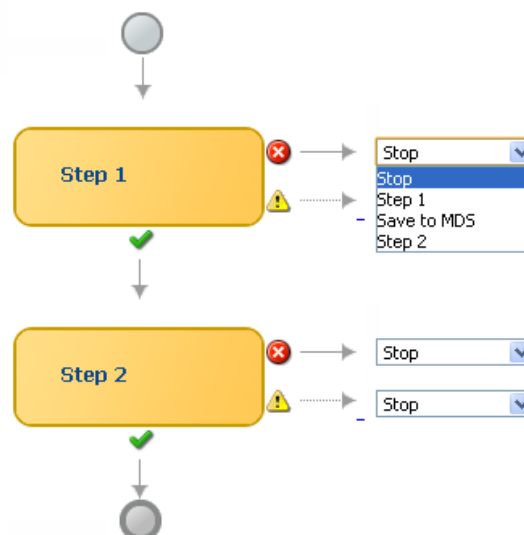
- From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
- From the Scheduling Services menu, select **Job Metadata** and then select **Job Sets**. The Job Sets page displays.
- From the **Applications** dropdown list, select the name of the application for which you want to create or edit job sets.
- Click **Create** to define a new job set, or **Edit** to modify an existing job set.
- In the **Name** field, enter the name of the job set. Fill in the **Display Name** field with a name that is shown to users. Optionally, add a description in the

Description text field, and the name of the relevant job set Java package in the **Package** text field.

6. In the Job Set Steps section, select **Serial** or **Parallel** to create a serial or parallel job set.
7. Add steps as required by clicking the **Add** button. Define each step as required.
 - a. In the Step tab, in the **Step ID** field, enter a meaningful ID for the step. The step ID must be unique for all steps in this job set.
 In the **Job** field, click the search button. In the window that displays, select the job or job set that you want to use for this step, then click **OK**.
 - b. Select **Insert into main diagram** or **Add to list of available steps**. If you choose to add the step to the list of available steps, use the dropdown lists that display to select actions for the possible job outcomes, namely **On Success**, **On Error** and **On Warning**.
 - c. In the Application Defined Properties tab, click the **Add** button to define any required parameters and enter their initial value in the field provided. For more information about defining parameters, see [Configuring Application Defined Properties](#).
 - d. In the System Properties tab, click the **Add** button to select system properties and enter their initial value in the field provided. For more information about selecting system properties, see [Configuring System Properties](#).
 - e. Click **OK**.
 - f. If you configured the step as a serial step, it displays in the job set flow diagram. Configure the action to be taken upon reaching error and warning states, respectively. From the dropdown list for the error and warning icons, select **Stop** or the name of the job definition to run upon reaching an error or warning state.

Job Set Steps

☒ Serial ☐ Parallel



8. Continue to define steps as required for the job set.
9. If your job set requires them, define application defined properties and system properties in sections toward the bottom of the job set window.
10. Configure access control for the job set. For more on defining access control, see [Configuring Access Control](#).
11. On the Create Job Set page, click **OK** to save the job set.

5.2.2.3 Deleting a Job Set

You can delete a job set.

To delete a job set:

1. Search for the relevant job set, as described in [Viewing Job Sets](#).
2. In the Results table, select the job set you want to delete and click the **Delete** button.

5.2.3 Managing Incompatibilities

An Oracle Enterprise Scheduler incompatibility specifies which job definitions should not run simultaneously with one another.

An incompatibility defines either a global incompatibility or a domain-specific, property-based incompatibility.

This section contains the following topics:

- [Viewing Incompatibilities](#)
- [Creating or Editing an Incompatibility](#)
- [Deleting an Incompatibility](#)

5.2.3.1 Viewing Incompatibilities

The Incompatibilities page displays an incompatibility's name, the Java package associated it, and a description for the incompatibility.

To view job incompatibilities:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Job Metadata** and then select **Incompatibilities**.

The Incompatibilities page displays.

3. From the **Application** dropdown list, select the name of the application whose incompatibilities you want to view.
4. Enter the name and package name of the incompatibilities you want to find, and click **Go**.

The incompatibilities for that application display in a table below the application dropdown list.

Column Name	Description
Name	This column displays the name of the incompatibility.
Display Name	A name displayed to users for this incompatibility.
Package	This column displays the name of the Java package associated with the incompatibility.
Incompatibility Type	This column displays the type of incompatibility, global or domain specific.
Description	This column displays a description of the incompatibility.

5. To view the details of a specific incompatibility, click the incompatibility's name.

5.2.3.2 Creating or Editing an Incompatibility

An incompatibility consists of one or more job definitions configured as incompatible, and optionally the resource over which they must be incompatible. A resource is not specified for a global incompatibility.

In a domain incompatibility, the resource is represented by the property name that forms the Incompatibility. The property name might be different across job definitions. For example, if two job definitions, JobA and JobB, are made incompatible, then the property identified for incompatibility might have different names in JobA and JobB.

The following gives more information about the difference between domain and global incompatibilities:

- **Domain-specific:** In a domain incompatibility, Oracle Enterprise Scheduler ensures that requests for the incompatible jobs or job sets, with a property that has the same property value, do not run at the same time.

You add two or more job definitions as incompatible within the scope of a resource, where the resource is identified by a system property name or a parameter name. Each entity can use the same property name or a different property name. For a property-based incompatibility, you must specify a job definition or a job set and a property name for each entity.

- **Global:** In a global incompatibility, Oracle Enterprise Scheduler ensures that requests for the incompatible jobs or job sets do not run at the same time.

Two or more job definitions cannot run together at any time. A global incompatibility is used when the incompatible entities are not to be run at the same time regardless of any property. For global incompatibility, only the job definition or job set is specified for each entity.

Usually an incompatibility is defined using two or more job or job set entities. An incompatibility can be defined for one entity as long as that entity is marked as self-incompatible. A self-incompatibility implies that multiple job requests associated with that job or job set cannot run together.

Defining an incompatibility involves the following:

- **Package and scope:** Select a relevant Java package to use for the incompatibility and set the scope for the incompatibility (global or domain only).
- **Jobs:** Select the jobs that are incompatible.

- Parameters and properties: Define parameters and properties as required.
- Access control: Define access control as required.

To create or edit an incompatibility:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Job Metadata** and then select **Incompatibilities**.

The Incompatibilities page displays.

3. From the **Application** dropdown list, select the name of the application whose incompatibilities you want to create or edit.
4. Click **Create** to define a new incompatibility, or **Edit** to modify an existing incompatibility.

The Create (or Edit) Incompatibilities page displays.

5. Enter the following information:
 - **Name:** Enter a name for the incompatibility.
 - **Display Name:** A name displayed to users for this job set.
 - **Package:** Enter the name of the relevant incompatibility Java package.
 - **Description:** Optionally, add descriptive text for the incompatibility.
 - **Type:** Configure the scope of the incompatibility by selecting **Global** or **Domain**.
6. In the Entities section, click the **Add** button to add jobs to the incompatibility.

The Add Entity dialog box displays. Select one or more jobs and click **OK**.
7. If this is a domain incompatibility, select each entity listed in the Entities table, click the **Edit** button, and select the property whose value should be incompatible with the other entities in the list.
8. If entities in the list are self incompatible -- meaning that they should not run simultaneously with other instances of the same entity -- edit the entity and select the Self Incompatible check box.
9. Configure access control for the incompatibility, as described in [Configuring Access Control](#).
10. Click **OK** to save the incompatibility.

5.2.3.3 Deleting an Incompatibility

Deleting an incompatibility results in the incompatible job requests or job sets becoming compatible again.

To delete an incompatibility:

1. Search for the relevant incompatibility, as described in [Viewing Incompatibilities](#).

2. In the Results table, select the incompatibility you want to delete and click the **Delete** button.

5.2.4 Bulk Customization of Job Metadata

Typically, customization of existing Oracle Enterprise Scheduler jobs involves editing job properties using Oracle Enterprise Manager Fusion Middleware Control or Oracle JDeveloper. This approach is limited and repetitive for customers interested in bulk updates of job properties. The System MBean Browser and the bulk customization APIs enable you to customize the property of a piece of metadata in bulk for a number of jobs without changing the base metadata.

The following properties can be updated or customized in bulk:

- `oracle.as.scheduler.SystemProperty.PRIORITY`
- `oracle.as.scheduler.SystemProperty.RETRIES`
- `oracle.as.scheduler.SystemProperty.REQUEST_CATEGORY`
- `oracle.as.scheduler.SystemProperty.ASYNC_REQUEST_TIMEOUT`
- `oracle.as.scheduler.SystemProperty.LOGICAL_CLUSTER_NAME`
- `enableTrace`
- `enableTimeStatistics`

5.2.4.1 Bulk Customization Using the System MBean Browser

The bulk customization API can be conveniently invoked from the Oracle Enterprise Manager Fusion Middleware Control System MBean Browser. The process has two steps:

1. Export the list of job definition properties that can be customized to an XML file. This step queries the job definitions that match the specified filter criteria in the Oracle Enterprise Scheduler hosting application MDS repository. The generated XML output file can be edited. This XML file serves as the input for the customization step.
2. Customize the job definition property metadata definitions contained in the previously exported XML file. This step customizes the job definition property metadata definitions defined in the input XML file generated in the export step without changing the base metadata. This process makes use of the MDS metadata customization functionality, and an exception is thrown if the properties are not members of the one of the `MetadataService.CustomizableProperty` enumerations. This step modifies the base metadata for job definition properties that belong to the custom name space specified in the `<cust-config>` element of the `adf-config.xml` file.

Note:

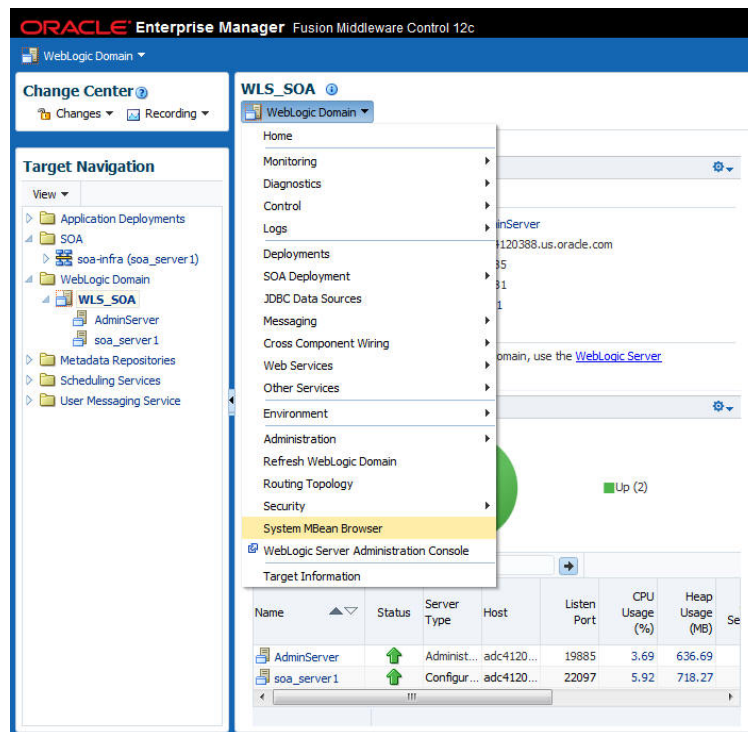
In order for this feature to work correctly, the application must define a customization class in its `adf-config.xml` file. Refer to the MDS documentation for more information.

5.2.4.1.1 Example

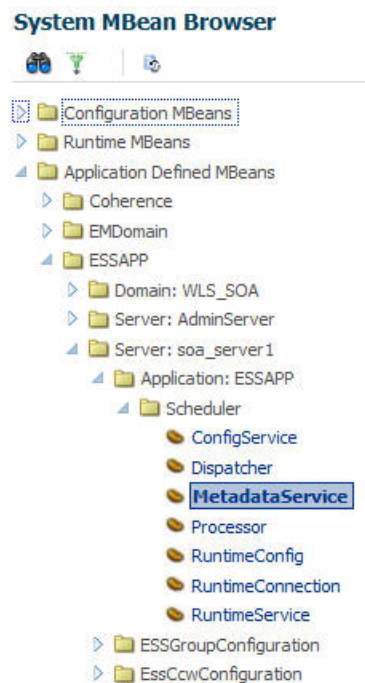
The following example shows how to use the System MBean Browser to export and customize job definition properties.

1. Start the Oracle Enterprise Manager Fusion Middleware Control System MBean Browser by choosing it from the **WebLogic Domain** menu as shown in [Figure 5-2](#).

Figure 5-2 Start the Oracle Enterprise Manager Fusion Middleware Control System MBean Browser



2. In the System MBean Browser, navigate to your application's MetadataService entry. [Figure 5-3](#) shows an example. Note that your server and application names may be different.

Figure 5-3 Navigation Path to MetadataService

3. Click on the **Operations** tab as shown in Figure 5-4.

Figure 5-4 Choose the exportESSJobMetadataBulk Item

The screenshot shows the 'System MBean Browser' with the 'Operations' tab selected for the 'Scheduler:MetadataService' MBean. The left pane shows the same navigation path as Figure 5-3. The right pane displays a table of operations. The operation 'exportESSJobMetadataBulk' is highlighted with a red box.

Name	Description	Parameters	Return Type
27 deleteJobType	Deletes a JobType definition from scheduler metadata repository	2	void
28 deleteScheduleDefinition	Deletes a Schedule definition from scheduler metadata repository	2	void
29 deleteTrigger	Deletes a Trigger definition from scheduler metadata repository	2	void
30 deleteWorkAssignment	Deletes a WorkAssignment definition from scheduler metadata repository	1	void
31 deleteWorkshift	Deletes a Workshift definition from scheduler metadata repository	1	void
32 exportESSJobMetadataBulk	Export the list of job metadata properties to a XML file for bulk customization	5	void
33 getCustomMDSNamespace	Operation exposed for management	1	Array of java.lang.String
34 getExclusionDefinition	Retrieves a ExclusionsDefinition from the scheduler metadata repository	2	javax.management.openmbean.CompositeData
35 getIncompatibilities	Retrieves Incompatibility definitions for a given entity	2	Array of javax.management.openmbean.Composite...
36 getIncompatibility	Retrieves an Incompatibility definition from the scheduler metadata repository	2	javax.management.openmbean.CompositeData
37 getJobDefinition	Retrieves a JobDefinition from the scheduler metadata repository	2	javax.management.openmbean.CompositeData
38 getJobPolicySubjectResourcePatterns	Retrieve a webservice job OWSM policy subject resource pattern	3	Array of java.lang.String
39 getJobSet	Retrieves a Job Set definition from the scheduler metadata repository	2	javax.management.openmbean.CompositeData

4. Click on the **exportESSJobMetadataBulk** item (Figure 5-4) to invoke the Operation:exportESSJobMetadatBulk page.
5. On the Operation:exportESSJobMetadatBulk page, fill in the fields in the **Value** column using the parameters listed in Table 5-2 and then click **Invoke**. Figure 5-5 shows an example.

Figure 5-5 Operation: exportESSJobMetadataBulk Page

WLS_SOA
WebLogic Domain

Page Refreshed Nov 12, 2013 4:49:20 PM PST

/Domain_WLS_SOA/WLS_SOA > System MBean Browser

System MBean Browser

Operation: exportESSJobMetadataBulk [Invoke] [Revert] [Return]

MBean Name: ESSAPP:Name=MetadataService,Location=soa_server1,type=Scheduler,Application=ESSAPP
 Operation Name: exportESSJobMetadataBulk
 Description: Export the list of job metadata properties to a XML file for bulk customization
 Return Type: void

Name	Description	Type	Value
xmlOutputFile	Absolute path to the output XML file where the list of job definition properties for bulk customization is to be exported	java.lang.String	/scratch/ame1/tmp/bulk-ess1.xml
filterQueryField	Name of the query field to filter by for generating bulk customization xml list	java.lang.String	Package
filterComparator	Name of the filter comparison comparator for generating bulk customization xml list	java.lang.String	CONTAINS
filterValue	Value to compare the filter query field with for generating bulk customization xml list	java.lang.String	oracle/aps/ess
application	Name of the ESS hosting application whose metadata repository is to be used	java.lang.String	EssDemoApp

Return Value

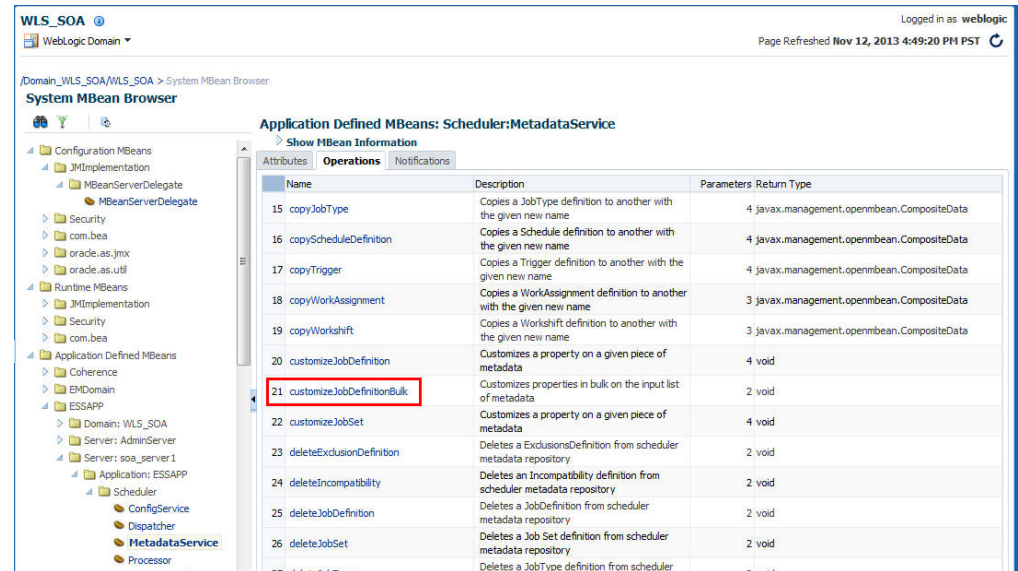
The following tables show additional examples of parameters and values that might be used in the Operation: exportESSJobMetadataBulk page.

Parameter	Value
filterQueryField	package
filterComparator	CONTAINS
filterValue	oracle/as/ess/customer-pkg

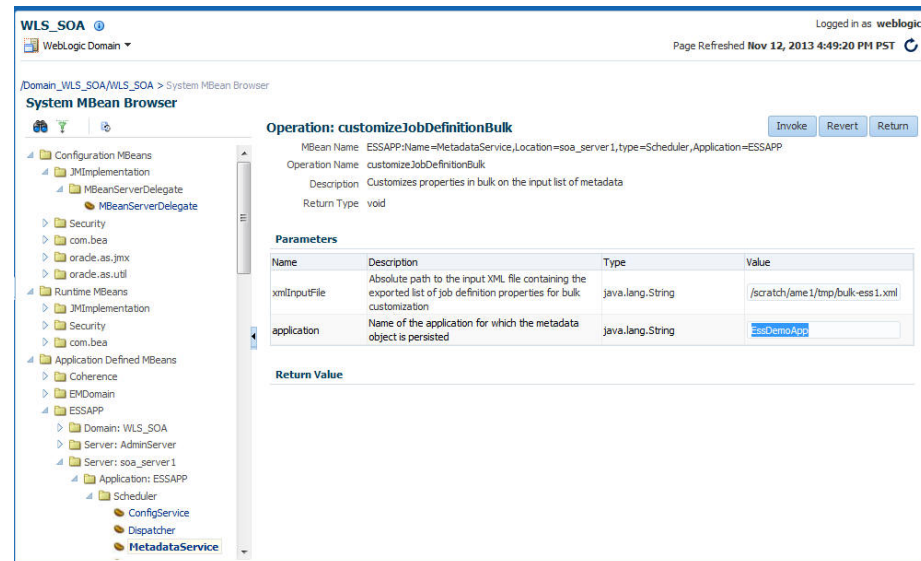
Parameter	Value
filterQueryField	job-type
filterComparator	ENDS_WITH
filterValue	PlsqlJobType

Parameter	Value
filterQueryField	name
filterComparator	EQUALS
filterValue	some_env_value

- Click **Return** to return to the MetadataService page and then click on the **customizeJobDefinitionBulk** item to invoke the Operation:customizeJobDefinitionBulk page.

Figure 5-6 Choose the customizeJobDefinitionBulk Item

- Fill in the fields in the **Value** column as shown in Figure 5-7. The `xmlInputFile` entry is the name of the file to which you previously exported the parameters. Click the **Invoke** button to complete the customization.

Figure 5-7 The Operation: customizeJobDefinition Page

5.2.4.2 Bulk Customization MBean API

The bulk customization API is comprised of two methods, `exportESSJobMetadataBulk` and `customizeJobDefinitionBulk`.

5.2.4.2.1 exportESSJobMetadataBulk Method

This method exports the list of job definition properties that can be customized. This method queries the job definitions that match the specified filter criteria in the Oracle Enterprise Scheduler hosting application MDS repository.

The generated XML output file can be edited by an administrator. This file serves as the input to the `customizeJobDefinitionBulk` method described in [customizeJobDefinitionBulk Method](#).

5.2.4.2.1.1 Syntax

```
void exportESSJobMetadataBulk (java.lang.String xmlOutputFile,
                              java.lang.String filterQueryField,
                              java.lang.String filterComparator,
                              java.lang.String filterValue,
                              java.lang.String application)
    throws java.io.IOException,
           javax.management.RuntimeOperationsException
```

Table 5-2 *exportESSJobMetadataBulk Parameters*

Parameter	Description
<code>xmlOutputFile</code>	The absolute path of the output XML file to which the list of the job definitions' customizable properties is exported.
<code>filterQueryField</code>	The name of the query field by which to filter. The following are the supported query fields: - Package - Name - JobType - ExecutionType These query fields are defined in the package <code>oracle/as/scheduler/metadata/query/MetadataQueryField</code> .
<code>filterQueryComparator</code>	The name of the filter comparison operator. The following are the supported comparators: - EQUALS - NOT_EQUALS - CONTAINS - NOT_CONTAINS - STARTS_WITH - ENDS_WITH These comparators are defined in the package <code>/oracle/as/scheduler/Filter.Comparator</code> .
<code>filterValue</code>	The value with which to compare the filter query field.
<code>application</code>	The name of the Oracle Enterprise Scheduler hosting application whose metadata repository is to be used.

Table 5-3 *exportESSJobMetadataBulk Exceptions*

Exception	Description
<code>java.io.IOException</code>	Thrown if a protocol error occurs.
<code>javax.management.RuntimeOperationsException</code>	Thrown if the metadata sub-system encounters failures.

5.2.4.2.2 customizeJobDefinitionBulk Method

This method customizes the job definition property metadata definitions defined in the input XML file generated by the `exportESSJobMetadataBulk` method without changing the base metadata. The MDS metadata customization functionality is used, and an exception is thrown if the properties are not members of the one of the `MetadataService.CustomizableProperty` enumerations.

This method modifies the base metadata for job definition properties that belong to the custom name space `/oracle/apps/ess/custom`.

Note:

In order for this feature to work correctly, the application must define a customization class in its `adf-config.xml` file. Refer to the MDS documentation for more information.

5.2.4.2.2.1 Syntax

```
void customizeJobDefinitionBulk (java.lang.String xmlInputFile,
                                java.lang.String application)
                                throws java.io.IOException,
                                       javax.management.InstanceNotFoundException,
                                       javax.management.OperationsException,
                                       javax.management.InvalidAttributeValueException,
                                       javax.management.RuntimeOperationsException
```

Table 5-4 customizeJobDefinitionBulk Parameters

Parameter	Description
<code>xmlInputFile</code>	The path to the input XML file that contains the exported list of job definitions. The format of the file is dictated by the output of the <code>exportESSJobMetadataBulk</code> method.
<code>application</code>	The name of the application whose metadata repository is to be used.

Table 5-5 customizeJobDefinitionBulk Exceptions

Exception	Description
<code>java.io.IOException</code>	Thrown if a protocol error occurs.
<code>javax.management.InvalidAttributeValueException</code>	Thrown if the validation of the definition fails.
<code>javax.management.RuntimeOperationsException</code>	Thrown if the metadata sub-system encounters failures.
<code>javax.management.InstanceNotFoundException</code>	Thrown if the specified MBean does not exist in the repository.
<code>javax.management.OperationsException</code>	Represents exceptions thrown in the MBean server when performing operations on MBeans.

5.3 Managing Work Assignments and Workshifts

With work allocation, you can define constraints on where and when jobs run, as well as the amount of resources that can be used to process the jobs. This process includes creating a work assignment and binding the work assignment to a request processor.

A work assignment consists of a specialization rule and one or more workshifts. The specialization rule defines restrictions on which jobs can be processed. A workshift defines the temporal windows when the jobs can be processed and what resources can be used during those periods. The resources defined in a workshift include threads, which are a local resource of the request processor, and asynchronous workers, a global resource. The number of asynchronous workers can be specified to throttle the use of a shared global resource, such as database jobs.

Binding associates a work assignment with a request processor on a server, determining where the jobs can run. An exclusive binding mode is supported to not only determine when the jobs can run but to prevent them from running anywhere else.

By default, no work assignments are bound. When there is no bound or active work assignment, a virtual default work assignment is started to process all jobs using all available resources. This default work assignment is limited to the number of threads configured for the processor, but there is no default limit for asynchronous workers. To limit asynchronous workers, it is necessary to create and bind one or more work assignments.

This section contains the following topics:

- [Managing Work Assignments](#)
- [Managing Workshifts](#)
- [Managing Schedules](#)

5.3.1 Managing Work Assignments

Work assignments provide the following controls for processing job requests:

- At runtime, work assignments can be bound to a request processor and can limit the period during which job requests of a certain type can be processed.
- At runtime, the request processor can be configured to limit the threads that are available to process all job requests.

5.3.1.1 Determining Active Work Assignments

A bound work assignment is active if it is enabled, has an active workshift and the enabled flag is set on the work assignment definition. A workshift is active if it has an allocation greater than zero and includes a current schedule (where the current time is within a time window defined by the schedule and duration), or the workshift is a 24x7 workshift. If there are no bound work assignments on the server, the default work assignment is active using all threads and having no limits on asynchronous workers.

5.3.1.2 Determining Work Assignment Thread Allocation

An active work assignment is assigned the thread allocation specified by the active workshift unless the total number of threads required to satisfy the allocations of all active work assignments exceeds the configured thread count. In this case, Oracle

Enterprise Scheduler weighs the thread allocation against the percentage of threads allotted to the workshift out of the total number of thread allocations for all work assignments.

For example, suppose work assignment 1 has a thread allocation of 70, work assignment 2 has a thread allocation of 30, and there are 20 processor threads configured. The total desired allocation is 100, so the weight for work assignment 1 is 70 percent while the weight for work assignment 2 is 30 percent. Oracle Enterprise Scheduler allocates 14 threads to work assignment 1 and 6 threads to work assignment 2.

If the default work assignment is active, the number of threads allocated to the work assignment is equal to the configured thread count.

Note:

Each active work assignment is assigned at least one thread.

5.3.1.3 Processing Active Work Assignments

After determining work assignments and thread allocations, Oracle Enterprise Scheduler begins a thread pool for each active work assignment. The threads are responsible for processing job requests that are specialized to the work assignment, except for requests that are specialized to an exclusive work assignment. The exclusion is effective for all work assignments, including the default work assignment. If an exclusive work assignment is bound to any server in the group, no other work assignment can process job requests specialized to that exclusive work assignment.

Note:

All work assignments bound in exclusive mode are excluded, including disabled work assignments. Exclusive bindings apply even if the server to which they are bound is unavailable. You must unbind an exclusive work assignment if you do not want it to be excluded.

5.3.1.4 Creating or Editing a Work Assignment

A work assignment includes two primary components that define request processor constraints:

- **Specialization rules:** Define restrictions for job request processing on a request processor.
- **Workshifts:** Specify temporal windows for processing job requests and thus describe the schedule for when job requests can be processed on a request processor

This combination of work assignment controls, including specialization rules and workshifts gives you the ability to select the kinds of job requests to be processed and decide how to allocate the request processor resources. For example, you can define two workshifts: a day shift and a night shift to allocate processing for these periods. The day shift could have more resources allocated for a peak usage period while the night shift could provide a different mix for its resource allocation.

By default, no work assignments are bound to a request processor and the request processor processes any ready job request. The default behavior is similar to using a

request processor with no specialization rules and a workshift of 24/7 duration, all configured threads are used and there are no limits on the number of asynchronous jobs.

Table 5-6 shows the properties that you can define for specialization rules.

Table 5-6 Specialization Properties Available for Specialization Rules

Specialization Property	Description
Application	Specifies the name of the application associated with a job request.
Product	Specifies the name of the product within an application.
Submitted By	Specifies a user who submitted a job request.
Job Definition	Specifies a specific job request name.
Request Category	Specifies labels defined by the system administrator allowing administrators to group job requests for specific requirements. You can include multiple labels that are separated by a delimiting character.
Logical Cluster Name	Specifies the name of the logical cluster.

The following operators are used for creating conditions: `equals`, `not equals`, `contains`, `starts with`, `ends with` and `NOT` (unary).

The following operators can be used to join the conditions of a rule: `AND` and `OR` (both of type binary).

Example 5-1 shows sample specialization rules that can be used in a work assignment.

If a job request is specialized to two different work assignments, either of those work assignments may process the job request depending on resource availability. Similarly, if the same work assignment is bound to two different servers, either server may process the job request. In fact, different stages of the same request may be processed on different servers, such as pre-processing on one server and job execution on another server.

Note that the `requestCategory` value can be several label terms separated by a comma. In this way you can specify labels that mimic the behavior of tags, with specializations based on combinations of labels.

For example, you could define the following labels:

Assuming the preceding labels, you could write a `requestCategory` expression such as the following examples:

If you use commas to delimit terms, you must enclose every separate label term with the comma on both sides, as shown in Example 5-2. Your `requestCategory` expression must specify the full term including the commas, as shown in Example 5-3, to ensure that the expression does not select another term that contains the one you specify.

To create or edit a work assignment:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Work Allocation** and then select **Work Assignments**.

The Work Assignments page displays.

3. Click the **Create** or **Edit** buttons to create or edit a work assignment.

The Create (or Edit) Work Assignment page displays.

4. Enter a name and description for the work assignment in the relevant text fields.
5. Select the **Enabled** check box to enable the work assignment.
6. Click **Create Specialization** to add conditions under which the work assignment runs.
7. In the Workshifts section, click the **Add** button to add one or more workshifts to the work assignment.
8. Click **OK** to save the work assignment.

Example 5-1 Sample Specialization Rules

```
application = 'EssDemoApp' AND (definition = 'JobDefinition://mypackage/
Job_essdemo1' OR definition = 'JobDefinition://mypackage/LongRunningJob')

requestCategory = 'Priority'

user = 'sam'
(requestCategory = 'LongRunning') AND NOT (definition = 'JobDefinition://mypackage/
LongRunningJob')
```

Example 5-2 Request Category Labels That Combine Terms

```
",NODE1,CRITICAL,"
",NODE1,STANDARD,"
",NODE1,IMPORTANT,"
```

Example 5-3 requestCategory Expressions That Use Multiple Combined Labels

```
(requestCategory contains ',NODE1,') AND (requestCategory contains ',CRITICAL,')

(requestCategory contains ',STANDARD,') OR (requestCategory contains ',IMPORTANT,')
OR (requestCategory contains ',CRITICAL,')
```

5.3.1.5 Deleting a Work Assignment

You can delete a work assignment from the list of work assignments. Before deleting a work assignment, make sure that it is not bound to a request processor.

To delete a work assignment:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Work Allocation** and then select **Work Assignments**.

The Work Assignments page displays.

3. Select the work assignment you want to delete and click **Delete**.

5.3.2 Managing Workshifts

A workshift indicates the operating, active times for a request processor. Specifically, a workshift defines a sequence of temporal windows in which resources or threads, are made available for processing job requests. When a work assignment is bound to a request processor, one or more workshifts are associated with the work assignment. At runtime, Oracle Enterprise Scheduler determines the resource allocation for the workshifts within the work assignment.

You can use different amounts of resources during different periods of time by using multiple workshifts, each with a different time window and resource limits. For example, you can create daytime workshift that starts at 8 a.m. and has your daytime thread allocation and async job limits; and you can create a nighttime workshift that starts at 6 p.m. and has your nighttime thread allocation and async job limits.

Note that workshifts may overlap, but Oracle Enterprise Scheduler chooses one workshift as the current workshift.

While there is at most one currently active workshift for a given work assignment, there may still be active requests for a workshift that is no longer active. This can happen if a request runs beyond the defined operating window for the workshift it started running in. For example, if a request started running while the daytime workshift was active and that request is still running when the nighttime workshift starts. Oracle Enterprise Scheduler uses the limits of the currently active workshift and accounts for all active requests that started during any workshift in the work assignment. If the number of active requests exceeds the current limit, no more requests are run until the number of active requests drops below the limit.

A workshift defines the following resources:

- Thread allocation
- Asynchronous job limits for asynchronous Java and PL/SQL jobs

Thread allocation specifies the number of threads that can be used by the request processor. These threads are used to perform local tasks, including processing synchronous jobs, initiating and finalizing asynchronous jobs, pre- and post-processing of job requests and updating events. When the workshift in a work assignment is active, each request processor for the work assignment can use the specified number of threads. For example, suppose a work assignment includes a 24x7 workshift with a thread allocation of 15. If that work assignment were bound to three request processors, each request processor could use 15 threads, for a total of 45 threads across all three servers.

Asynchronous, PL/SQL, and asynchronous Java jobs use a global resource that must be shared across the entire system. The workshift can specify limits on the number of PL/SQL jobs and asynchronous Java jobs that can be active for the work assignment. This limit applies across all request processors for the work assignment in the entire system. For example, suppose a work assignment includes a 24x7 workshift with a PL/SQL job limit of ten. If that work assignment were bound to three request processors, all three request processors share the ten PL/SQL asynchronous workers, for a maximum of ten PL/SQL jobs active for that work assignment.

An asynchronous job reserves and holds an asynchronous worker before pre-processing starts and through finalization. In the case of pre-process delay, the async worker is released during the delay and reserved when the delay ends. A request that submits a subrequest and pauses, releases its async worker and the async worker is

reserved when the request is resumed. A BLOCKED request does not hold an async worker.

When deciding on thread allocation and asynchronous job limits for a workshift, take note of the types of jobs specialized to the work assignment where the workshift is to be used.

This section contains the following topics:

- [Creating or Editing a Workshift](#)
- [Deleting a Workshift](#)

5.3.2.1 Creating or Editing a Workshift

Creating a workshift involves the following:

- **Schedule:** Associate a schedule with the workshift.
- **Duration:** Enter a duration for the workshift.
- **Thread allocation:** Specify the number of threads you want to allocate to the workshift.
- **Asynchronous job limits:** Specify limits on the number of asynchronous Java jobs and PL/SQL jobs that can be active for the work assignment.

To create or edit a workshift:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Work Allocation** and then select **Workshifts**.

The Workshifts page displays.

3. Click the **Create** or **Edit** buttons to create or edit a workshift.

The Create (or Edit) Workshift page displays.

4. Enter the following information for the workshift.
 - **Schedule Name:** Enter a name for the workshift.
 - **Schedule Description:** Enter a description for the workshift.
5. In the Active Period section, select one of the following activation periods for the workshift.
 - **Active 24x7:** Select to always enable the workshift. Selecting this option disables the **Duration** text field.
 - **Use existing schedule:** Select to enable the workshift using a schedule you created previously. Click the **Browse** button to display the Select Schedule window and search for the schedule using the **Name** and **Package** fields.
In the **Duration** text field, enter the duration of the workshift in minutes.
 - **Specify schedule:** Select to create a schedule for the workshift. Enter a name and description for the schedule in the text fields provided.

Choose a frequency from the **Frequency** dropdown list. You are presented with different options based on which frequency item you choose.

From the **Time Zone** dropdown list, select a time zone for the schedule.

In the **Start Date** field, click the **Calendar** button to select a date and time.

In the **Duration** text field, enter the duration of the workshift in minutes.

6. Expand the Advanced region. Specify the thread allocation and the asynchronous job limits.
 - a. In the Thread Allocation field, enter the number of threads to be allocated to the processor when the workshift is active. If the total thread allocation of the active workshifts exceeds the number of available threads, then the threads are allocated proportionately to this workshift.
 - b. In the Asynchronous Job Limits region, enter the number of asynchronous Java and PL/SQL jobs that can be active for the work assignment. This limit applies across all processors to which the work assignment is bound.
7. Click **OK** to save the workshift.

5.3.2.2 Deleting a Workshift

You can delete a workshift from the list of workshifts. Before attempting to delete a work assignment, make sure that it is not bound to a work assignment.

To delete a workshift:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Work Allocation** and then select **Workshifts**.

The Workshifts page displays.

3. Select the workshift you want to delete and click **Delete**.

5.3.3 Managing Schedules

Schedules are used to configure the start and end times as well as the frequency of work allocations and purge policies.

Note:

For information about scheduling job request execution, see [Creating or Editing Predefined Job Schedules](#).

This section contains the following topics:

- [Creating a Schedule](#)
- [Editing a Schedule](#)
- [Deleting a Schedule](#)

5.3.3.1 Creating a Schedule

A job schedule specifies a predefined time or recurrence for execution. Work allocation and purge policy schedules are defined separately from job schedules. A job schedule can be associated with one or more jobs

To create a schedule:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. There are two ways to create schedules:
 - To create a schedule for job requests, select **Job Requests** and then select **Define Schedules** from the Scheduling Services menu.
 - To create a schedule for work assignments and purge policies, select **Work Allocation** and then select **Schedules** from the Scheduling Services menu.
3. Click **Create** to define a new schedule.

The Create Schedule page displays.

4. Enter the following information:
 - **Name:** Enter a name for the schedule.
 - **Description:** Optionally, add descriptive text for the schedule.
 - Choose a frequency from the **Frequency** dropdown list. You are presented with different options based on which frequency item you choose.
 - From the **Time Zone** dropdown list, select a time zone for the schedule.
 - In the **Start Date** field, use the **Calendar** button to select a date and time.
5. Click **OK** to save the schedule.

5.3.3.2 Editing a Schedule

To modify an existing schedule:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Work Allocation** and then select **Schedules**.

The Schedules page displays.

3. Select a schedule from the table and click **Edit**.

The Edit Schedule page displays.

4. Edit the following information:
 - **Description:** Optionally, change the descriptive text for the schedule.
 - Click the **Change Frequency** button. You are presented with different options based on which frequency item you choose.

- From the **Time Zone** dropdown list, select a time zone for the schedule.
- In the **Start Date** field, use the **Calendar** button to select a date and time.

5. Click **OK** to save the schedule.

5.3.3.3 Deleting a Schedule

You can delete a schedule from the list of schedules.

To delete a schedule:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. From the Scheduling Services menu, select **Job Request** and then select **Schedules**.

The Schedules page displays.

3. Select the schedule you want to delete and click the **Delete** button.

Note:

To delete a work assignment or purge policy schedule, from the Scheduling Services menu, select **Work Allocation** and then select **Schedules**.

5.4 Managing Metadata Security for Jobs

You can manage the level of access assigned for working with job metadata.

Using Fusion Middleware Control, you can create policies that grant permissions for resources in your application.

5.4.1 Metadata Security Actions

You grant permissions by associating specific actions with resources, then granting those permissions (or permission/resource combinations) to particular users, groups, or roles.

When granting permissions, you specify the actions that someone can take with a given resources. [Table 5-7](#) lists possible actions.

Table 5-7 Grant Actions for Metadata Security

Action	Effect
READ	Read the job metadata.
EXECUTE	Submit a job request.
CREATE	Add metadata.
UPDATE	Change the metadata.
DELETE	Delete the metadata.

The resources with which you associate permissions are expressed as entities known to the application, such as by the entity's containing package. That list includes items you or application developers might have defined as well as items Oracle Enterprise

Scheduler has defined. [Table 5-8](#) lists a few examples of the effect of granting permission for certain actions to certain resources.

Table 5-8 Sample Permission Grants for Security

Resource	Actions	Effect
<code>mypackage.JobDefinition.MyJavaSucJobDef</code>	EXECUTE	Grants the ability to submit requests for a single metadata item.
<code>mypackage.subpackage.*</code>	CREATE,EXECUTE	Grants to ability to create and execute any new metadata items in <code>/mypackage/subpackage</code>
<code>JobDefinition.SYS_AdHocRequest</code>	CREATE,EXECUTE	Grants ad hoc submission permission
<code>mypackage.*</code>	CREATE,EXECUTE,DELETE	Grants wide-open permissions

5.4.2 How to Create Metadata Policies for Oracle Enterprise Scheduler Resources

You can use Enterprise Manager to create functional To manage metadata permissions:

1. From the navigation pane, expand the **WebLogic Domain** folder and select the domain for which you're creating policies.
2. From the WebLogic Domain menu, select **Security** and then select **Application Policies**.

The Application Policies page displays.

3. In the Search section, from the **Application Stripe** dropdown, select the application stripe with which you want to work.
4. Click **Create** to begin granting permissions to certain users, groups, or application roles.

The Create Application Grant page appears.

5. In the Create Application Grant page, in the Grantee section, click **Add**.
6. In the Add Principal window, from the **Type** dropdown, select a type of principal, then enter a principal name or display name and click the search button to find the principal you want to add.
7. Under Search Principals, click the principal you want, then click **OK**.
8. In the Permissions section, click **Add**.
9. In the Search section, click **Permissions**.
10. From the Permission Class dropdown, select **oracle.as.scheduler.security.MetadataPermission**.
11. Click the **Search** button.
12. Under Search Results, select the resource to which you want to assign permissions.

The Search Results table lists resources that represent entities known to the application. [Table 5-8](#) lists a few examples.

13. Click **Continue**.

14. In the Add Permission window, in the Permission Actions field, edit the comma-separated list of permission actions so that the list reflects the permissions you want to grant.

[Table 5-7](#) lists possible actions.

15. Click **OK**.

Monitoring Oracle Enterprise Scheduler

This chapter describes how you can monitor performance of Oracle Enterprise Scheduler job requests and server instances. You can also get component status and log information.

This chapter includes the following sections:

- [Overview of Monitoring Oracle Enterprise Scheduler](#)
- [Monitoring Oracle Enterprise Scheduler Request Activity](#)
- [Monitoring the Performance of a Single Oracle Enterprise Scheduler Instance](#)
- [Monitoring Key Performance Metrics for Oracle Enterprise Scheduler](#)
- [Viewing General Oracle Enterprise Scheduler Instance Information](#)
- [Viewing Scheduler Component Status](#)
- [Managing Logging for Oracle Enterprise Scheduler](#)
- [Managing Alerts for Oracle Enterprise Scheduler in Cloud Control](#)
- [Auditing Administrative Actions](#)

6.1 Overview of Monitoring Oracle Enterprise Scheduler

Much of the monitoring you'll do is related to job requests, where Oracle Enterprise Scheduler work is done. You can get a snapshot of recent work done as job requests, including the load those requests are placing on the system. You can also view the status of current and pending requests.

For an Oracle Enterprise Scheduler instance, you can find out how it is performing currently and how it has performed historically.

You can get the status of a service instance, as well as service components such as the request dispatcher and processor.

Finally, you can get log information about job requests and Oracle Enterprise Scheduler.

Use Fusion Middleware Control to do most of the monitoring you'll need. Its user interface provides data about system performance in both graphical and tabular form.

6.2 Monitoring Oracle Enterprise Scheduler Request Activity

You can monitor job requests using Oracle Enterprise Manager Fusion Middleware Control Console.

On the console's Scheduling Service Home page, you can get the following information:

- A list of the 10 job requests that have been running for the longest period of time.
- A list of the top 10 job requests that are waiting to be run.
- A chart showing completion status percentages for job requests over the last hour.
- A chart showing the number of requests completed in the last hour, along with the average processing time over that period.
- A Scheduler Components region that you can use to start and stop the Request Processor and Request Dispatcher components.
- A list of performance data, such as average and maximum processing and wait times.

6.2.1 Viewing Top Ten Long Running Requests and Top Ten Ready Requests

In the Oracle Enterprise Manager Fusion Middleware Control Console, the Oracle Enterprise Scheduler home page displays two important categories of information: data about the top ten long running scheduled jobs and data about the top ten job requests awaiting execution. These are displayed on separate tabs.

By default, only the job requests within the current scheduling service display. However, you can change the scope such that all relevant scheduled jobs running on all scheduling services sharing the Oracle Enterprise Scheduler repository display in the tabs.

Each of the tabs includes a table that displays a short list of job requests. The Show All Ready or Show All Running link at the bottom of the region displays a search page in which it is possible to search for a more comprehensive list of relevant job requests.

Each tab displays the following information about scheduled job requests:

- **Request ID:** The ID associated with the job request.
- **Run Time/Wait Time:** The period of time during which the job request has been running or awaiting execution, respectively.
- **Job Definition:** The job definition associated with the job request.
- **Submitted by:** The name of the user who submitted the job request.
- **Application:** The name of the application with which the job request is associated.
- **Execution Type:** The category of job being executed—Java, SQL or process job.
- **Description:** An optional description of the scheduled job request.

This section contains the following topics:

- [Viewing Top Ten Long Running Oracle Enterprise Scheduler Requests](#)
- [Viewing Top Ten Oracle Enterprise Scheduler Jobs Ready to Be Executed](#)

6.2.1.1 Viewing Top Ten Long Running Oracle Enterprise Scheduler Requests

The Top Ten Long Running Requests tab displays the top ten long running scheduled job requests, including request ID, job run time, job definition used, executing application, job execution type and description. You can set the scope of the top ten long running requests displayed to the current scheduling service only, or all scheduling services sharing the Oracle Enterprise Scheduler repository.

Use the Top Ten Long Running Job Requests tab to view the job requests that have been running for the longest period of time on the Oracle Enterprise Scheduler service. You can alternatively change the scope of the region to display the top ten long running job requests on all scheduling services sharing the same repository.

The job requests displayed have a status of **RUNNING**.

To view the top ten long running requests:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. At the top of the Oracle Enterprise Scheduler home page, click the **Top 10 Long Running Job Requests** tab to display the ten job requests that have been running the longest.
3. To display all running job requests, click **Show All Running** at the bottom of the chart. This displays a search form that searches for all requests that are currently running. Click **Search** to display the requests.

6.2.1.2 Viewing Top Ten Oracle Enterprise Scheduler Jobs Ready to Be Executed

The Top Ten Ready Job Requests tab displays the top ten scheduled job requests that await execution. The tab displays the same information as the Top Ten Long Running Requests tab, except the wait time displays for each job rather than the run time.

Use the Top Ten Ready Job Requests tab to view the job requests that are awaiting execution on the Oracle Enterprise Scheduler. You can also change the scope of the region to display the top ten ready job requests on all scheduling services sharing the same repository.

The job requests displayed have a status of **READY**.

To view the top ten ready job requests:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduler application.
2. At the top of the Oracle Enterprise Scheduler page, click the **Top 10 Ready Job Requests** tab to display the ten job requests with the status **READY**.
3. To display all ready job requests, click **Show All Ready** at the bottom of the chart. This displays a search form that searches for all requests with the status **READY**. Click **Search** to display the requests.

6.2.2 Viewing Completed Job Requests

You can view the status of completed job requests on the Scheduling Service home page of the Fusion Middleware Control. There, in the **Completed Job Requests - Last Hour** region, each completion status is shown as a percentage of the total of job requests completed in the last hour. Status shown includes Success, Warning, Error, and Cancelled.

6.2.3 Viewing Job Request Response and Load

You can view the information about the processing load that has been required to complete job requests on the Scheduling Service home page of the Fusion Middleware Control. In the Response and Load region, you can view performance monitoring statistics regarding the time required to process to job requests.

[Table 6-1](#) describes the performance monitoring statistics in the **Response and Load** region.

Table 6-1 Response and Load Statistics

Statistic	Description
Average Processing time for Last Hour	This metric specifies the average time required to process jobs during the last hour.
Requests Completed Per Hour	This metric specifies the number of scheduled job requests completed within the last hour.

6.2.4 Viewing Performance as Processing and Wait Times

You can view job request processing and wait times on the Scheduling Service home page of the Fusion Middleware Control.

[Table 6-2](#) describes the performance monitoring statistics in the **Performance** region.

Table 6-2 Performance Statistics

Statistic	Description
Maximum Processing Time	The maximum amount of time required to process a scheduled job.
Average Processing Time	The average amount of time required to process a scheduled job.
Maximum Wait Time	The maximum amount of time during which a scheduled job waits before running.
Average Wait Time	The average amount of time during which a scheduled job waits before running.

6.3 Monitoring the Performance of a Single Oracle Enterprise Scheduler Instance

You can monitor the performance of an Oracle Enterprise Scheduler instance, specifically current and historical performance data, as well as a summary of all performance data. Performance data displays in a chart set that you can save, delete, and reorder. You can also modify the time period for which performance data displays.

This section contains the following topics:

- [Viewing a Summary of Oracle Enterprise Scheduler Activity](#)
- [Viewing Current Oracle Enterprise Scheduler Activity](#)
- [Viewing Historical Oracle Enterprise Scheduler Reports](#)
- [Managing Chart Sets](#)
- [Selecting the Performance Time Period](#)

6.3.1 Viewing a Summary of Oracle Enterprise Scheduler Activity

The Oracle Enterprise Scheduler Performance Summary page displays current charts of job requests such as requests completed per hour, running requests, requests by processing time and requests by status.

To view a summary of Oracle Enterprise Scheduler activity:

1. From the **Scheduling Service** menu, choose **Performance > Service Summary**.

By default, this page displays job requests running during a selected time period in the following categories:

- Requests completed per hour
- Processing times, average, minimum and maximum
- Running requests and maximum running requests
- Requests over the past hour with the status SUCCESS, WARNING, ERROR or CANCEL.

You can change the metrics displayed on the Service Summary performance page. Click the **Show Metric Palette** button to display the metric palette for Oracle Enterprise Scheduler.

2. You can change the duration for which performance data display. See [Selecting the Performance Time Period](#) for more information.
3. Optionally, you can change the display format of performance data on the page. Choose **View > Grid** to remove the grid display format from the charts. Click the **Table View** link to display the performance data in table format.
4. You can overlay an additional scheduler service over the current service performance metrics.
 - a. Choose **Overlay > Another Scheduler Service**.
In the Search and Select: Targets window, enter the name of the target and host server and click **Search** to search for the relevant service.
 - b. From the search results table, select the relevant service and click **Select** to choose the scheduler service to overlay.
5. Additionally, you can take any of the following actions.
 - Save the chart set, as described in [Saving a Chart Set](#).
 - Reorder the chart set, as described in [Reordering Charts](#).
 - Revert to an old chart set, as described in [Reverting to an Old Chart Set](#).
 - Delete a chart set, as described in [Deleting a Chart Set](#).

6.3.2 Viewing Current Oracle Enterprise Scheduler Activity

The Oracle Enterprise Scheduler Current Activity page displays pending and running requests by user and application.

To view current Oracle Enterprise Scheduler activity:

1. From the **Scheduling Service** menu, choose **Performance > Current Activity**.
2. You can view pending and running requests by user or application, or workload by work assignment.

To view pending and running requests by user, choose **View > Requests By User**.

To view pending and running requests by application, choose **View > Requests By Application**.

To view workloads by work assignment, choose **View > Workload By Work Assignment**. For more information about work assignment details, see [Creating or Editing a Work Assignment](#).

6.3.3 Viewing Historical Oracle Enterprise Scheduler Reports

The Oracle Enterprise Scheduler Historical Reports page displays completed job requests by user, application and work assignment.

To view historical Oracle Enterprise Scheduler reports:

1. From the **Scheduling Service** menu, choose **Performance > Historical Reports**.
2. From the **View** list, choose one of the following report views:
 - Completed Request Statistics By Job Name
 - Completed Request Statistics By User
 - Request Metrics By Work Assignment
 - Completed Request Statistics
3. Adjust the filter criteria as required. The filter criteria vary depending on the report view you choose.

Table 6-3 Completed Request Statistics by Job Name

Filter Criterion	What to Do...
Application	From the list, choose the application whose statistics you want to view.
Minimum Run Time	Enter the shortest run time period for jobs whose statistics you want to view. Use the text field to enter a number, and select a unit of time from the list: Seconds, Minutes, Hours or Days .
Completed between	Use the calendar buttons on either side of the text fields to select a start and end date and time.

Table 6-4 Completed Request Statistics by User

Filter Criterion	What to Do...
Application	From the list, choose the application whose statistics you want to view.
Minimum Total Run Time	Enter the shortest total time period for jobs whose statistics you want to view. Use the text field to enter a number, and select a unit of time from the list: Seconds, Minutes, Hours or Days .

Table 6-4 (Cont.) Completed Request Statistics by User

Filter Criterion	What to Do...
Completed between	Use the Calendar buttons on either side of the text fields to select a start and end date and time.

Table 6-5 Request Metrics By Work Assignment

Filter Criterion	What to Do...
Application	From the list, choose the application whose statistics you want to view.
Completed after	Use the Calendar button to choose the date and time after which the work assignment has completed.

Table 6-6 Completed Request Statistics

Filter Criterion	What to Do...
Application	From the list, choose the application whose statistics you want to view.
Completed between	Use the Calendar buttons on either side of the text fields to choose a start and end date and time.

4. Click **Go** to search for and display the historical data.

6.3.4 Managing Chart Sets

You can save or delete the chart set displayed in the Oracle Enterprise Scheduler Performance Summary page. You can also re-organize the order of a chart and revert to an old chart set.

This section contains the following topics:

- [Saving a Chart Set](#)
- [Deleting a Chart Set](#)
- [Reordering Charts](#)
- [Reverting to an Old Chart Set](#)

6.3.4.1 Saving a Chart Set

You can save the chart set displayed in the Oracle Enterprise Scheduler Performance Summary page. Saving a chart set allows preserving a specific set of metrics selected for display, as opposed to the default metrics.

To save a chart set:

1. Display the Oracle Enterprise Scheduler Performance Summary page.
2. Click the **Save Charts** button.
3. In the Save Charts dialog box, enter a name for the chart set and click **OK**.
4. After you have saved the chart set, you can display it by choosing **Chart Set > My Chart Set**.

6.3.4.2 Deleting a Chart Set

You can delete saved chart set files created in the Oracle Enterprise Scheduler Performance Summary page.

To delete a chart set:

1. Display the Oracle Enterprise Scheduler Performance Summary page. [Viewing a Summary of Oracle Enterprise Scheduler Activity](#)
2. Choose **View > Delete Chart Sets**.
3. In the Delete Chart Sets dialog box, select the name of the chart set to be deleted and click **OK**.

6.3.4.3 Reordering Charts

You can organize the display of performance charts in the Oracle Enterprise Scheduler Performance Summary page. Charts can be moved up or down, or removed entirely.

To reorder chart sets:

1. Display the Oracle Enterprise Scheduler Performance Summary page. [Viewing a Summary of Oracle Enterprise Scheduler Activity](#)
2. Choose **View > Reorder Charts**.
3. In the Reorder Charts popup window, select the chart you want to move and click **Move Up** or **Move Down**, respectively.

To delete a chart from the Performance Summary page, select the chart and click **Remove**.

6.3.4.4 Reverting to an Old Chart Set

If you have re-organized your chart sets, you can revert to the chart set as you had originally saved it.

To revert chart sets:

1. Display the Oracle Enterprise Scheduler Performance Summary page. [Viewing a Summary of Oracle Enterprise Scheduler Activity](#)
2. From the **Chart Set** list, choose the chart you want to revert to its original display.
3. Choose **View > Revert Chart Set**.

6.3.5 Selecting the Performance Time Period

In the Oracle Enterprise Scheduler Performance Summary page, you can change the time period for which the performance summary displays.

To select the performance time period:

1. Display the Oracle Enterprise Scheduler Performance Summary page. [Viewing a Summary of Oracle Enterprise Scheduler Activity](#)
2. Change the duration for which job requests display.

- **All:** Click to display all job requests within a specified, finite period.
- **15 minutes:** Click to display all job requests within the past fifteen minutes.
- **Slider:** Click to display a slider that enables scrolling horizontally to show job requests from a specified period. The slider defaults to whichever time segment is selected, "all," 2 hours, or 15 minutes.
- **Calendar:** Click the **Calendar** button to select a time period for which you want to display job requests.

6.4 Monitoring Key Performance Metrics for Oracle Enterprise Scheduler

Key performance metrics include the number of active processor threads for each instance, processing times, wait times, and job running times.

Use [Table 6-7](#) to locate the key performance metrics for Oracle Enterprise Scheduler.

Table 6-7 Oracle Enterprise Scheduler Key Performance Metrics

Metric Name	Location in Fusion Middleware Control
Number of active processor threads for each instance	1. From the navigation pane, expand Scheduling Services. 2. Select the Oracle Enterprise Scheduler component. 3. From the Scheduling Service menu, choose Performance > Current Activity. 4. From the View list, choose Workload by Work Assignment. 5. In the Work Assignments with active Workshifts report, view the Thread Allocation columns.
Processing Time (for jobs) Wait Time	1. From the navigation pane, expand Scheduling Services. 2. Select the Oracle Enterprise Scheduler component. 3. From the Scheduling Service menu, choose Performance > Historical Reports. 4. In the Historical Reports page, from the View list, choose Request Metrics By Work Assignment. 5. In the Results section, view the Wait Time and Processing Time columns.
Job running time	1. From the navigation pane, expand Scheduling Services. 2. Select the Oracle Enterprise Scheduler component. 3. From the Scheduling Service home page, in the Top 10 Long Running Request tab, select a job. 4. In the Request Detail page, under Execution Trail, view the Run Time metric.

6.5 Viewing General Oracle Enterprise Scheduler Instance Information

You can view information about a particular Oracle Enterprise Scheduler instance by using the Scheduling Service General Information popup window in Fusion Middleware Control.

The popup window displays the locations of Oracle Fusion Middleware home, the domain and the target application, as well as the version of Oracle Fusion Middleware currently running and the URI of the hosting server.

To view general information about Oracle Enterprise Scheduler:

1. From the Scheduling Service menu, select one of the following:
 - From the Scheduling Service menu, select **General Information**.
 - In Cloud Control, from the Scheduling Service menu, select **Target Information**.
2. Fusion Middleware Control displays the following information.
 - **Target Name:** The location of the target application, in this case, Oracle Enterprise Scheduler, running on the domain.
 - **Version:** The version number of Oracle Fusion Middleware.
 - **Middleware Home:** The location of the Oracle Fusion Middleware directory.
 - **Domain Home:** The full path of the domain.
 - **Host:** The URI of the hosting server.

6.6 Viewing Scheduler Component Status

You can view status for components of Oracle Enterprise Scheduler. On the Scheduling Service home page of Fusion Middleware Control, the Scheduler Components region displays status for the job request processor and dispatcher.

The region displays the status (such as whether it is started or stopped) of each component, the name of the server to which it is deployed, and whether or not the component is enabled. You can start and stop each component there.

6.7 Managing Logging for Oracle Enterprise Scheduler

You can search for and view log data for individual job requests, as well as set log levels for Oracle Enterprise Scheduler. You can also save job request logs to a file, and trace job requests for additional troubleshooting information.

This section contains the following topics:

- [Viewing Log Information for the Domain](#)
- [Viewing Job Request Logs](#)
- [Viewing Log Messages for Scheduled Job Requests](#)
- [Setting Oracle Enterprise Scheduler Log Levels](#)
- [Tracing Oracle Enterprise Scheduler Jobs](#)
- [Saving Job Request Logs](#)

Note:

The Oracle WebLogic Server logger (`logging.xml`) only shows logs written by Oracle Enterprise Scheduler job running in Oracle WebLogic Server. After Oracle Enterprise Scheduler transfers control of running PL/SQL to the PL/SQL process, PL/SQL job logging data is not written to the Oracle Enterprise Scheduler logs as they run in a separate process.

Note:

The procedures in this section assume that your log information is stored in the file system. When you select a job, the View Log Messages button is active. If instead, your log information is stored in a Universal Content Management (UCM) repository, the button is not active and you access the log by clicking on the `fnd:attachment` component.

6.7.1 Viewing Log Information for the Domain

You can use the Fusion Middleware Control to view log messages for the domain on which Oracle Enterprise Scheduler is deployed.

To view domain logs:

- From the Scheduling Service menu, select **Logs > View Log Messages**.

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

6.7.2 Viewing Job Request Logs

You can view the log for a particular job request from the Request Details pane.

Note:

On Windows, request log output is formatted as a single line without line breaks.

To view job request logs:

1. Search for the relevant job requests as described in [Searching for a Job Request Using Simple Search](#)
2. In the table displaying the job request search results, select the job request whose log you want to view.
3. To view job request details, click the job request ID. Alternatively, click the parent ID associated with the job request to view the details of the job set with which the job is associated.
4. In the job request details page, select **Action** and then select **Request Log** to display log information for the job request.

6.7.3 Viewing Log Messages for Scheduled Job Requests

You can use the Oracle Enterprise Scheduler log messages page to view logging information regarding scheduled job requests.

To view log messages for scheduled job requests with Fusion Middleware Control:

1. From the navigation pane, expand the farm, and then **Scheduling Services**.
2. Select the **ESSAPP** application for the appropriate Managed Server.
3. In the Scheduling Service home page, from the **Scheduling Service** menu, choose **Logs** and then select **View Log Messages**.
4. Search for the relevant log messages using the Date Range, Message Types and Message fields. You can optionally add additional search fields.
5. You can use the ECID to retrieve more information about a given job request. The Execution Context ID (ECID) is a global unique identifier of the execution of a particular request in which the originating component participates. You can use the ECID to correlate error messages from different components.
6. By default, when you view the logs for a request, Fusion Middleware Control displays only messages logged in the scope of the Oracle Enterprise Scheduler cluster. If the Oracle Enterprise Scheduler application is not deployed to a cluster, Fusion Middleware Control displays the messages logged in the managed server scope. However, Oracle Enterprise Scheduler propagates the ECID associated with the job request across sub-systems such as SOA, ADF, and so on.

To view messages logged by other sub-systems, broaden the target scope to display messages logged across the domain or farm. Click **Target Log Files** and select the target whose log messages you want to view.

6.7.4 Setting Oracle Enterprise Scheduler Log Levels

It is possible to configure Oracle Enterprise Scheduler server logging for an Oracle WebLogic Server by modifying the `logging.xml` file of that Oracle WebLogic Server. By default, there is no explicit logger entry for Oracle Enterprise Scheduler. Oracle Enterprise Scheduler inherits the logging level and log handlers configured for the parent logger, typically the `oracle` logger or the root logger.

By default, the log messages for the Oracle Enterprise Scheduler logger can be found in the Oracle WebLogic Server diagnostic log file for that Oracle WebLogic Server. The `logging.xml` file is located under `DOMAIN_HOME/config/fmwconfig/servers/WebLogic_Server_Name`, where `DOMAIN_HOME` is the domain home directory for the Oracle WebLogic Server domain and `WebLogic_Server_Name` is the name of the Oracle WebLogic Server that uses the `logging.xml` file.

[Table 6-8](#) shows the Oracle Enterprise Scheduler logger names, log levels and a description for each level.

Table 6-8 *Loggers and log levels for Oracle Enterprise Scheduler*

Logger Name	Log Level	Description
oracle.as.ess	SEVERE	Problems encountered by Oracle Enterprise Scheduler runtime in the context of request processing that result in the request errors. Errors include exceptions thrown by the job code, unchecked exceptions when running the job code and exceptions when running Oracle Enterprise Scheduler code. Problems encountered by Oracle Enterprise Scheduler runtime outside the context of request processing, such as dispatching, system event handling, and so on.
-	WARNING	Less severe problems encountered by Oracle Enterprise Scheduler runtime during or outside of request processing, which might not cause requests to enter error state.
-	INFO	Messages for request state transitions. Messages related to work assignment activities. Messages about batch delete failures. Start and stop of Oracle Enterprise Scheduler resource adapter.
-	CONFIG	Application endpoint activation and de-activation for Oracle Enterprise Scheduler resource adapter.

- Request Logging: Oracle Enterprise Scheduler job implementation might write business-specific job request execution log information to the job request log file. This log file is specific to each request, and is automatically enabled by default. For more information about viewing the log file for a job request, see [Viewing Job Request Logs](#).

For information about setting the log levels for Oracle WebLogic Server, see [Tracing Oracle Enterprise Scheduler Jobs](#).

Set the log levels for the Oracle WebLogic Server running Oracle Enterprise Scheduler as described in the Oracle WebLogic Server documentation.

6.7.5 Tracing Oracle Enterprise Scheduler Jobs

Enabling tracing for Oracle Enterprise Scheduler jobs provides additional information regarding job requests, which can then be relayed to Oracle technical support to help solve any Oracle Enterprise Scheduler-related issues.

This section contains the following topics:

- [Before You Begin](#)
- [Define the Loggers](#)
- [Attach the Loggers](#)

- [Download the Trace File](#)

6.7.5.1 Before You Begin

Before enabling tracing, note that tracing log messages may produce a considerable amount of output. By default, traces go to the Oracle WebLogic Server console `stdout`. If the `stdout` is re-directed to a file, it may occupy a significant amount of disk space and possibly affect server performance.

To enable tracing for Oracle Enterprise Scheduler jobs, navigate to the Specify Loggers Region:

In the Log Levels tab, from the View dropdown list select Loggers with Persistent Log Level State. Expand the Specify Loggers region toward the bottom of the pane.

1. In Fusion Middleware Control, in the navigation pane, select WebLogic Domain and then select the domain and server names.
2. Click the WebLogic Server menu and select Logs and then select Log Configuration.

The Log Configuration Pane displays.

3. In the Log Levels tab, from the View dropdown list select Loggers with Persistent Log Level State. Expand the Specify Loggers region toward the bottom of the pane.
4. Navigate to the Log Configuration pane and expand the Specify Loggers region.

6.7.5.2 Define the Loggers

Create the following loggers:

- `oracle.as.scheduler.security.internal.SecurityExecutorFactory`
 - `oracle.as.scheduler.security.internal.SecurityHelper`
1. In the Name text field, enter the name of the logger.
 2. From the Oracle Diagnostic Logging Level (Java Level) dropdown list, select the desired logging level.
 3. Repeat for the second logger name.

6.7.5.3 Attach the Loggers

Attach the loggers to the Oracle Diagnostic Logging Handler.

1. In the Log Configuration pane, click the Log Files tab.
2. From the list of handlers, select `odl-handler` and click Edit Configuration.
3. From the Loggers to Associate dropdown list, select the loggers you created and click OK.

6.7.5.4 Download the Trace File

Download the Trace File

1. In Fusion Middleware Control, in the navigation pane, select WebLogic Domain and then select the domain and server names.

2. Click the WebLogic Server menu and select Logs and then select View Log Messages.
The Log Messages pane displays.
3. In the Search field, expand the Selected Targets region and select the relevant Oracle Enterprise Scheduler target.
4. Select the relevant date range, message type and so on, and click Search.
5. From the list of log messages that displays, select the relevant message.
6. In the Log File column, click the name of the log file.
The View Log File pane displays.
7. Next to the name of the log file, click Download to download the file.

6.7.6 Saving Job Request Logs

It is possible to save job request log data to the server log file. Job request logs are normally stored to Oracle WebCenter Content. However, when setting the log level to FINER, all job request logs are copied to the server log file.

To save job request logs to the server diagnostic file:

1. In Fusion Middleware Control, in the navigation pane, select **WebLogic Domain** and then select the domain and server names.
2. Click the **WebLogic Server** menu and select **Logs** and then select **Log Configuration**.
The Log Configuration pane displays.
3. In the Log Configuration pane, click the **Log Levels** tab.
4. In the Search text field, search for **oracle.as.scheduler.** and find the **oracle.as.scheduler.security.internal.SecurityExecutorFactory** logger.
5. Set the log level to FINER and click **Apply**.
6. Run a diagnostic test, as described in *Oracle Fusion Middleware Administrator's Guide*. The log messages are now saved to the `<server>-diagnostic.log` file.

The logs are saved to the server diagnostic file, `<Oracle Enterprise Scheduler server name>-diagnostic.log`.

6.8 Managing Alerts for Oracle Enterprise Scheduler in Cloud Control

Cloud Control enables you to configure, view and manage alerts for Oracle Enterprise Scheduler entities including the scheduling service itself, work assignments and job definitions. Alerts are based on performance metrics for Oracle Enterprise Scheduler.

Each metric includes monitored objects with upper and lower thresholds indicating warning or critical states for that particular metric. For example, you can configure an alert on number of job requests that have entered a state of error, such that 10 errors constitute a warning state, and 20 errors a critical state.

After you have configured alerts, alerts are generated automatically and its details can be viewed from Incident Manager provided in Cloud Control. Any changes to

configuration made within a specified period of time can also be tracked as described in [Searching for Configuration Changes to Oracle Enterprise Scheduler in Cloud Control](#).

Table 6-9 lists the alerts available for Oracle Enterprise Scheduler.

Table 6-9 Oracle Enterprise Scheduler Alerts

Alert	Scope	Collected Data	Alert Automatically Cleared?
Alert for job requests in ready state and the average wait time for job requests in ready state.	Work assignment	The number of job requests in ready state and average wait time of the ready job requests by work assignment.	No. The alert is only cleared when the number of ready job requests is smaller than the threshold. The same is true for alerts pertaining to the average wait time of a ready job request.
Alert for job requests that have entered a state of error.	Work assignment	The number of job requests that have entered an error state following the last collection.	Yes. The alert is automatically cleared if the delta value of the subsequent collection is below the threshold.
Alert for long running job requests.	Job definition	The longest running time of a job request listed by job definition. The data is queried for only those job definitions for which a threshold is specified.	No. The alert is cleared when the long running request transitions to another state.
Alert for job requests that have gone to a state of <code>ERROR_MANUAL_RECOVERY</code> .	Scheduling service	The number of job requests in a state of <code>ERROR_MANUAL_RECOVERY</code> .	No. An administrator must recover the job request in order to clear the alert.
Alert for timed out job requests.	Scheduling service	The number of timed out job requests.	No. An administrator must cancel or recover the job request in order to clear the alert.
Alert for job requests that have gone into <code>ERROR</code> state.	Scheduling service	The number of job requests that have gone into <code>ERROR</code> state during the last hour.	Yes. The alert automatically clears if the delta value in the subsequent collection is below the threshold.
Alert for blocked job requests.	Scheduling service	The number of job requests in <code>BLOCKED</code> state.	No. An administrator must cancel the job request, or the job request must be selected for processing in order to clear the alert.

This section contains the following topics:

- [Configuring Metric Alerts for Oracle Enterprise Scheduler in Cloud Control](#)
- [Viewing Alerts for Oracle Enterprise Scheduler in Cloud Control](#)

6.8.1 Configuring Metric Alerts for Oracle Enterprise Scheduler in Cloud Control

You can configure the thresholds for each object monitored by the metric. For a given metric, you can configure alert rules as follows:

- Select an object that the metric monitors, such as job requests in an error state.
- Configure the values at which the object reaches warning and critical states, including an operator value such as greater than (>) or less than (<).
- Enter a corrective action to be taken in the event that the alert criteria are fulfilled.
- Arrange the alert rules in the order you desire.

To configure metric alerts for Oracle Enterprise Scheduler in Cloud Control:

1. Navigate to the scheduling service.
 - a. From the **Targets** menu, choose **Middleware**.
 - b. Navigate to the Oracle Fusion Middleware farm with Oracle Enterprise Scheduler deployed.
 - c. In the **Target Navigation** pane, expand the **Scheduling Services** folder.
 - d. Click on the Oracle Enterprise Scheduler cluster target or individual Oracle Enterprise Scheduler target.

2. From the Scheduling Service **Targets** menu, select **Monitoring** and then click **Metric and Collection Settings**.

A list of metrics displays.

3. Select **All metrics** from the **View** dropdown to view all metrics.
4. Select the metric for which you want to configure alert thresholds and click the **Edit** icon. For example, you can configure the thresholds for the **Average Wait Time for Requests in Ready State (in seconds)** metric.

The **Edit Advanced Settings** page displays, including the Monitored Objects list.

5. In the **Edit Advanced Settings** page, click **Add** to add a row to the list of monitored objects.

6. Select the object you want to monitor.

- a. In the **Name** field of the new row, click the **Search** button.

The Search and Select Objects window displays.

- b. Search for the object you want to use and click **Select**.

7. In the spaces provided, enter the values to be used for the Critical and Warning thresholds.

8. Click **Continue**, then click **OK** to save your settings.

9. Configure any additional thresholds as required.

For more information about configuring alerts, see the following topics in the Cloud Control online help:

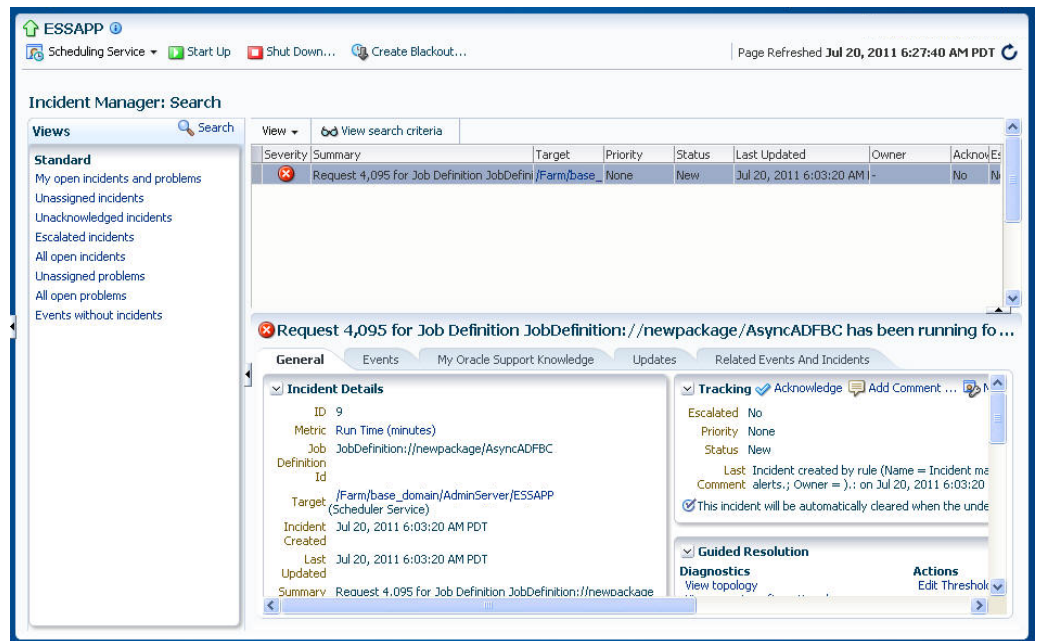
- "Incident Rules—Common Tasks"
- "Incident Rules—Advanced Tasks"

6.8.2 Viewing Alerts for Oracle Enterprise Scheduler in Cloud Control

The Oracle Enterprise Scheduler Service home page displays a summary of alerts. You can navigate to the Incident Manager page to view additional information about incidents by clicking the number of incidents displayed in the Monitoring and Diagnostics area.

To view alerts for Oracle Enterprise Scheduler with Cloud Control:

1. Navigate to the scheduling service.
 - a. From the **Targets** menu, choose **Middleware**.
 - b. Navigate to the Oracle Fusion Middleware farm with Oracle Enterprise Scheduler deployed.
 - c. In the **Target Navigation** pane, expand the **Scheduling Services** folder.
 - d. Click on the Oracle Enterprise Scheduler cluster target or individual Oracle Enterprise Scheduler target.
2. The Scheduling Service home page displays. The Monitoring and Diagnostics pane displays the following information.
 - **Incidents:** The number of incidents that have occurred in the scheduling service instance displays here. The number of incidents is determined by those incidents collected as a result of configuring job request alerts. For more information about configuring job request alerts, see [Configuring Metric Alerts for Oracle Enterprise Scheduler in Cloud Control](#). If the number of incidents is greater than zero, click the number for more information about the incidents that have occurred.
 - **Descendant Target Incidents:** The number of incidents that have occurred anywhere in the scheduling service cluster displays here. If the number of incidents is greater than zero, click the number for more information about the incidents that have occurred.
 - **Configuration Changes:** The number of changes that have been made to alert configurations displays here. Click the number of configuration changes to search for the changes made within a particular period of time. The default time period is seven days.
3. Click the number of incidents to display detailed information about the alerts in the Incident Manager pane. If there are no incidents, the Incident manager can be opened from the Scheduling Service target menu by selecting **Monitoring** and then clicking **Incident Manager**.



The Incident Manager displays incidents in a table, with the following information listed for each incident: severity, summary, target, priority, status, last updated, owner, acknowledged, escalated, type and category.

You can click an incident to display its details. For more information about using Incident Manager, see the following topics in the Cloud Control online help:

- "Incident Manager—Common Tasks"
- "Incident Manager—Advanced Tasks"

6.9 Auditing Administrative Actions

You can have administrative actions recorded for later review. With this record, you can audit administrative steps when required. Auditing provides a measure of accountability and answers the "who has done what and when" types of questions.

Auditing support is built into Oracle Enterprise Scheduler. To learn more about how auditing works generally, see the *Application Security Guide*.

You can configure auditing support by using WLST commands. For more information, see *WebLogic Scripting Tool Command Reference*.

For reference information about commands supported by Oracle Enterprise Scheduler, see [Audit Reference for Oracle Enterprise Scheduler](#).

This section includes the following:

- [Getting the List of Auditable Events](#)
- [Viewing Event Audit Data](#)

6.9.1 Getting the List of Auditable Events

You can get the full list of auditable events from the WLST command line. After using the connect command to connect WLST to the server instance, use the listAuditEvents command to display the list of events. In the command, specify "ESS" as the component type to list supported event types for Oracle Enterprise Scheduler:

```
listAuditEvents(componentType="ESS")
```

This and other useful commands are described in *WebLogic Scripting Tool Command Reference*.

Table 6-10 includes a partial described list of event categories and the auditable events available in them. Be sure to use the `listAuditEvents` command for a full list.

Table 6-10 Oracle Enterprise Scheduler Audit Event Categories

Category	Description
Security Check	Includes events related to security checks, including checking data security privileges, checking metadata permission, checking runtime permission, switching user context, and querying requests based on a data security check.
Operations to Requests	Includes request events related to requests, including executing, deleting, canceling, holding, locking, purging, releasing, replacing, submitting, and updating.
Metadata Configuration	Includes metadata events such as creating, deleting, and updating schedules, triggers, incompatibilities, exclusions, job definitions and job sets.
Configuration Change	Includes events such as adding, deleting, and updating ESS configuration properties.
Administrative Operations to RequestProcessor	Includes events such as enabling and disabling the request processor.
Administrative Operations to RequestDispatcher	Includes events such as starting and stopping the request dispatcher.

6.9.2 Specifying Which Events Get Logged

You can use WLST to specify filter present level for audit event logging, and so specify which events are logged. You do this by setting the filter preset level to one of three levels: low, medium, or high.

You can set the filter preset level by using the following WLST command:

```
setAuditPolicy(componentType='ESS', filterPreset="Medium")
```

Table 6-11 describes the filter preset levels:

Table 6-11 FilterPreset Levels for Event Auditing

Level	Description
low	The fewest and more important events are logged. These include the following: Submit and Execute events from the Operations to Requests category.

Table 6-11 (Cont.) FilterPreset Levels for Event Auditing

Level	Description
medium	A medium level for events between the fewest and most verbose settings. This includes all events from the following categories: • Operations to Requests • Metadata Configuration • Configuration Change • Administrative Operations to RequestProcessor • Administrative Operations to RequestDispatcher
high	The most verbose level, including all events from the following categories: • Security Check • Operations to Requests • Metadata Configuration • Configuration Change • Administrative Operations to RequestProcessor • Administrative Operations to RequestDispatcher

For more on the event categories, see [Getting the List of Auditable Events](#).

6.9.3 Viewing Event Audit Data

Data generated by event auditing is stored either in a database or a file, depending on how auditing is configured.

For more on configuring auditing, see the *Application Security Guide*.

If data is being written to a file, the file's default location is the following:

`DOMAIN_HOME/servers/SERVER_NAME/logs/audit log`

High Availability for Oracle Enterprise Scheduler

This chapter describes how you can configure and manage a highly available Oracle Enterprise Scheduler environment.

This appendix includes the following sections:

- [Introduction to High Availability for Oracle Enterprise Scheduler](#)
- [Oracle Enterprise Scheduler Concepts](#)
- [Configuring High Availability for Oracle Enterprise Scheduler](#)
- [Managing an Oracle Enterprise Scheduler Cluster](#)

A.1 Introduction to High Availability for Oracle Enterprise Scheduler

A highly available cluster of Oracle Enterprise Scheduler servers is recommended for optimal job performance. This is especially useful for running asynchronous jobs remotely, which may require returning a status message upon completion.

For example, suppose an asynchronous ADF Business Components job runs remotely. Oracle Enterprise Scheduler expects the job to send a status upon completion using a web services callback. If Oracle Enterprise Scheduler runs on only one node, if that node is down, the callback message does not arrive and the status of the job is unknown. The job would then require manual intervention to mark its status as complete.

A two node cluster, however, allows all callbacks to process and arrive at their destination even if one server is down. A clustered Oracle Enterprise Scheduler environment allows callbacks to be delivered as required, and jobs to complete with the correct status automatically assigned by the system.

The main steps required for configuring a highly available Oracle Enterprise Scheduler environment are as follows:

1. Use the Configuration Wizard to set up a domain and configure a cluster.
2. Add nodes to the cluster as required in order to enhance scalability, allowing more processing power for jobs.

When a cluster node is added, the new node's processor configuration might have to be adjusted to assign appropriate work assignments.

For more information, see the Oracle WebLogic Server documentation.

3. Configure the load balancer. For more information, see the Oracle HTTP Server documentation.

Note:

For information about troubleshooting an Oracle Enterprise Scheduler cluster, see [Troubleshooting Oracle Enterprise Scheduler](#).

A.2 Oracle Enterprise Scheduler Concepts

In order to configure an Oracle Enterprise Scheduler environment, it helps to understand concepts such as the architecture of Oracle Enterprise Scheduler, its components and life cycle.

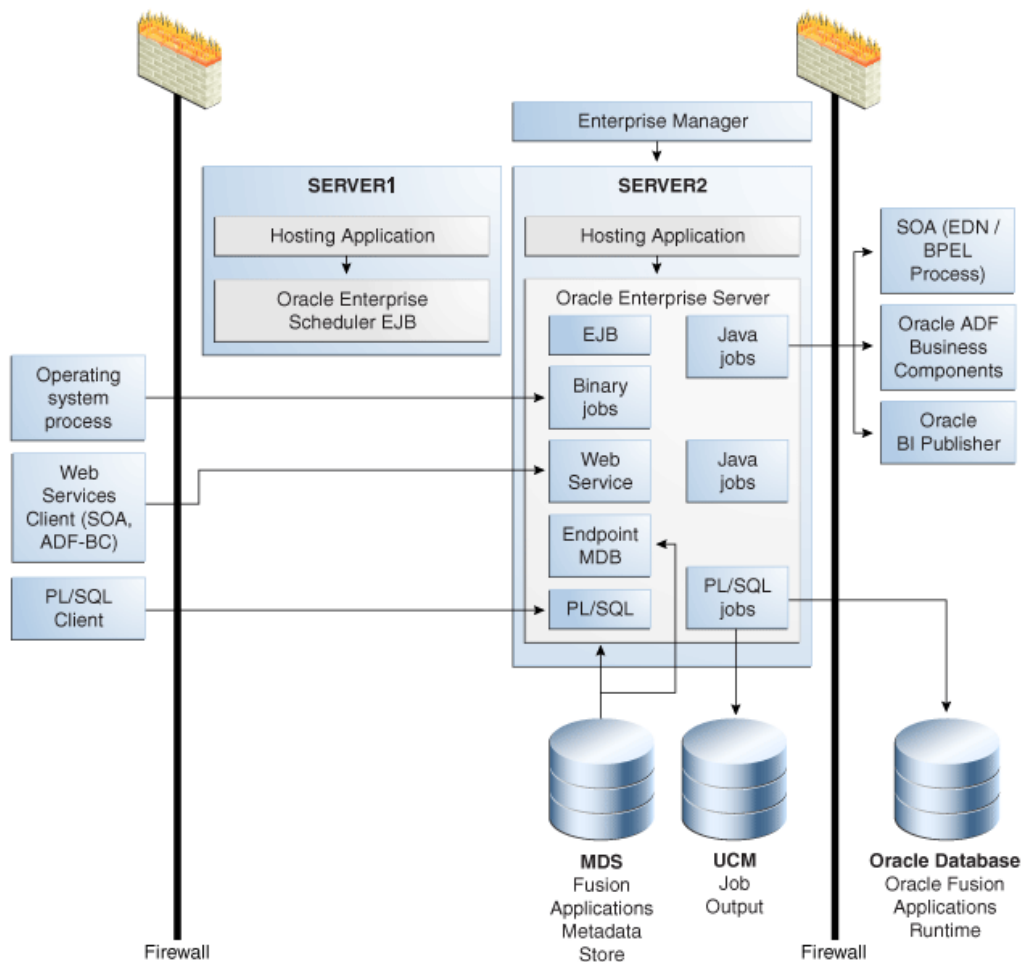
This section includes the following topics:

- [Oracle Enterprise Scheduler Architecture](#)
- [Oracle Enterprise Scheduler Components](#)
- [Oracle Enterprise Scheduler Life Cycle](#)
- [Oracle Enterprise Scheduler Life Cycle Tools](#)

A.2.1 Oracle Enterprise Scheduler Architecture

Oracle Enterprise Scheduler is installed to an Oracle WebLogic Server instance, on which it runs. The Oracle Enterprise Scheduler service component sits on top of Oracle JRF and is secured by Oracle Web Services Manager. Oracle Enterprise Scheduler manages scheduled job submissions and job definitions.

[Figure A-1](#) shows the Oracle Enterprise Scheduler runtime architecture in the context of Oracle Fusion Middleware components.

Figure A-1 Oracle Enterprise Scheduler Runtime Architecture

The components of the Oracle Enterprise Scheduler runtime architecture are as follows:

- Oracle Enterprise Scheduler client applications: Various applications can request the execution of a scheduled job. Applications include Oracle Fusion applications, Web service clients such as SOA or Oracle ADF Business Components and PL/SQL applications.
- Oracle Enterprise Scheduler: Fusion Middleware Control enables you to manage Oracle Enterprise Scheduler clusters, services and jobs. Oracle Enterprise Scheduler accesses metadata using MDS. Scheduled job output is saved to Oracle WebCenter Content. Oracle Enterprise Scheduler includes interfaces and APIs that enable interaction with applications and external components. For example, a PL/SQL client uses the Oracle Enterprise Scheduler PL/SQL API to request a scheduled job.
- Components accessed by Oracle Enterprise Scheduler: Oracle Enterprise Scheduler supports creating Java jobs that access SOA components, Oracle ADF Business Components services, and Oracle Business Intelligence Publisher (there is no direct support for jobs specific to these components).

Client applications accessing EJBs connect to Oracle Enterprise Scheduler over RMI, whereas client applications using Oracle Enterprise Scheduler web services use HTTP. Connections from client applications to the server are persistent, short-lived asynchronous interactions that sometimes use callback functions.

A.2.2 Oracle Enterprise Scheduler Components

Oracle Enterprise Scheduler components are as follows:

- Oracle ADF Server (Oracle Enterprise Scheduler client): `RuntimeServiceEJB` and `MetadataServiceEJB` are deployed as shared libraries. These libraries are imported in the ADF client applications (ears).
- Oracle Enterprise Scheduler Server (Oracle Enterprise Scheduler Runtime): The Oracle Enterprise Scheduler service component manages all scheduled jobs.
- Core runtime: This is an Oracle Enterprise Scheduler application EAR file which contains a JCA resource adapter, multiple EJB components and JRF web service modules (WAR files).
- Hosting applications: A hosting application is an EAR file that imports the `ESSEndpointMDB`, `RuntimeServiceEJB` and `MetadataServiceEJB` shared libraries. An Oracle Enterprise Scheduler hosting application submits job requests using the Oracle Enterprise Scheduler libraries or an integrated job request submission interface.
- Oracle Database Scheduler: The standard Oracle Database Scheduler is used to execute Oracle Enterprise Scheduler PL/SQL jobs.
- Oracle Enterprise Scheduler uses Java process APIs to spawn native binary jobs.

Oracle Enterprise Scheduler relies on the following data sources:

- Oracle Enterprise Scheduler runtime (XA emulation)
- Oracle Enterprise Scheduler runtime (non-XA)
- Oracle Enterprise Scheduler Metadata Store (non-XA)

An XA transaction, in the most general terms, is a global transaction that may span multiple resources. A non-XA transaction always involves just one resource, and generally cannot participate in a global transaction.

External dependencies include a run-time database, an MDS repository, as well as Oracle SOA Suite, Oracle ADF Business Components, Oracle BI Presentation Services, Oracle WebCenter Content and so on, depending on the components involved in implementing a given job.

A.2.3 Oracle Enterprise Scheduler Life Cycle

The Oracle Enterprise Scheduler engine starts up as part of the standard J2EE application initialization by Oracle WebLogic Server. The Oracle Enterprise Scheduler JCA adapter connects to the run-time schema and polls for scheduled work items.

The following is the sequence of the execution of a client request in Oracle Enterprise Scheduler.

1. A client application submits a job request.
2. Oracle Enterprise Scheduler reads the metadata for the job request.
3. Oracle Enterprise Scheduler places the job request and the job metadata in a queue in the Oracle Enterprise Scheduler run-time data store.

4. Based on schedule and request processor availability, Oracle Enterprise Scheduler sends a message to the hosted application, which includes all the job request parameters and metadata captured at the time of submission.
5. The hosting application executes the job and returns a status. Job output and logs are written to Oracle WebCenter Content.
6. Oracle Enterprise Scheduler updates the history with the job request status.

Figure A-2 shows the changes in job state during the life cycle of an executed job request.

Figure A-2 Job Request Changes in State During Runtime

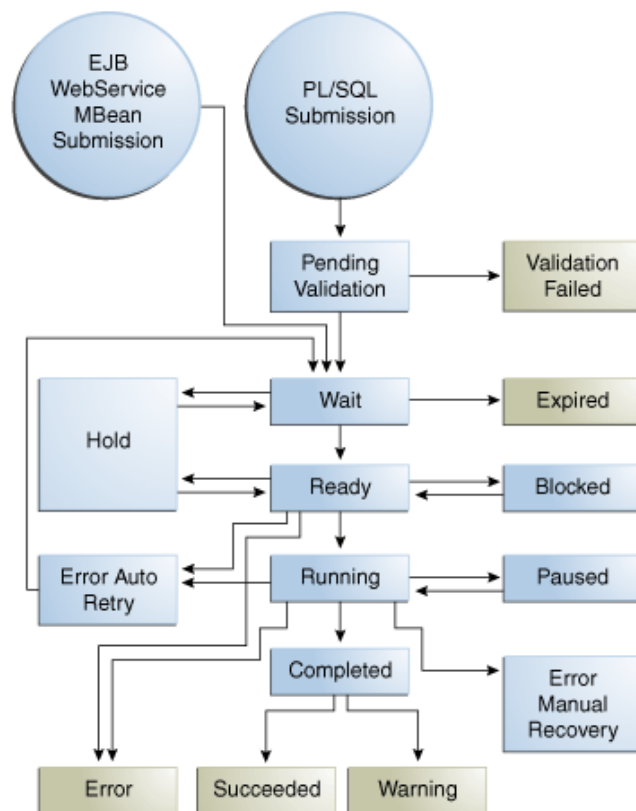


Figure A-3 displays the changes in state for an executable job request for which the executing user has canceled the request.

Figure A-3 Changes in Job State Following Cancellation

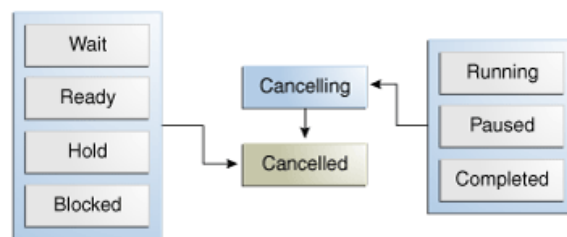
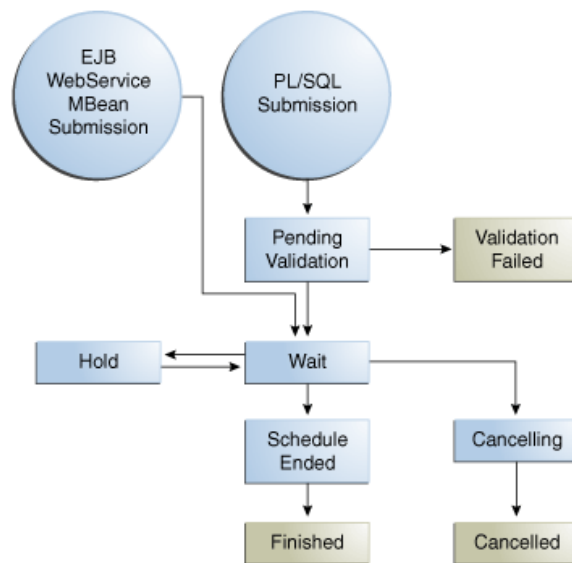


Figure A-4 displays state transitions for a job request submitted with a schedule.

Figure A-4 State Transitions for a Job Request Submitted with a Schedule

A.2.4 Oracle Enterprise Scheduler Life Cycle Tools

As Oracle Enterprise Scheduler runs on an Oracle WebLogic Server instance, you can manage Oracle Enterprise Scheduler using Oracle Fusion Middleware Node Manager for SOA.

Oracle Enterprise Scheduler jobs can be hosted on the same Oracle WebLogic Server instance (or a remote Oracle WebLogic Server instance), database and binary processes. Oracle Enterprise Scheduler controls the life cycle of Oracle Enterprise Scheduler jobs. Use Oracle Enterprise Manager to monitor and manage Oracle Enterprise Scheduler jobs. For more information about managing Oracle Enterprise Scheduler jobs, see [Managing Oracle Enterprise Scheduler Requests](#).

For more information about Oracle Fusion Middleware Node Manager, see the chapter "Using Node Manager" in *Oracle Fusion Middleware Node Manager Administrator's Guide for Oracle WebLogic Server*.

A.3 Configuring High Availability for Oracle Enterprise Scheduler

In order to enable a highly available environment, it is recommended to run Oracle Enterprise Scheduler in a cluster of at least two nodes.

This section includes the following topics:

- [Oracle Enterprise Scheduler Configuration and Deployment Artifacts](#)
- [Oracle Enterprise Scheduler Logging](#)
- [Oracle Enterprise Scheduler Cluster Architecture](#)
- [Failover Requirements](#)
- [Scalability](#)
- [Backup and Recovery](#)
- [Load Balancing](#)

A.3.1 Oracle Enterprise Scheduler Configuration and Deployment Artifacts

Configuration files are as follows:

- **ess.xml:** This file is part of the Oracle Enterprise Scheduler EAR file deployed to the Oracle Enterprise Scheduler cluster.
- **MDS repository:** The Oracle Metadata repository stores Oracle Enterprise Scheduler job metadata. Oracle Enterprise Scheduler supports both database and file-based MDS repositories.

Deployment artifacts are as follows:

- J2EE application for core run-time and hosting applications.
- Job metadata within Oracle MDS for core runtime and jobs loaded at startup.

The Oracle WebLogic Server deployment is non-staged.

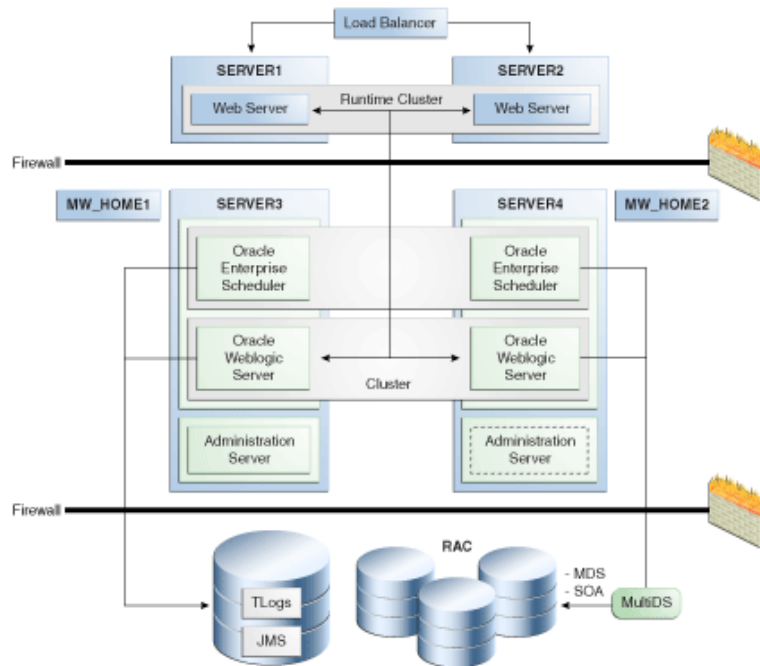
A.3.2 Oracle Enterprise Scheduler Logging

Use standard Oracle WebLogic Server logging for an Oracle Enterprise Scheduler cluster. Use logs in Oracle WebCenter Content to examine Oracle Enterprise Scheduler behavior. Oracle Enterprise Scheduler logging is configured by default in Oracle WebLogic Server.

The default location for log files for Oracle Enterprise Scheduler process jobs on UNIX servers is `/tmp/ess/requestFileDirectory`. Oracle Enterprise Scheduler operational log files can be found under `<DOMAIN_HOME>/servers/<SERVER_HOME>/logs/<server name>-diagnostic.log`, and `<MW_HOME>\user_projects\domains\<DOMAIN_HOME>\servers\<SERVER_HOME>\logs\<server name>-diagnostic.log` on Windows.

A.3.3 Oracle Enterprise Scheduler Cluster Architecture

[Figure A-5](#) is an architectural diagram of a two node Oracle Enterprise Scheduler cluster.

Figure A-5 Oracle Enterprise Scheduler Two Node Cluster

This configuration includes the following components:

- Hardware load balancer.
- A cluster of Oracle WebLogic Servers running on two servers.
- Two Oracle Fusion Middleware homes running on two servers.
- A two node cluster of Oracle Enterprise Scheduler instances, each running on a different instance of Oracle Fusion Middleware.
- A cluster of Oracle RAC databases with multi-DS configuration is required for availability of run-time and MDS schemas. Multi-DS and Oracle RAC provide for database failure.
- Shared persistence storage
- An HTTP load balancer is required for web service interactions. The HTTP load balancer must be appropriately configured to balance requests to Oracle Enterprise Scheduler web services. Oracle Enterprise Scheduler includes the following web services at the specified URLs:
 - ESSWebService
`http://essHost:essPort/ess/essweb service`
 - ESSAsyncCallback webservice
`http://essHost:essPort/ess-async/essasynccallback`
 - WebServiceJobCallback
`http://essHost:essPort/ess-wsjob/async/callback`

For more information regarding high availability architectures, see “Configuring High Availability for Oracle Fusion Middleware SOA Suite” in the *Oracle Fusion Middleware High Availability Guide*.

A.3.4 Configuring the Oracle Enterprise Scheduler Front End Host and Port

For enterprise deployment, Oracle Fusion Applications have removed frontend `host.port` settings and added a checklist to all middleware components so that they do not rely on using the frontend `host.port`. To enable this EDG requirement, ESSAPP supports explicit configuration for the frontend host and port for its web services.

A.3.4.1 Configuring the ESSAPP Frontend Host and Port

ESSAPP support the following two new configuration properties:

- `ServerURL`

The value of this property value is a string with the following format:

`http://host:port`

It is used at runtime to determine the `ESSWebService` end point address and is published as part of the `ESSWebService` concrete WSDL.

- `CallbackServerURL`

The value of this property is a string with the following format:

`http://host:port`

It is used at runtime to determine the end point address of Oracle Enterprise Scheduler callback web services (including `EssAsyncCallbackService`, and `EssWSJobAsyncCallbackService`). These end point addresses are published as part of the respective WSDLs.

If these ESSAPP properties are not configured, then ESSAPP looks up the frontend host and port configuration at runtime by querying the Oracle WebLogic Server cluster. If this cluster's frontend host and port is also not configured, it uses the local `ess_server` host name and port.

ESSAPP advertises the WSDL URLs for its web services (computed using the above preference order) under its home page at: `http://essHost:essPort/ess`.

It is recommended that Oracle Enterprise Scheduler HA/Cluster administrators configure the `ServerURL` and `CallbackServerURL` properties in ESSAPP as explained in this section. The configuration of the cluster frontend host and port described in [Death Detection and Restart](#) can be done as a second preference with lower priority.

It is recommended that end users of Oracle Enterprise Scheduler web services look up the advertised WSDL URLs for individual Oracle Enterprise Scheduler web services and use them to access Oracle Enterprise Scheduler web services.

A.3.4.2 Configuring the WebLogic Server Cluster Frontend Host and Port

For an Oracle Enterprise Scheduler cluster setup involving two or more Oracle Enterprise Scheduler instances, you must configure the cluster front end host and port in the WebLogic Server Administration Console, as follows:

1. Log in to the WebLogic Server Administration Console, click on the **Clusters** link and select the Oracle Enterprise Scheduler cluster (for example, **ess_cluster**) on the Summary of Clusters page.
2. On the Settings for **ess_cluster** page, click on the **Configuration** tab. On the **General** sub-tab, fill in the **Configure Cluster Address** field with the host and port addresses of the Oracle Enterprise Scheduler instances (for example, **ess_server1_host:port, ess_server2_host:port**, and so on).
3. Click on the **HTTP** sub-tab and configure the **Frontend Host**, **Frontend HTTP Port** and **Frontend HTTPS Port** with the Oracle HTTP Server details.
4. Click on the **Servers** sub-tab and verify that the Server Name Prefix and Oracle Enterprise Scheduler instances are configured.

A.3.5 Failover Requirements

An HTTP load balancer provides load balancing so as to re-route requests in the event of node failure. There are no time out requirements for the load balancer or firewalls, as long as components use persistent connections. Likewise, session state replication and failover are not required.

Load balancing is used for actions such as submitting a job and querying its status using the Oracle Enterprise Scheduler web service interface. This load balancing occurs independently of where the job is scheduled to execute.

As Oracle Enterprise Scheduler does not use JMS, no JMS failover is required. For a remote EJB invocation to an ESS cluster, server affinity must be enabled. The property `weblogic.jndi.enableServerAffinity` must be set to `true` in the context. If `oracle.as.scheduler.request.RemoteConnector` is used, server affinity is set automatically.

This section contains the following topics:

- [Request Processor Failover](#)
- [External Component Failover](#)

A.3.5.1 Request Processor Failover

Oracle Enterprise Scheduler includes a request processor component, which represents a single Managed Server in the Oracle Enterprise Scheduler cluster. Request processors process job requests, such that job execution is connected to one or more request processors.

If all jobs are targeted at a number of request processors, jobs are not dependent on a particular request processor. If a job is targeted at a particular request processor, any jobs tied to that request processor execute only when the Managed Server is available and an active workshift exists for the job.

A.3.5.2 External Component Failover

Oracle Enterprise Scheduler interacts with Oracle Fusion Middleware and other components such as Oracle SOA Suite, Oracle ADF Business Components, and so on. If one of these external components fail, it is possible that any running jobs may fail.

You can prevent external component failure from affecting jobs using proper configuration. [Table A-1](#) lists the external components that may fail, along with the steps to take to prevent failed Oracle Enterprise Scheduler jobs.

Table A-1 Oracle Enterprise Scheduler External Component Failover

External Component	Steps to Prevent Failure
Oracle WebCenter Portal	Integrate with a cluster of Oracle WebCenter Portal service through a load balancer.
Oracle RAC Database	Use multi-DS for Oracle RAC database integration.
Oracle SOA Suite, Oracle ADF, and so on	Configure retries for jobs that depend on these components.

A.3.6 Scalability

Horizontal scalability—adding Managed Servers on different machines—tends to enable better performance than vertical scalability—(adding Managed Servers on the same machine.

Use standard Oracle WebLogic Server cluster scaling methodologies for horizontal scaling. For more information about Oracle WebLogic Server clustering, see the “High Availability for WebLogic Server” chapter in the *Oracle Fusion Middleware High Availability Guide*. You can increase the concurrent processing of jobs within a work assignment by increasing the thread allocation of the request processor (by editing the workshift for the request processor) or by binding the work assignment to more than one request processor. For more information, see [Creating or Editing a Workshift](#) and [Creating or Editing a Work Assignment](#).

A.3.7 Backup and Recovery

Following are the backup and recovery guidelines for various components:

- Components stored on the file system: Product binaries, deployed application EAR files and standard Oracle WebLogic Server files in the domain root.
- Changes to the file system: The file system artifacts change when new EAR files are deployed or when the product is patched.
- Data stored in the database: The database stores all metadata and run-time data.
- Changes to database artifacts: Metadata changes when metadata is created and deployed from Oracle JDeveloper or Fusion Middleware Control. Runtime data changes when jobs are submitted, undergo state changes, and so on.

There is no consistency requirement between the artifacts stored on the file system and those in database. The file system stores EAR files and temporarily stores scheduled job output and log files.

A.3.8 Load Balancing

Per Oracle best practice, use any hardware load balancer to load balance two or more Oracle HTTP servers, which, in turn, load balance the Oracle WebLogic Server Managed Servers in the cluster.

A.4 Managing an Oracle Enterprise Scheduler Cluster

Managing an Oracle Enterprise Scheduler cluster involves starting the cluster, propagating configuration changes throughout the cluster, deploying applications and handling unexpected behavior.

This section contains the following topics:

- [Starting and Stopping the Cluster](#)
- [Propagating Configuration Changes to the Cluster](#)
- [Deploying Applications to the Cluster](#)
- [Failures and Expected Behavior](#)

A.4.1 Starting and Stopping the Cluster

Oracle Enterprise Scheduler uses standard J2EE components. As such, Oracle WebLogic Server determines the startup sequence. Oracle Enterprise Scheduler also allows implementing throttling to prevent surges in load.

When stopping a cluster, Oracle Enterprise Scheduler and all local Java jobs terminate. Oracle Enterprise Scheduler also attempts to stop all local binary jobs. However, asynchronous jobs such as SOA or PL/SQL jobs continue. The asynchronous callback from SOA or Oracle ADF Business Component services cannot be delivered if the entire Oracle Enterprise Scheduler cluster is down.

In the case of an abrupt shutdown, the server attempts to recover any on-going transactions.

A.4.2 Propagating Configuration Changes to the Cluster

There are two types of configuration: container or server, and job metadata configuration. Job metadata is stored in Oracle Metadata Repository. The process of configuring the server is the same as that of configuring a standard Oracle WebLogic Server, as part of platform configuration maintained by the Oracle WebLogic Server Configuration Framework. Job metadata configuration changes are deployed from Oracle JDeveloper.

At the cluster level, any configuration changes are propagated by deploying EAR files or metadata. Configuration data is stored either in the database or an EAR file. You can modify the configuration files in the EAR file, such as `ess-config.xml`, using the MBean browser in Fusion Middleware Control.

Cluster members are independent, sharing only the database. There is no communication among members of a cluster.

A.4.3 Deploying Applications to the Cluster

Applications are deployed to the cluster using standard Oracle WebLogic Server EAR files, using standard J2EE defined Oracle WebLogic Server mechanisms. EAR files can be deployed without restarting the server.

An application deployment includes an EAR file, which contains a JAZN and MAR file. The JAZN file, which contains access privileges for the scheduled jobs, is stored in LDAP under the control of Oracle Authorization Policy Manager. The MAR file contains metadata, and is stored in MDS. Oracle WebLogic Server deploys the application EAR file to all Managed Servers in the cluster.

A.4.4 Failures and Expected Behavior

In the event of failure, the main way to ensure the continuation of job processing is to configure a cluster of Oracle Enterprise Schedulers. If a server fails, another node in the cluster transitions all jobs running on the failed server to the relevant state. Synchronous jobs for example, end in an error state and may be retried depending upon whether retries are configured.

In order to enable high availability for the data tier, use Oracle RAC.

This section contains the following topics:

- [Retries](#)
- [Death Detection and Restart](#)
- [Oracle Java Transaction API Migration and Oracle Java Message Service](#)

A.4.4.1 Retries

You can configure retries for jobs and job time outs for asynchronous jobs. For more information about configuring retries and time-outs for a job, see [Creating a Job Request](#).

A.4.4.2 Death Detection and Restart

In order to enable death detection and recovery, each Oracle Enterprise Scheduler cluster node updates its record in the database every minute. This is called a heartbeat. Other nodes monitor the heartbeat, and when the record does not change for a period of time, the server is assumed to be dead. When a server death is detected, each job running on that server is handled. Synchronous jobs are marked as completed with errors, whereas asynchronous jobs continue to run remotely. If retry has been configured, the job marked as completed with errors restarts. Death detection tends to take about ten minutes.

On death detection, the output and log files of a node must be accessible from another node. As such, the file directory containing the job output and log files must be located on a shared file system. This directory is listed in the file `ess-config.xml`.

A.4.4.3 Oracle Java Transaction API Migration and Oracle Java Message Service

Oracle JTA migration is not necessary for Oracle Enterprise Scheduler.

Oracle Enterprise Scheduler does not use Oracle Java Message Service (JMS) such that JMS recovery is not required.

Troubleshooting Oracle Enterprise Scheduler

This appendix describes common problems that you might encounter when using Oracle Enterprise Scheduler and explains how to solve them.

This appendix includes the following sections:

- [Introduction to Troubleshooting Oracle Enterprise Scheduler](#)
- [Getting Started with Troubleshooting Oracle Enterprise Scheduler Jobs](#)
- [Getting Started with Troubleshooting an Oracle Enterprise Scheduler Cluster](#)
- [Problems and Solutions](#)
- [Using My Oracle Support for Additional Troubleshooting Information](#)

In addition to this chapter, review the *Oracle Fusion Middleware Error Messages Reference* for information about the error messages you may encounter.

B.1 Introduction to Troubleshooting Oracle Enterprise Scheduler

This section provides guidelines and a process for troubleshooting Oracle Enterprise Scheduler.

Using the following guidelines and process focuses and minimizes the time you spend resolving problems.

Guidelines

When using the information in this chapter, Oracle recommends:

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

B.2 Getting Started with Troubleshooting Oracle Enterprise Scheduler Jobs

You may want to troubleshoot some typical issues that can arise when running Oracle Enterprise Scheduler jobs.

- Asynchronous jobs remain in running state indefinitely.
- An asynchronous job hangs or crashes.
- Oracle Enterprise Scheduler is down when the remote scheduled job completes or there are network problems such that Oracle Enterprise Scheduler does not receive the completion status from the remote job.
- An invoked SOA composite returns an error.
- A scheduled job is ready to execute, but does not execute.
- A scheduled job is placed in manual error recovery state where troubleshooting is required.
- Oracle Enterprise Scheduler is throwing errors.
- A scheduled job ends in error.

For troubleshooting Oracle Enterprise Scheduler, use the standard Oracle WebLogic Server system log. For information about viewing job request logs, see [Managing Logging for Oracle Enterprise Scheduler](#)

This section contains the following topics:

- [Troubleshooting Asynchronous Scheduled Jobs](#)
- [Troubleshooting SOA Composite Jobs](#)
- [Troubleshooting Process Jobs](#)
- [Steps for Manual Recovery](#)
- [Job Diagnostics](#)

B.2.1 Troubleshooting Asynchronous Scheduled Jobs

Asynchronous jobs run on separate JVMs, including asynchronous Oracle BI Reporting and Publishing, PL/SQL and Java jobs that invoke asynchronous SOA or ADF Business Components services. When handling asynchronous scheduled jobs, Oracle Enterprise Scheduler depends on the remote job sending a completion status defining the job outcome after running. However, the completion status may not be generated, or it may get lost for any of the following reasons:

- The scheduled job has crashed.
- The job is stuck in a hanging state.
- Oracle Enterprise Scheduler was down when the job completed.
- Network problems.

In any of these cases, an asynchronous job stays in running state indefinitely. As a result, subsequent steps in a job set may not execute, or an incompatible job may be blocked indefinitely.

Oracle Enterprise Scheduler displays information in Fusion Middleware Control that enables you to locate the job on the remote system, including an external identifier.

You can take any of the following actions to troubleshoot an asynchronous job that is stuck in running state:

- Use the remote system to troubleshoot the job and determine the outcome of its execution. For more information, see [Troubleshooting Asynchronous Java Jobs](#) and [Troubleshooting Asynchronous PL/SQL Jobs](#)
- Cancel the job. For more information, see [Viewing Job Request Details](#).
- Configure a timeout for asynchronous jobs. For more information, see [Submitting an Oracle Enterprise Scheduler Job Request](#).

When configuring timeouts for jobs, you can use Fusion Middleware Control to display all jobs that have timed out. However, a job that has timed out is still in running state. You must manually change the state of jobs that have timed out. Status callbacks are still accepted for timed out jobs and the job transitions to completion.

You can also change the status of asynchronous jobs that have not timed out. This might happen if a timeout has not been configured, the completion status was lost and you notice that the job has been running for a long time.

B.2.1.1 Troubleshooting Asynchronous Java Jobs

In the case of asynchronous Java jobs (including jobs that invoke remote Oracle SOA Suite or Oracle ADF Business Components services), the log records are tagged with the ECID. You can view logs across the domain by ECID to troubleshoot the job execution. Oracle ADF Business Components, SOA composites, the web services stack and application log records with the ECID.

For asynchronous SOA jobs, the audit trail for the instance in Fusion Middleware Control can be used to troubleshoot composite execution, as described in [Asynchronous Java SOA Job Remains In RUNNING State](#).

[Example B-1](#) shows a sample log message file the JRF web services stack in `server-diagnostic.log`, with inline comments for each log. This log is stored in the following directories:

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

For more information about configuring parameters for a job, see [Submitting an Oracle Enterprise Scheduler Job Request](#). For more information about viewing the `server-diagnostic.log` file, see [Saving Job Request Logs](#).

Example B-1 Asynchronous job logging messages in server-diagnostic.log

Sending message to JMS queue "oracle.j2ee.ws.server.async.DefaultRequestQueue" for asynchronous processing of service b99d80e5-42aa-423a-9e98-f6f88b8b79dfRequest.

```
[2010-08-30T18:28:13.519-07:00] AdminServer NOTIFICATION []
oracle.j2ee.ws.common.jaxws.JAXWSMessages [tid: ACTIVE.ExecuteThread: '0' for queue:
'weblogic.kernel.Default (self-tuning)'] [userId: <anonymous>]
[ecid: 00001f5kZ6S76EWLHyo2yf1CV5eh000000,0:1] [WEBSERVICE_PORT.name:
EmployeeModuleServiceSoapHttpPort] [APP: ADFBCAsync] [J2EE_MODULE.name: ADFBCAsync-ejb]
[WEBSERVICE.name: EmployeeModuleService] [J2EE_APP.name: ADFBCAsync] [MessageID:
urn:uuid:01994234-4442-4dee-82a6-e1e04407af56]
```

An asynchronous request message is received and successfully recorded for the service EmployeeModuleService with the replyTo address http://adc2180314:7001/ADFBCAsyncCallback/EmployeeModuleServiceCallbackResponseImplService.

```
[2010-08-30T18:28:13.724-07:00] AdminServer NOTIFICATION []
oracle.j2ee.ws.common.jaxws.JAXWSMessages [tid: ACTIVE.ExecuteThread: '0' for queue:
```

```
'weblogic.kernel.Default (self-tuning)'] [userId: <anonymous>]
[ecid: 0000If5kZGS76EWLHyo2yf1CV5eh000000,0:1] [WEBSERVICE\_PORT.name:
EmployeeModuleServiceSoapHttpPort] [APP: ADFBCAsync] [J2EE\_MODULE.name: ADFBCAsync\~ejb]
[WEBSERVICE.name: EmployeeModuleService] [J2EE\_APP.name: ADFBCAsync] [MessageID:
urn:uuid:01994234-4442-4dee-82a6-e1e04407af56]
Unknown macro: {/service/common/}
```

Started asynchronous request processing for the service EmployeeModuleService with the message selector "b99d80e5-42aa-423a-9e98-f6f88b8b79dfRequest". Transaction enabled: "true".

```
[2010-08-30T18:28:13.811-07:00] AdminServer NOTIFICATION []
oracle.j2ee.ws.common.jaxws.JAXWSMessages [tid: ACTIVE.ExecuteThread: '2' for queue:
'weblogic.kernel.Default (self-tuning)'] [userId: OracleSystemUser]
[ecid: 0000If5kZGS76EWLHyo2yf1CV5eh000000,0] [APP: ADFBCAsync] [MessageID:
urn:uuid:01994234-4442-4dee-82a6-e1e04407af56]
```

Completed asynchronous request processing. A response is sent to the client.

```
[2010-08-30T18:28:17.307-07:00] AdminServer NOTIFICATION []
oracle.j2ee.ws.common.jaxws.JAXWSMessages [tid: ACTIVE.ExecuteThread: '2' for queue:
'weblogic.kernel.Default (self-tuning)'] [userId: OracleSystemUser]
[ecid: 0000If5kZGS76EWLHyo2yf1CV5eh000000,0] [APP: ADFBCAsync] [MessageID:
urn:uuid:01994234-4442-4dee-82a6-e1e04407af56]
```

Started asynchronous response processing for the service EmployeeModuleService with the message selector "b99d80e5-42aa-423a-9e98-f6f88b8b79dfResponse".

```
[2010-08-30T18:28:17.330-07:00] AdminServer NOTIFICATION []
oracle.j2ee.ws.common.jaxws.JAXWSMessages [tid: ACTIVE.ExecuteThread: '0' for queue:
'weblogic.kernel.Default (self-tuning)'] [userId: OracleSystemUser]
[ecid: 0000If5kZGS76EWLHyo2yf1CV5eh000000,0] [APP: ADFBCAsync] [MessageID:
urn:uuid:01994234-4442-4dee-82a6-e1e04407af56]
Unknown macro: {/service/common/}
```

Completed asynchronous response processing successfully. The client should have received the response at this point.

```
[2010-08-30T18:28:17.825-07:00] AdminServer NOTIFICATION []
oracle.j2ee.ws.common.jaxws.JAXWSMessages [tid: ACTIVE.ExecuteThread: '0' for queue:
'weblogic.kernel.Default (self-tuning)'] [userId: OracleSystemUser]
[ecid: 0000If5kZGS76EWLHyo2yf1CV5eh000000,0] [APP: ADFBCAsync] [MessageID:
urn:uuid:01994234-4442-4dee-82a6-e1e04407af56]
```

Completed asynchronous response processing with exceptions. The client does not receive any response.

```
[2010-08-31T09:55:33.939-07:00] AdminServer ERROR [] oracle.j2ee.ws.common.jaxws.JAXWSMessages
[tid: ACTIVE.ExecuteThread: '5' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId:
OracleSystemUser] [ecid: 0000If94mKW76EWLHyo2yf1CVJG1000000,0] [APP: j2wpojoasync] [MessageID:
urn:uuid:5b9a5134-1416-4bda-95fd-da6e624466d7]
```

The response is not sent again as the callback service has replied with a SOAP fault. The HTTP response code is 500.

```
[2010-08-31T10:17:35.852-07:00] AdminServer ERROR [] oracle.j2ee.ws.common.jaxws.JAXWSMessages
[tid: ACTIVE.ExecuteThread: '0' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId:
```

OracleSystemUser] [ecid: 0000If99p1A76EWLHyo2yf1CVj000000,0] [APP:
ADFBCAsyncInvalidCallbackCreds] [MessageID: urn:uuid:7e355428-f68f-4c3f-a368-baf529048323]

B.2.1.2 Troubleshooting Asynchronous PL/SQL Jobs

PL/SQL jobs can be identified in the Oracle Database by their job names. You can find the database job names associated with the request on the Job Details page in Fusion Middleware Control. For more information about viewing job request details, see [Viewing Job Request Details](#).

B.2.2 Troubleshooting SOA Composite Jobs

If a SOA composite that was invoked as a web service job is stuck or returns an error, go to the Oracle Enterprise Manager Fusion Middleware Control Request Detail page and navigate to the SOA flow trace.

B.2.3 Troubleshooting Process Jobs

Spawned job processes can be identified by a combination of host name, process ID and process group ID. The Job Details page in Fusion Middleware Control shows this and additional information. For more information about viewing job request details, see [Viewing Job Request Details](#).

B.2.4 Steps for Manual Recovery

Job requests may require manual error recovery for a number of reasons. For example, in the case of an asynchronous job request, the job implementation may not know whether the job request launched successfully and an error manual recovery exception is thrown. When a job request is placed in manual error recovery, the job request must be manually completed. If the job request actually was successfully launched and returns a completion status on completion, this status is ignored.

A job request may be placed in manual error recovery for the following reasons:

- An asynchronous Java job request throws an `ExecutionManualRecoveryException`, which indicates to Oracle Enterprise Scheduler that manual recovery is necessary. The job request is placed in `ERROR_MANUAL_RECOVERY` state. The cause is set to `Cause.PROCESS_MANUAL_RECOVER_ERROR (209)`.
- An asynchronous Java job request throws a run-time exception or an error. If a thrown exception is not handled by the job implementation, Oracle Enterprise Scheduler cannot know whether the remote job was invoked, such that manually recovery is required. The cause is set to `Cause.PROCESS_MANUAL_RECOVER_ERROR (209)`.
- If Oracle Enterprise Scheduler crashes before it has finished initiating an asynchronous job request, it cannot know whether the remote job has been invoked. The job request transitions to `ERROR_MANUAL_RECOVERY`, which holds onto incompatibility locks. The cause is set to `Cause.PROCESS_RECOVER (210)`. If the service crashes after it has finished initiating the asynchronous job and the job has transitioned to running, then the job continues in a running state and complete when the callback occurs.
- A spawned job runs in a clustered environment, the job request runs on the first instance of Oracle Enterprise Scheduler, which goes down, along with the associated spawn agent. If the first instance of Oracle Enterprise Scheduler is not back up and running for a while, Oracle Enterprise Scheduler does not know

whether the spawned process is actually still running. Manual detection and recovery are required. The cause is set to Cause .PROCESS_RECOVER(210).

B.2.4.1 Handling Synchronous Java Jobs Requiring Manual Recovery

If a Java job times out or runs for too long, you can either let it run to completion as usual, or attempt to terminate it.

To recover synchronous Java jobs:

1. If the job is not in the `ERROR_MANUAL_RECOVERY` state, cancel the job. See [Canceling Oracle Enterprise Scheduler Job Requests](#). If the job remains in `CANCELLING` state for an unreasonable amount of time, continue to the next step.
2. Take one of the following actions:
 - If, after some time, the request transitions to a terminal state, no other intervention is required.
 - If the request remains in `CANCELLING` state, then determine the Oracle Enterprise Scheduler server on which the request is running by finding the server and cluster names.

Determine the cluster name from the Oracle Enterprise Scheduler process group. Process group information is shown in search results only when the scope for the job request search is **All Scheduling Services sharing the ESS repository**. See [Searching for Oracle Enterprise Scheduler Job Requests](#) for information on performing searches.
3. If canceling the job is not effective, restart the relevant Oracle Enterprise Scheduler server:
 - a. From the navigation pane, expand the farm and then **WebLogic Domain**.
 - b. Expand the Oracle Enterprise Scheduler cluster and select the Oracle Enterprise Scheduler server.
 - c. In the WebLogic Server home page, from the **WebLogic Server** menu, choose **Control > Shut Down**.
 - d. After the server is shut down, from the WebLogic Server home page, from the **WebLogic Server** menu, choose **Control > Start Up**.
4. In the Request Details page, from the **Action** menu, select **Recover Stuck Request**.

B.2.4.2 Handling Stuck Asynchronous Jobs Requiring Manual Recovery

When an asynchronous job request requires manual recovery, follow these basic steps. Additional steps depend on the job type.

If a request is stuck (marked for manual recovery or taking too long or timed out) and it is an asynchronous remote job, you first check to see if the remote job is still running.

To handle asynchronous jobs:

1. Check to see if the remote job is still running:
 - a. Identify the remote job by navigating to the Request Details page. See [Viewing Job Request Details](#) for information about how to view the Request Details page.

- For PL/SQL jobs, navigate to the Request Details page and click the **Execution Type** icon. The database session information displays, correlating the scheduled job request with the database scheduler job. This indicates whether the job is still executing.
 - For spawned jobs, navigate to the Request Details page, click the **Execution Type** icon to display the spawned process information. This correlates the scheduled job request with the external job, in this case the operating system process. This correlation indicates whether the job is still executing.
- b. Verify that the remote job is no longer running.
 - If the remote job was not successfully created on the remote system, set the status of the job request to **ERROR**.
 - If the remote job was created and has finished executing, determine its status and set the status of the job request accordingly.
 - If the remote job instance has not finished executing, wait until it completes and set the job request status accordingly.
2. After the remote job is no longer running, terminate the job request in Oracle Enterprise Scheduler, so that Oracle Enterprise Scheduler is no longer is keeping track of the job.
 - a. Navigate to the Search request page by clicking the **Scheduling Service** menu and selecting **Search Job Request**.
 - b. From the **Quick Search** list, for asynchronous jobs marked for manual recovery (**ERROR_MANUAL_RECOVERY**), select **Asynchronous requests that need manual recovery**. Requests are in the **RUNNING** state. For asynchronous jobs not in the **ERROR_MANUAL_RECOVERY** state, rather than search with the **Asynchronous requests that need manual recovery** option, search for the known requests that must be updated.
 - c. In the search results, click the request ID to display the Request Details page.
 - d. In the Request Details page, from the **Action** menu, choose **Recover Stuck Request**.
 - e. In the Recover Stuck Request dialog box, set the state accordingly for the job request. Optionally, add a description for the status of the job request.
If you set the status to **ERROR**, the description you add displays in the Request Details page.

B.2.5 Job Diagnostics

A scheduled job may not execute for a number of reasons, or it may fail. Either way, Fusion Middleware Control provides built-in diagnostics in the Job Details page. For jobs that fail with an error, the Job Details page displays the reason and provides access to the job request log.

Fusion Middleware Control provides the following:

- Access logs for the job request.
- Database session information for PL/SQL job requests, shown in the Job Details page for a PL/SQL job request.

- Spawned process information for process job requests, shown in the Job Details page for process job requests.
- A message displays in the Job Details page for job requests in wait, ready or blocked state specifying the reason the job request is in that state.
- Error and warning messages display in the Job Details page. Additional details also display, such as stack traces and so on.
- For retried job requests, the Job Details page displays the number of times the job request was retried, the time of the next retry and the number of additional times the job request is to be retried in the event of an error.
- For job requests that require manual recovery or have timed out, the Job Details page displays a message regarding a requirement for manual recovery.

For more information about viewing job request details, see [Viewing Job Request Details](#).

Table B-1 shows the associated diagnostic codes for each state along with a description and additional information that is provided. If a request is in a state that does not appear in the table, its diagnosis contains only the request state.

Table B-1 Job Request States and Associated Diagnostic Codes

State	Diagnostic Code	Description	Related Documentation
BLOCKED	BLOCKED	Blocked due to incompatible job request or requests. Includes the request ID of the blocking request.	For information about canceling a job request, see Canceling Oracle Enterprise Scheduler Job Requests .
COMPLETED	POSTPROCESS_DELAY	The job request is delayed by the post-processor. Includes the time at which the delay ends.	
PAUSED	PAUSED	The job request is the parent of one or more subrequests and has been paused. Includes the request ID of a subrequest in a non-terminal state, if there is one.	For information about resuming a paused job request, see Holding and Resuming Oracle Enterprise Scheduler Job Requests .
READY	NO_ACTIVE_SERVER	No server is active in the process group. Includes the name of the process and isolation groups.	For information about activating an Oracle Enterprise Scheduler, see Starting and Stopping Oracle Enterprise Scheduler Components .
READY	REQUESTED_PROCESSOR_NOT_ACTIVE	The server specified by the job request <code>SYS_requestedProcessor</code> property is not available. Includes the name of the requested processor.	-
READY	NO_APPLICATION	The application is either not deployed or not active. Includes the name of the application, process group and isolation group.	-

Table B-1 (Cont.) Job Request States and Associated Diagnostic Codes

State	Diagnostic Code	Description	Related Documentation
READY	PROCESSOR_STOPPED	The request cannot be processed because there is no server with the application deployed with an active processor.	For information about starting a request processor, see Starting and Stopping a Request Processor or Dispatcher .
READY	PROCESSOR_FAILED	The request cannot be processed because the processor has failed on all servers to which the application is deployed.	-
READY	PROCESSOR_WAIT	The job request is waiting for an available processor thread. Includes the name of a work assignment that could process the job request.	-
READY	INACTIVE_WORK_ASSIGNMENT_WAIT	Waiting for a work assignment to become active. Includes the name of an inactive work assignment that could process the job request if it were active.	-
READY	NO_LOADED_WORK_ASSIGNMENT	There is a bound work assignment that could process the request, but the binding is not loaded. The server may be down. Includes the name of a work assignment that could process the request but is not loaded.	For information about activating an Oracle Enterprise Scheduler, see Starting and Stopping Oracle Enterprise Scheduler Components .
READY	NO_BOUND_WORK_ASSIGNMENT	You must bind a specialized work assignment to the job request.	For information about binding a work assignment, see Configuring a Request Processor .
READY	THROTTLED	The job request is asynchronous and the number of active asynchronous jobs of the same type is at the allowed limit. Includes the work assignment, workshift, the asynchronous job type and the asynchronous limit.	Change the number of threads allocated to jobs, or the asynchronous job limits for the work shift. For more information about editing a work shift, see Creating or Editing a Work Assignment .
READY	DISABLED_WORK_ASSIGNMENT	There is a bound work assignment that could process the request, but the work assignment is disabled. Includes the name of the disabled work assignment.	Enable the work assignment. For more information about editing a work assignment, see Creating or Editing a Work Assignment .
RUNNING	PREPROCESS_DELAY	The request is delayed by the pre-processor. Includes the time at which the delay ends.	-

Table B-1 (Cont.) Job Request States and Associated Diagnostic Codes

State	Diagnostic Code	Description	Related Documentation
RUNNING	TIMED_OUT	The job request has timed out. Typically, this code displays for timed out asynchronous Java job requests.	-
RUNNING	NO_ACTIVE_SERVER	The job request processing cannot continue because no server is active in the process group. Includes the name of the process and isolation groups.	Verify that the Oracle Enterprise Scheduler server is running (see Monitoring Oracle Enterprise Scheduler Request Activity). If necessary, restart one of the Oracle Enterprise Scheduler components, as described in Starting and Stopping Oracle Enterprise Scheduler Components .
RUNNING	REQUESTED_PROCESSOR_NOT_ACTIVE	Job request processing cannot continue because the server specified by the job request SYS_requestedProcessor property is not available. Includes the name of the requested processor.	Verify that the Oracle Enterprise Scheduler server is running (see Monitoring Oracle Enterprise Scheduler Request Activity). If necessary, restart one of the Oracle Enterprise Scheduler components, as described in Starting and Stopping Oracle Enterprise Scheduler Components .
RUNNING	NO_APPLICATION	Job request processing cannot continue because the application is either not deployed or not active. Includes the name of the application, process group and isolation group.	-
RUNNING	PROCESSOR_STOPPED	Job request processing cannot continue because there is no server with the application deployed with an active processor.	Verify that the Oracle Enterprise Scheduler server is running (see Monitoring Oracle Enterprise Scheduler Request Activity). Start a request processor, as described in Starting and Stopping a Request Processor or Dispatcher .

Table B-1 (Cont.) Job Request States and Associated Diagnostic Codes

State	Diagnostic Code	Description	Related Documentation
RUNNING	PROCESSOR_FAILED	Job request processing cannot continue because the processor has failed on all servers to which the application is deployed.	Verify that the Oracle Enterprise Scheduler server is running (see Monitoring Oracle Enterprise Scheduler Request Activity). Restart the request processor, as described in Starting and Stopping a Request Processor or Dispatcher .
RUNNING	PROCESSOR_WAIT	Waiting for an available processor thread to handle the update event for an asynchronous Java job request. Includes the name of a work assignment that could process the request; indicates whether the job request has disabled update events.	-
RUNNING	INACTIVE_WORK_ASSIGNMENT_WAIT	Waiting for a work assignment to become active to handle the update event for an asynchronous Java request. Includes the name of an inactive work assignment that could process the request if it were active; indicates whether the job request has disabled update events.	Activate the inactive work assignment so that the job request can be processed. For more information, see Creating or Editing a Work Assignment .
RUNNING	NO_LOADED_WORK_ASSIGNMENT	Processing of an asynchronous Java job request may be delayed. There is a bound work assignment that could process the update event, but the binding is not loaded. The server may be down. Includes the name of a work assignment that could process the job request but is not loaded; indicates whether the job request has disabled update events.	Verify that the Oracle Enterprise Scheduler server is running (see Monitoring Oracle Enterprise Scheduler Request Activity). If necessary, restart one of the Oracle Enterprise Scheduler components, as described in Starting and Stopping Oracle Enterprise Scheduler Components .
RUNNING	NO_BOUND_WORK_ASSIGNMENT	Processing of an asynchronous Java job request may be delayed. There is no bound work assignment that can handle the update event for the asynchronous Java job request; indicates whether the request has disabled update events.	Bind a work assignment to the request processor. For more information, see Configuring a Request Processor .

Table B-1 (Cont.) Job Request States and Associated Diagnostic Codes

State	Diagnostic Code	Description	Related Documentation
RUNNING	DISABLED_WORK_ASSIGNMENT	Processing of an asynchronous Java job request may be delayed. There is a bound work assignment that could process the update event for the job request, but the work assignment is disabled. Includes the name of the disabled work assignment; indicates whether the job request has disabled update events.	Enable the work assignment. For more information about editing a work assignment, see Creating or Editing a Work Assignment .
RUNNING	ASYNC_WAIT_COMPLETION	The job request is an asynchronous request whose execution has been initiated. Oracle Enterprise Scheduler is waiting for a notification that the asynchronous executable has completed.	The request might need to be recovered manually if the remote executable has completed. For information about asynchronous job requests, see Troubleshooting Asynchronous Scheduled Jobs .
WAIT	FUTURE_START	The job request has a scheduled time in the future. Includes the scheduled time for the job request.	-
WAIT	RETRY_DELAY	An error occurred in the job request, and the job request has been delayed before being retried. Includes the scheduled retry time.	-
WAIT	NO_APPLICATION	The job request scheduled time has been reached, but the job request cannot be dispatched because the application is not available (is either not deployed or not active). Includes the name of the application, process group and isolation group.	-
WAIT	PARENT_NOT_PAUSED	The job request is a subrequest whose parent has not paused. The subrequest remains in WAIT state until its parent pauses. Includes the parent request ID.	For information about pausing a paused job request, see Holding and Resuming Oracle Enterprise Scheduler Job Requests .
WAIT	DEFERRED	The job request is an instance of a recurrence for which the previous recurrence instance is still running. Oracle Enterprise Scheduler prevents concurrent execution of recurrent instances, and the next recurrence remains in WAIT state while the current recurrence is active.	-

Table B-1 (Cont.) Job Request States and Associated Diagnostic Codes

State	Diagnostic Code	Description	Related Documentation
WAIT	DISPATCHER_STOPPED	The job request cannot be dispatched because there is no server with the application deployed with a running dispatcher.	Restart the request dispatcher. For more information, see Starting and Stopping a Request Processor or Dispatcher
WAIT	DISPATCHER_FAILED	The job request cannot be dispatched because the dispatcher has failed on all servers to which the application is deployed.	Restart the request dispatcher or other Oracle Enterprise Scheduler components. For more information, see Starting and Stopping a Request Processor or Dispatcher and Starting and Stopping an Oracle Enterprise Scheduler Service Instance .
WAIT	WAIT_QUEUE_MISSING	The request has no wait queue entry when one was expected.	-
terminal state	TERMINAL	The job request is in a terminal state (SUCCEEDED, WARNING, ERROR, CANCELLED, EXPIRED, VALIDATION_FAILED, FINISHED). Includes the application, process group, isolation group, work assignment and workshift.	-

Note:

For more information about job states, see *Oracle Fusion Middleware Java API Reference for Oracle Enterprise Scheduler*.

B.3 Getting Started with Troubleshooting an Oracle Enterprise Scheduler Cluster

Troubleshooting an Oracle Enterprise Scheduler clustered environment may involve finding performance and scalability issues, using a shared database or tuning Oracle Enterprise Scheduler system performance.

Troubleshooting an Oracle Enterprise Scheduler clustered environment involves the following.

- [Finding Performance and Scalability Issues](#)
- [Using a Shared Database](#)
- [Tuning Oracle Enterprise Scheduler System Performance](#)

B.3.1 Finding Performance and Scalability Issues

It is possible to detect any issues with performance and scalability problems by viewing performance metrics in Fusion Middleware Control. Metrics include Oracle WebLogic Server, JVM-level metrics and plots, as well as Oracle Enterprise Scheduler level metrics.

The system-level tools specific to the operating system running on the server can be used as an additional diagnostic tool. System level tools indicate how machine resources are utilized at various times, such as network, memory, CPU, I/O, and so on. Such system tools are especially useful for tuning job implementations. Database tools enable identifying problems in the database. For remote jobs, such as ADF Business Components or SOA Java jobs running on a remote system, you can use the corresponding Fusion Middleware Control and system level tools on those servers.

Fusion Middleware Control provides the following types of Oracle Enterprise Scheduler metrics to help identify problems:

- Completed job request statistics by job name. Shows run count, run time, success rate and last run job request statistics for completed requests by job name.
- Completed job request statistics by user. Shows completed request count and run time for completed requests by user.
- Completed job request statistics by work assignment. Shows wait time, processing time, completed and failed count for completed job requests by work assignment.
- Completed job request count by status. Displays completed job requests in a variety of terminal states.

For more information about metrics, see [Monitoring Key Performance Metrics for Oracle Enterprise Scheduler..](#)

B.3.2 Using a Shared Database

A common database can be used across multiple application domains. In this case, the database may be loaded from multiple sources. To help with this, Oracle Enterprise Manager allows you to see running and waiting jobs, as well as metrics across the database from multiple Oracle Enterprise Scheduler systems.

B.3.3 Tuning Oracle Enterprise Scheduler System Performance

The following potential performance and scalability issues may occur in the context of job requests or Oracle Enterprise Scheduler runtime.

- Jobs are saturating the CPU of Oracle Enterprise Scheduler servers.
- Jobs are overloading the remote systems where asynchronous jobs are running.
- Ready jobs are filling the queue, despite the availability of spare CPU power, such that job output is delayed.
- Multiple domains are sharing a database. A great number of concurrently running database-intensive jobs across domains are slowing down the database.
- Performance and scalability are affected by the running of large jobs at the end of a financial quarter or month.

- Performance and scalability are affected by the concurrent running of two or more jobs that interact very intensively with the database.

Tuning involves changing job implementations or changing schedules, Oracle Enterprise Scheduler cluster size, processor bindings for work assignments, throttling and thread limits.

B.3.3.1 Tuning Clusters

Clusters are the basic mechanism for enabling scalability and high availability. When a job runs, it is equally likely to run on any processor on which it is eligible to run at that time. By carefully controlling the size of the Oracle Enterprise Scheduler cluster, it is possible to better distribute job executions across the cluster so that servers do not become overloaded. In the case of remote jobs, the jobs actually execute on a remote system and consume very few resources in the cluster (except for a blocked thread for synchronous jobs). If jobs running locally are overloading the system, the first step is to revisit the cluster size configuration.

B.3.3.2 Processor Bindings

Some jobs can physically execute only on a given server; these jobs have been bound to run only on that server. If too many jobs are bound to a particular processor, the benefits of a clustered environment are effectively moot. For the purposes of high availability, avoid tying a job to just one server while enabling the job to run on at least two servers. The job does not run if the bound processor is down.

Rather than relying on clusters to randomly distribute work, you may have a set of long, resource intensive jobs to be run locally within a given scheduling window. In this case, you can bind jobs to specific processors and explicitly control the distribution of these jobs.

A processor is an Oracle Enterprise Scheduler instance. One Oracle Enterprise Scheduler instance runs on one cluster node. As one cluster node typically runs on a single computer, a processor normally equates to a computer.

At times, a clustered environment runs well until the scheduled periods during which a number of resource intensive jobs run. In order to maintain performance, you can configure the cluster with extra idle nodes that are activated during busy periods so as to handle the extra job load. You use standard Oracle WebLogic Server cluster methodologies to enable this clustering. For more information about Oracle WebLogic Server clustering, see the "High Availability for WebLogic Server" chapter in the *Oracle Fusion Middleware High Availability Guide*.

Job performance can vary depending on whether executing jobs are synchronous or asynchronous. Synchronous jobs consume a single thread throughout their execution, and are normally short lived. (An exception might be a process or spawned job that loads a database.) Asynchronous jobs consume a thread at the beginning and end of execution for a very short time, but they otherwise run independently. Asynchronous jobs are typically long running and continue execution across server restarts. Typical examples of asynchronous jobs are PL/SQL jobs, Java jobs that invoke remote asynchronous ADF Business Components services, and Java jobs that invoke remote asynchronous SOA services.

Throttling limits the maximum number of jobs that may execute concurrently. This is important to avoid flooding the system with too many concurrently running jobs. For synchronous jobs, this limit is imposed by limiting the number of threads available for execution. For PL/SQL jobs and other asynchronous jobs, this limit is imposed by defining a maximum concurrency limit for PL/SQL and asynchronous jobs respectively. Asynchronous throttling limits are set on the work assignment to which

the jobs are assigned. For more information about setting asynchronous job limits, see [Creating or Editing a Workshift](#)

It is possible that all threads configured are used up for synchronous jobs thereby blocking asynchronous jobs from starting. This can be avoided if asynchronous and synchronous jobs are not combined in a single work assignment.

B.3.3.3 Using Job Incompatibility to Manage Performance

You can configure a job incompatibility not only to prevent two incompatible jobs from running, but in order to prevent both intensive jobs from heavily loading the same resource. In order to maintain good performance, you can define an incompatibility for such jobs so that they never run at the same time. For more information about defining a job incompatibility, see [Creating or Editing an Incompatibility](#).

B.3.3.4 Tuning Oracle Enterprise Scheduler for Optimal Performance

You can tune the following Oracle Enterprise Scheduler components:

- Request dispatcher
- Request processor
- Connection pool size
- RDBMS Scheduler

Tuning the Dispatcher

The dispatcher tuning parameters apply to the Oracle Enterprise Scheduler request dispatcher. The request dispatcher manages requests that are awaiting their scheduled execution. The request processor handles the job requests after they have executed.

Parameters are as follows.

- **Dispatcher Enabled:** Indicates whether the request dispatcher is enabled on the Oracle Enterprise Scheduler server. When disabled, that Oracle Enterprise Scheduler server does not dispatch job requests whose scheduled execution time has arrived. By default, this parameter is enabled.
- **Maximum Poll Interval:** Specifies the maximum frequency, in seconds, at which the request dispatcher checks for job requests that are ready to be dispatched. The default value is 15 seconds.

Tuning the Processor

The processor tuning parameters apply to the Oracle Enterprise Scheduler request processor. The request processor manages job requests whose scheduled execution time has arrived, and are ready to execute.

Parameters are as follows.

- **Processor Enabled:** Indicates whether the request processor is enabled on the Oracle Enterprise Scheduler server. If disabled, the Oracle Enterprise Scheduler server does not process requests that are ready to be executed. By default, this parameter is enabled.
- **Maximum Processor Threads:** Specifies the maximum number of threads used to process job requests. This represents the total number of worker threads that might

run concurrently for all active work assignments for the Oracle Enterprise Scheduler server. By default, this parameter is set to 25.

- **Starvation Threshold:** Indicates the wait time, in minutes, before a job request that is ready to be executed is considered starved and eligible to be processed by a starvation worker thread. The starvation worker processes only those job requests that have been ready longer than the starvation time. A starvation worker is not created if the threshold value is equal to zero.

If enabled (meaning the parameter value is greater than zero), a starvation worker thread is created for each active work assignment for the Oracle Enterprise Scheduler server. The Maximum Processor Threads parameter does not apply to starvation workers. By default, the value of this parameter is set to zero, such that no starvation worker is created.

Tuning the Connection Pool Size for the Oracle Enterprise Scheduler Internal Data Source

The connection pool size for the Oracle Enterprise Scheduler internal JDBC data source should be based on the request processor tuning values configured for the Maximum Processor Threads and Starvation Threshold parameters.

The recommended pool size if the Starvation Threshold parameter is disabled (its value is equal to zero) is the number of maximum processor threads plus twenty.

The recommended pool size if the Starvation Threshold parameter is enabled (its value is greater than 0) is the number of maximum processor threads, along with the number of bound work assignments plus twenty.

Tuning the RDBMS Scheduler

The RDBMS scheduler is capable of auto-tuning. To enable auto-tuning, set `job_queue_processes` to 0. Leave `JOB_QUEUE_PROCESSES` to the default value at 1000. For more information about the `JOB_QUEUE_PROCESSES` parameter, see the *Oracle Database Reference*.

B.3.3.5 Tuning Dead Database Connections

Oracle Enterprise Scheduler spawned jobs connect to the database using SQL*Net. If the spawned jobs are canceled, Oracle Enterprise Scheduler kills these processes at the operating system level. It is possible, however, that the database connections used by these processes still exist in the database.

To reduce dead connections in the database, use the `SQLNET.EXPIRE_TIME` configuration option by setting this value to the desired value. For more information about the `SQLNET.EXPIRE_TIME` parameter, see the "Parameters for the sqlnet.ora File" chapter in *Oracle Database Net Services Reference*.

B.4 Problems and Solutions

This section describes common problems and solutions for Oracle Enterprise Scheduler.

It contains the following topics:

- [Job Remains in WAIT State](#)
- [Synchronous Job Continues in RUNNING State for Too Long](#)
- [Asynchronous Jobs Remain in RUNNING State and Do Not Complete](#)

- [Asynchronous Java SOA Job Remains In RUNNING State](#)
- [Asynchronous Java Oracle ADF Business Components Job Remains In RUNNING State](#)
- [Asynchronous PL/SQL Job Remains in RUNNING State](#)
- [Job Does Not Execute at Scheduled Time](#)
- [Asynchronous Java Job Requires Manual-Error Recovery](#)
- [Spawned \(Process Type\) Job Requires Manual Error Recovery](#)
- [Job Remains in CANCELLING State](#)
- [Newly Added Server Is Not Being Utilized or Running Inappropriate Jobs](#)
- [Oracle Enterprise Scheduler Run-Time System Is Throwing Errors](#)
- [Oracle Enterprise Scheduler Is Running Out Of Database Connections](#)
- [Job Queue Full Due to a Hanging Job](#)

In addition to the recommended solutions, consider reviewing [Tuning Oracle Enterprise Scheduler System Performance](#) for tuning tips.

B.4.1 Job Remains in WAIT State

Problem

When a user submits a job, the job can remain in the WAIT state for too long without progressing to the RUNNING state.

Solution

To resolve this problem, verify the current status of Oracle Enterprise Scheduler from Fusion Middleware Control:

1. Verify the request processor and request dispatcher are running:
 - a. From the navigation pane, expand the farm, and then **Scheduling Services**.
 - b. Select the **ESSAPP** application for the appropriate Managed Server.
 - c. In the Scheduling Service home page, in the **Scheduler Components** section, ensure the Request Processor has a status of **Started**.

If it is not running, start it. See [Starting and Stopping a Request Processor or Dispatcher](#).
 - d. In the Scheduling Service home page, in the **Scheduler Components** section, ensure the Request Dispatcher has a status of **Started**.

If it is not running, start it [Starting and Stopping a Request Processor or Dispatcher](#).
2. Verify the ESSAPP application is running:
 - a. From the navigation pane, expand the farm, and then **Scheduling Services**.
 - b. Select the **ESSAPP** application for the appropriate Managed Server.

- c. In the Scheduling Service home page, in the **Scheduler Components** section, ensure the Request Dispatcher has a status of **Started**.
 - d. In the WebLogic Server home page, in the **Deployments** section, ensure the ESSAPP applications is running.
If it is not running, start it. See [Starting and Stopping an Oracle Enterprise Scheduler Service Instance](#).
3. Check if concurrency or threads are configured is too small by looking at processor and work assignment configuration.
- From the **Scheduling Service** menu, choose **Request Processor > Configure** to review the **Thread Count** field in the Configure Request Processor page. See [Configuring a Request Processor](#).
 - From the **Scheduling Service** menu, choose **Work Allocation > Work Assignments** to review the configuration in the Work Assignments page. See [Managing Work Assignments](#).

B.4.2 Synchronous Job Continues in RUNNING State for Too Long

Problem

When the user submits a job, it remains in the RUNNING state for too long.

Solution

The job may be in RUNNING state because the Oracle Enterprise Scheduler server has crashed and recovery has not taken place.

To resolve this problem, determine the current status of Oracle Enterprise Scheduler from Fusion Middleware Control:

1. Verify the request processor is running:
 - a. From the navigation pane, expand the farm and then **Scheduling Services**.
 - b. Select the **ESSAPP** application for the appropriate Managed Server.
 - c. In the Scheduling Service home page, in the **Scheduler Components** section, ensure the **Request Processor** is enabled and is started.
If it is not running, start it. See [Starting and Stopping a Request Processor or Dispatcher](#).
2. Verify the Oracle Enterprise Scheduler server is running and start if it is not running:
 - a. From the navigation pane, expand the farm and then **WebLogic Domain**.
 - b. Select the Oracle Enterprise Scheduler cluster.
 - c. From the WebLogic Cluster page, in the **Servers** section, view the **Status** column to determine if the Oracle Enterprise Scheduler server is running.
 - d. If it shows a status of down (red down arrow) for a server, click the server name in the **Name** column.
 - e. In the WebLogic Server home page, from the **WebLogic Server** menu, choose **Control > Start Up**.

3. Look at the job output to see if the job is making progress. See [Viewing Job Request Details](#).

When an Oracle Enterprise Scheduler server is restarted, synchronous jobs running on that server are transitioned to the ERROR state.

B.4.3 Asynchronous Jobs Remain in RUNNING State and Do Not Complete

Problem

PLSQL and Java jobs that invoke asynchronous SOA services and Java jobs that invoke asynchronous Oracle Application Development Framework (Oracle ADF) Business Component services run on separate Java Virtual Machines (JVMs) or machines. In these cases, Oracle Enterprise Scheduler depends on the remote job sending a completion status at end of processing that defines the job outcome. However this message may never get generated or lost, resulting in the job staying in the RUNNING state. In addition, subsequent steps in a job set may not execute, or an incompatible job may be blocked indefinitely.

Solution

To resolve this issue, follow the actions described in [Troubleshooting Asynchronous Scheduled Jobs](#).

B.4.4 Asynchronous Java SOA Job Remains In RUNNING State

Problem

Jobs that invoke asynchronous SOA services run on separate Java Virtual Machines (JVMs) or machines. In these cases, Oracle Enterprise Scheduler depends on the remote job sending a completion status at end of processing that defines the job outcome. However this message may never get generated due to various reasons.

Solution

To resolve this issue, you must troubleshoot the native job.

To resolve this problem for asynchronous SOA jobs:

1. Search for the request, as described in [Searching for Oracle Enterprise Scheduler Job Requests](#).
2. On the Request Details page, from the **Action** menu, select **Request Log** to view the log message. See [Viewing Job Request Details](#) for further information about the Request Details page.

The Log Message page displays. By default when user navigates to view the logs for a request, only messages that are logged in Oracle Enterprise Scheduler cluster scope are shown. (If the ESSAPP application is not deployed to a cluster the messages that are logged in the Managed Server scope are shown). However, Oracle Enterprise Scheduler propagates the ECID associated with the request across subsystems, such as Oracle SOA Suite and Oracle ADF.
3. Make note of the value in the **ECID** field.
4. From the **Broaden Target Scope** list, select the */farm_name/domain_name* (Oracle **WebLogic Domain**) to view messages across the domain.
5. In the Log Messages page for the Oracle WebLogic Server domain, in the **Selected Targets** section, ensure the search includes the **ECID** field with the value from the Request Details page.

6. Search and view log records for Oracle SOA Suite and the ECID and note any issues. [Setting Oracle Enterprise Scheduler Log Levels](#).
7. View the audit trail for an SOA composite application instance using the ECID:
 - a. From the navigation pane, expand the farm, **SOA**, and then **soa-infra**.
 - b. From the SOA Infrastructure page, click the **Instances** tab.
 - c. In the **Search** section, enter the ECID in the **ECID** field.
 - d. Click **Search** to find the instance with the ECID.
 - e. Select the instance by clicking the ID in the **Instance ID** field from the Instances table.
The Flow Trace page displays.
 - f. View the audit trail for the instance and observe if the composite completed successfully or completed with error. See the "Viewing the Audit Trail and Process Flow of a BPEL Process Service Component" section in the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*. for more information about the Flow Trace window.
8. If the SOA composite is complete and the job is still running in Oracle Enterprise Scheduler, manually complete the job in the Request Details page. For related information, see [Troubleshooting Asynchronous Scheduled Jobs](#).

B.4.5 Asynchronous Java Oracle ADF Business Components Job Remains In RUNNING State

Problem

Jobs that invoke asynchronous Oracle ADF Business Components services run on separate JVMs or computers. In these cases, Oracle Enterprise Scheduler depends on the remote job sending a completion status at end of processing that defines the job outcome. However this message may never get generated due to various reasons.

Solution

To resolve this issue, you must troubleshoot the native job.

To resolve this problem for synchronous Oracle ADF Business Components jobs:

1. Search for the request, as described in [Searching for Oracle Enterprise Scheduler Job Requests](#).
2. On the Request Details page, from the **Action** menu, select **Request Log** to view the log message. See [Viewing Job Request Details](#) for further information about the Request Details page.

The Log Message page displays. By default when user navigates to view the logs for a request, only messages that are logged in Oracle Enterprise Scheduler cluster scope are shown. (If the ESSAPP application is not deployed to a cluster the messages that are logged in the Managed Server scope are shown). However, Oracle Enterprise Scheduler propagates the ECID associated with the request across subsystems, such as Oracle SOA Suite and Oracle ADF.

3. Make note of the value in the **ECID** field.

4. From the **Broaden Target Scope** list, select the */farm_name/domain_name* (**Oracle WebLogic Domain**) to view messages across the domain.
5. In the Log Messages page for the Oracle WebLogic Server domain, in the **Selected Targets** section, ensure the search includes the **ECID** field with the value from the Request Details page and the **Component Name** field.
6. Search and view log records for Oracle ADF Business Components and web services stack for the ECID and note any issues. See the "Viewing and Searching Log Files" section in the *Oracle Fusion Middleware Administrator's Guide*.
7. Observe if the Oracle ADF Business Components completed successfully or completed with error.
8. If the service is complete and the job is still running in Oracle Enterprise Scheduler, manually complete the job in the Request Details page. For related information, see [Troubleshooting Asynchronous Scheduled Jobs](#).

B.4.6 Asynchronous PL/SQL Job Remains in RUNNING State

Problem 1

PL/SQL jobs run on separate machines. In these cases, Oracle Enterprise Scheduler depends on the remote job sending a completion status at end of processing that defines the job outcome. However this message may never get generated due to various reasons.

Solution 1

PL/SQL jobs can be identified in the Oracle Enterprise Scheduler by their job names. Job definition names are available from the Request Details page in the Fusion Middleware Control associated with the request. See [Viewing Job Request Details](#).

To resolve this issue, troubleshoot the native job. For more information, see [Troubleshooting Asynchronous PL/SQL Jobs](#).

Problem 2

Oracle Enterprise Scheduler internally uses the Database Management System (DBMS) scheduler to schedule PL/SQL jobs. In some cases, the DBMS scheduler has not scheduled the job request, even though Oracle Enterprise Scheduler has submitted the job to the DBMS scheduler and set its state to **RUNNING**.

Solution 2

To resolve this issue:

1. Verify the DBMS Scheduler resource usage. See the "Administering Oracle Scheduler" chapter in the *Oracle Database Administrator's Guide*.
2. Change the PL/SQL job throttle limit by configuring PL/SQL job limits. See [Managing Workshifts](#).

B.4.7 Job Does Not Execute at Scheduled Time

Problem

When a job's scheduled time arrives, it does not execute.

Solution

To resolve this problem, view the Request Details page in the Fusion Middleware Control. This page provides built in diagnostics to show what the issue is. For jobs that fails with an error, the Request Details page shows the reason and provides access to the job request log from the **Action** menu. See [Viewing Job Request Details](#).

For more information job diagnostics, see [Job Diagnostics](#).

B.4.8 Asynchronous Java Job Requires Manual-Error Recovery

Problem

A job gets placed in the `ERROR_MANUAL_RECOVERY` state. There are a number of reasons why a job may end up in error manual recovery. For example for an asynchronous job, the job implementation may not know if the job was successfully launched due to an error and throws the error manual recovery exception. For more reasons on why a job can end up in error manual recovery, see [Steps for Manual Recovery](#).

Solution

To resolve this issue, manually update the job status to complete it. See [Handling Synchronous Java Jobs Requiring Manual Recovery](#) and [Handling Stuck Asynchronous Jobs Requiring Manual Recovery](#).

B.4.9 Spawned (Process Type) Job Requires Manual Error Recovery

Problem

A spawned process type job gets placed in the `ERROR_MANUAL_RECOVERY` state.

Solution

To solve this issue, transition the request to a terminal state:

1. Identify the spawned host and process ID in the Request Details page. See [Viewing Job Request Details](#) for more information about the Request Details page.
2. If the process is still running on the host, wait for it to complete or terminate it.
3. When the process is no longer running, recover the request to begin the transition to an error state. It is subject to auto-retries if configured. From Fusion Middleware Control, perform the following steps:
 - a. Search for the request, as described in [Searching for Oracle Enterprise Scheduler Job Requests](#). When selecting a value for **Status**, select **Cancelled** and **Error Manual Recovery**.
 - b. On the Request Details page, from the **Action** menu, select **Recover Stuck Request**. See [Viewing Job Request Details](#) for more information about the Request Details page.

A request in the `CANCELLED` state is put to `CANCELLED` state and a request in `ERROR_MANUAL_RECOVER_STATE` is put to `ERROR` state, specifying an appropriate error message. The error message specified by the user is shown on Request Details page.

- c. Manually update the job status to complete it. See [Handling Synchronous Java Jobs Requiring Manual Recovery](#) and [Handling Stuck Asynchronous Jobs Requiring Manual Recovery](#).

For more information on manually recovering a job, see [Steps for Manual Recovery](#)

B.4.10 Job Remains in CANCELLING State

Problem

Sometimes when a job is canceled, it stays in CANCELLING state and does not get canceled. Results of a cancellation request depends on the stage of processing for the request when the cancel happens and the results of that stage. Many jobs are implemented as asynchronous ADF service invocations. The application infrastructure does not support cancellation of in-flight service requests, and as a result, the job does not cancel as expected. In addition, there are additional reasons why a job may get stuck in CANCELLING state.

- For PL/SQL jobs, Oracle Enterprise Scheduler attempts to kill the RDBMS scheduler job. For spawned process, Oracle Enterprise Scheduler tries to kill the running process. If the job is successfully killed, the request transitions to the CANCELLED state. If the job completes before it can be killed, the state to which the request transitions depends on the result of the job execution. For these types of jobs, this issue should not occur.
- Asynchronous Java job: The request was canceled, but the remote job never contacted Oracle Enterprise Scheduler with its terminal status. This could happen if the job is still executing because either `AsyncCancelable` interface was not implemented or the remote cancel operation did not succeed. It could also happen if the remote system is unable to respond.
- Synchronous Java job: The request was canceled and the job is still executing. This could happen if either `Cancelable()` interface was not implemented or the job's `Executable.execute()` method still did not return after `Cancelable.cancel` was invoked. For more information about the `Executable` interface, see the chapter "Use Case Oracle Enterprise Scheduler Sample Application" in *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Scheduling Service*.
- Job sets (Parent-Child Requests): In cases of job sets, the cancellation operation propagates to all eligible child requests. Until all child requests are completed, the parent request remains in the CANCELLING state.
- SOA Java job: In cases where a job ends in an error, look for the ECID in the Fusion Middleware Control by locating the composite instance and looking at the audit trail and logs tagged with the ECID to see what happened. For more information about finding the ECID, see the "Viewing the Audit Trail and Process Flow of a BPEL Process Service Component" section in the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.
- Oracle ADF Business Component Java Job: In cases where requests end in an error, look for the message ID in the log file for the message ID in the `server-name-diagnostic.log` file in the following directories to see what happened:

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

For more information about viewing log files, see the "Viewing and Searching Log Files" section in the *Oracle Fusion Middleware Administrator's Guide*.

Solution

Manually intervene in the Fusion Middleware Control to complete the job stuck in CANCELLING state.

To recover asynchronous requests that are stuck, determine whether the remote job is still executing. If it is, nothing should be done on the Oracle Enterprise Scheduler side. If the remote job is no longer executing, then perform the following from Fusion Middleware Control to complete the request:

1. Search for the request, as described in [Searching for Oracle Enterprise Scheduler Job Requests](#).
2. On the Request Details page, from the **Action** menu, select **Recover Stuck Request**. See [Viewing Job Request Details](#) for more information about the Request Details page.

For synchronous Java jobs, wait for the job to complete. If the job is irrevocably hung, then the server on which it is executing must be restarted.

B.4.11 Performance and Scalability Goes Down When Two Very Database Intensive Jobs Run at the Same Time

Problem

Performance and scalability goes down when two very database intensive jobs run at the same time.

To determine job performance issues:

1. Use operating-system level or Database Management System (DBMS) tools to check performance metrics across the Oracle Database to ensure the performance bottleneck is isolated to these two jobs running simultaneously and that tuning practices have been attempted.
2. Determine the execution time of the job:
 - a. From the navigation pane, expand the farm, and then **Scheduling Services**.
 - b. Select the **ESSAPP** application for the appropriate Managed Server.
 - c. In the Scheduling Service home page, click the **Top 10 Long Running Job Requests** tab to view the jobs.

Solution

To solve this problem, perform one of two solutions:

- Mark the two very database intensive jobs as incompatible. Sometimes we want to ensure two jobs never run at the same time, not because they are incompatible and would corrupt the system if run simultaneously, but because they heavily load the same resource. In this case, the jobs can be defined to be incompatible so they never run at the same time. See [Managing Incompatibilities](#).
- Schedule the jobs at different times. See [Submitting an Oracle Enterprise Scheduler Job Request](#).

B.4.12 Newly Added Server Is Not Being Utilized or Running Inappropriate Jobs

Problem

A newly added Managed Server does not run jobs as expected or is not being utilized at all.

You see in the job history over time that no jobs are running on this server or you notice in the job history that a particular job is running on this server that should not run.

To view the job history:

1. From the navigation pane, expand the farm, and then **Scheduling Services**.
2. Select the **ESSAPP** application for the appropriate Managed Server.
3. You see in the job history over time that no jobs are running on this server.
4. Search for the request, as described in [Searching for Oracle Enterprise Scheduler Job Requests](#), to view the list of jobs that already have executed.
5. From the **Request ID** column in the Request Search page, click a job to go to the Request Details page for the job.
6. In the **Execution Trail** section of the Request Details page, view the **Dispatcher**, **Processor**, **Work Assignment**, and **Workshift** the job ran on.

Solution

After a new server is added, Oracle Enterprise Scheduler determines if the default work assignment can be used based on how other processors are bound. If it cannot use the default work assignment, it configures the new server to only run the health check service (internal work assignment `ESSInternalWA`). Revisit this default configuration to configure the work assignment binding to this server as desired and removing the internal work assignment after the health check is complete. For more details, see [Expanding an Oracle Enterprise Scheduler Cluster](#).

B.4.13 Oracle Enterprise Scheduler Run-Time System Is Throwing Errors

Problem

Oracle Enterprise Scheduler run-time system is not behaving properly and is throwing errors or encountering problems when processing a job.

Solution

To identify and solve this problem, review the Oracle Enterprise Scheduler system logs to troubleshoot this issue. From Fusion Middleware Control, perform the following:

1. Search for the request, as described in [Searching for Oracle Enterprise Scheduler Job Requests](#).
2. On the Request Details page, from the **Action** menu, select **Request Log** to view the log message. See [Viewing Job Request Details](#) for further information about the Request Details page.
3. To adjust the log levels:
 - a. From the navigation pane, expand the farm, **WebLogic Domain**, Oracle Enterprise Scheduler cluster, and select the Oracle Enterprise Scheduler server (for example, `ess_server1`).
The WebLogic Server home page displays.
 - b. From the **WebLogic Server** menu, select **Logs > Log Configuration** to display the Log Configuration page.

You can configure the Oracle Enterprise Scheduler server loggers for an Oracle WebLogic Server by modifying the `logging.xml` file of that Oracle WebLogic Server. By default, there is no explicit logger entry for the Oracle Enterprise Scheduler and it inherits the logging level and log handlers configured for the parent logger, typically the "oracle" logger or the ("") root logger.

By default, the log messages for the Oracle Enterprise Scheduler server logger can be found in the Oracle WebLogic Server diagnostic log file for that Oracle WebLogic Server. For more information on logging and log levels, see [Setting Oracle Enterprise Scheduler Log Levels](#).

Note:

The logger only shows logs written by Oracle Enterprise Scheduler job running in Oracle WebLogic Server. After Oracle Enterprise Scheduler transfers control of running PL/SQL and C jobs to the PL/SQL or C process, respectively, PL/SQL and C job logging data is not written to the Oracle Enterprise Scheduler logs as they run in a separate process.

B.4.14 Oracle Enterprise Scheduler Is Running Out Of Database Connections

Problem

If Oracle Enterprise Scheduler is running out of database connections, there could be a problem with connection leaks in Oracle Enterprise Scheduler.

Description

See the "Running out of Data Source Connections" section in the *Oracle Fusion Middleware Administrator's Guide*.

B.4.15 Job Queue Full Due to a Hanging Job

Problem

At times, some jobs may not behave as expected. The job queue may become large due to some job spinning or hanging or if a job has memory leaks.

Solution

Here are some typical scenarios:

- Java job goes into an infinite loop and there are other jobs waiting in the queue for it to finish.

To resolve this problem, perform the following with Fusion Middleware Control:

1. From the navigation pane, expand the farm, and then **Scheduling Services**.
2. Select the **ESSAPP** application for the appropriate Managed Server.
3. From the **Scheduling Service** menu, choose **Request Processor > Configure** to review the **Thread Count** field in the Configure Request Processor page. See [Configuring a Request Processor](#).

If the **Thread Count** field is set to 25 synchronous Java jobs, only 25 Java jobs are permitted to be in the **RUNNING** state, and all other Java jobs are in the queue have to wait to be processed.

If some jobs are performing heavy processing or seem to hang, you can isolate such jobs by defining dedicated work assignments to process them on a specific Oracle Enterprise Scheduler server, leaving the other servers to process the rest of the jobs. See [Creating or Editing a Work Assignment](#).

4. Restart the Oracle Enterprise Scheduler server:

- a. From the navigation pane, expand the farm and then **WebLogic Domain**.
- b. Expand the Oracle Enterprise Scheduler cluster and select the Oracle Enterprise Scheduler server.
- c. In the WebLogic Server home page, from the **WebLogic Server** menu, choose **Control > Shut Down**.
- d. After the server is shut down, from the WebLogic Server home page, from the **WebLogic Server** menu, choose **Control > Start Up**.

After restarting the server, the Oracle Enterprise Scheduler moves **RUNNING** jobs to the **ERROR** state and starts processing the next batch of jobs in the queue.

- PL/SQL job goes into infinite loop and never exits. Many other PL/SQL jobs are also submitted.

Cancel the PLSQL job to move the job to **CANCELLED** state. See [Canceling Oracle Enterprise Scheduler Job Requests](#).

- At times, heavy database inserts and updates cause jobs to wait.

For example, consider a PL/SQL procedure that performs 1 million inserts, updates, and deletes on a table. Such a PL/SQL procedure can take about 20 minute to two hours to complete, depending on the DBMS load. If a work Assignment named **wa1** has been created with a PLSQL concurrency limit setting of 25. There are many PL/SQL jobs and each job runs this PL/SQL procedure.

In this case, the Oracle Enterprise Scheduler server processes only 25 PL/SQL jobs concurrently. Since each PL/SQL job is taking one to two hours to complete, the remaining PL/SQL jobs are not in the **WAIT** state. After the completion of a **RUNNING** job, the first job from the **WAIT** queue is picked and is scheduled to run.

B.5 Using My Oracle Support for Additional Troubleshooting Information

You can use My Oracle Support (formerly MetaLink) to help resolve problems.

My Oracle Support contains several useful troubleshooting resources, such as:

- Knowledge Base articles
- Community Forums and Discussions
- Patches and Upgrades
- Certification information

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

Audit Reference for Oracle Enterprise Scheduler

This appendix provides reference information for auditing in Oracle Enterprise Scheduler.

This appendix includes the following sections:

- [About Custom and Standard Audit Reports](#)
- [Audit Events in Oracle Enterprise Scheduler](#)

C.1 About Custom and Standard Audit Reports

The Common Audit Framework in Oracle Fusion Middleware provides a set of standard reports based on your audit records. It also enables you to modify the standard reports and create your own custom audit reports.

This appendix provides details about events that can be audited in Oracle Enterprise Scheduler. Use this information to understand the structure of each event record to develop custom reports.

The following documents provide more information to help you write custom reports:

- Attributes of Audit Reports in Oracle Business Intelligence Publisher in the *Oracle Fusion Middleware Application Security Guide*
- Customizing Audit Reports in the *Oracle Fusion Middleware Application Security Guide*

The following documents provide additional information about how to configure auditing and view standard reports:

- Configuring auditing for Oracle Enterprise Scheduler - See [Auditing Administrative Actions](#).

C.2 Audit Events in Oracle Enterprise Scheduler

This table lists the event categories, events, and the attributes of each event that can be audited in Oracle Enterprise Scheduler.

Table C-1 Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
EssSecurity	MetadataSecurityCheck	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	RuntimeDataCheck	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	DataSecurityCheck	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	DataSecurityQuery	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	UserAssert	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
EssRequestOperation	Submit	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	Execute	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	Cancel	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	Delete	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	Hold	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	Lock	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	PublishEvent	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	Purge	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	Release	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	Update	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	ReplaceSchedule	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
EssMetadataChange	CustomizeJobDef	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	CustomizeJobSet	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	AddJobType	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	CopyJobType	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	UpdateJobType	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	DeleteJobType	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	AddJobDef	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	CopyJobDef	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	UpdateJobDef	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	DeleteJobDef	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	AddJobSet	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	CopyJobSet	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	UpdateJobSet	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	DeleteJobSet	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	AddIncompatibility	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	CopyIncompatibility	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	UpdateIncompatibility	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	DeleteIncompatibility	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	AddSchedule	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	CopySchedule	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	UpdateSchedule	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	DeleteSchedule	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	AddExclusionDefinition	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	CopyExclusionDefinition	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	UpdateExclusionDefinition	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	DeleteExclusionDefinition	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	AddTrigger	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	CopyTrigger	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
-	UpdateTrigger	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId

Table C-1 (Cont.) Oracle Enterprise Scheduler Audit Events

Event Category	Event	Attributes used by Event
-	DeleteTrigger	PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId, PolicyStripe, Subject, PermissionAction, PassedCheckList, FailedCheckList, PermissionCheckResult, Requests, PermissionTarget, TargetUser, PermCheckStripes, ExecStripe, RequestId, RequestParameters, Submitter, SubmitterGuid, RunAsUser, RunAsUserGuid, ParentId, AbsParentId, ExecutionType, EnterpriseId, ExecAttempt, objId
EssConfigChange	EssConfigAdd	-
-	EssConfigUpdate	-
-	EssConfigDelete	-
-	EssConfigUncustomize	-
RequestDispatcher Administration	StartDispatcher	-
-	StopDispatcher	-
RequestProcessor Administration	StartProcessor	-
-	StopProcessor	-

