

## **Oracle® Fusion Middleware**

Administrator's Guide for Oracle Enterprise Scheduling Service

11g Release 1 (11.1.1.6.0)

**E24712-01**

November 2011

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

Oracle Fusion Middleware Administrator's Guide for Oracle Enterprise Scheduling Service, 11g Release 1 (11.1.1.6.0)

E24712-01

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

Primary Authors: Steve Traut, Shelly Butcher

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

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

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

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

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

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

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

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

---

---

# Contents

<b>Preface</b> .....	xi
<b>1 Introduction to Oracle Enterprise Scheduling Service</b>	
1.1 Introduction to Administering Oracle Enterprise Scheduling Service .....	1-1
1.2 Basic Tasks for Configuring and Managing Oracle Enterprise Scheduling Service .....	1-2
1.2.1 Set Up and Configuration.....	1-2
1.2.2 Creating Jobs to Run.....	1-2
1.2.3 Bundling Jobs and Refining Their Work .....	1-2
1.2.4 Monitoring the System.....	1-3
<b>2 Getting Started Managing Oracle Enterprise Scheduling Service</b>	
2.1 Logging into Oracle Enterprise Manager Fusion Middleware Control Console .....	2-1
2.2 Navigating to Oracle Enterprise Scheduling Service Administration Tasks .....	2-2
2.3 The Scheduling Service Home Page.....	2-4
2.4 Logging Out of Oracle Enterprise Manager Fusion Middleware Control .....	2-6
<b>3 Performing Basic Administration Tasks</b>	
3.1 Introduction to Performing Basic Administration Tasks .....	3-1
3.2 Creating a Domain That Includes Oracle Enterprise Scheduling Service Components Using the Configuration Wizard	3-2
3.3 Configuring Perl to Support Process Jobs .....	3-4
3.4 Configuring Oracle Enterprise Scheduling Service .....	3-4
3.4.1 Expanding an Oracle Enterprise Scheduling Service Cluster .....	3-5
3.4.2 Configuring a Request Processor .....	3-5
3.4.3 Configuring a Request Dispatcher .....	3-7
3.5 Starting and Stopping Oracle Enterprise Scheduling Service Components .....	3-7
3.5.1 Starting and Stopping an Oracle Enterprise Scheduling Service Service Instance ....	3-8
3.5.2 Starting and Stopping a Request Processor or Dispatcher .....	3-8
3.6 Managing Application Properties .....	3-8
3.7 Managing Application and Role Policies .....	3-9
3.7.1 Managing Application Roles.....	3-10
3.8 Managing Oracle Enterprise Scheduling Service Web Services .....	3-10
<b>4 Managing Oracle Enterprise Scheduling Service Requests</b>	
4.1 Introduction to Managing Oracle Enterprise Scheduling Service Job Requests.....	4-1

4.2	Managing Oracle Enterprise Scheduling Service Requests .....	4-1
4.2.1	Submitting an Oracle Enterprise Scheduling Service Job Request.....	4-2
4.2.1.1	Creating a Job Request.....	4-2
4.2.1.2	Specifying a Job Request Schedule .....	4-3
4.2.2	Searching for Oracle Enterprise Scheduling Service Job Requests.....	4-3
4.2.2.1	Searching for a Job Request Using Simple Search .....	4-4
4.2.2.2	Searching for a Job Request Using the Advanced Search Feature .....	4-5
4.2.2.3	Configuring the Display of Columns in the Search Results Table .....	4-7
4.2.3	Creating or Editing Predefined Job Schedules .....	4-7
4.2.4	Viewing Job Request Details .....	4-8
4.2.5	Holding and Resuming Oracle Enterprise Scheduling Service Job Requests.....	4-9
4.2.6	Cancelling Oracle Enterprise Scheduling Service Job Requests .....	4-9
4.2.6.1	Initiating Cancellation During Pre-Process Handler Execution.....	4-10
4.2.6.2	Initiating Cancellation During Synchronous Java Job Execution.....	4-10
4.2.6.3	Initiating Cancellation During Asynchronous Java Job Execution.....	4-11
4.2.6.4	Initiating Cancellation During PL/SQL Job Execution .....	4-11
4.2.6.5	Initiating Cancellation During Spawned Job Execution.....	4-11
4.2.6.6	Cancelling Parent Job Requests .....	4-12
4.2.6.7	Cancelling a Child Job Request .....	4-12
4.3	Managing Purge Policies .....	4-12
4.3.1	Setting Up a New Purge Policy .....	4-13
4.3.2	Physically Purging Job Requests from the Database.....	4-14
4.3.3	Viewing Purge Policies .....	4-15
4.3.4	Updating a Purge Policy.....	4-15
4.3.5	Deleting a Purge Policy.....	4-15
4.4	Configuring Simple Data Security for Job Requests.....	4-16
4.4.1	Oracle Simple Data Security Actions.....	4-16
4.4.2	How to Create Data Security Policies for Oracle Enterprise Scheduling Service Components .....	4-17

## 5 Managing the Work of Oracle Enterprise Scheduling Service Jobs

5.1	Introduction to Managing the Work of Oracle Enterprise Scheduling Service Jobs.....	5-1
5.2	Managing Job Metadata .....	5-1
5.2.1	Managing Job Definitions.....	5-2
5.2.1.1	Viewing Job Definitions.....	5-2
5.2.1.2	Creating a Job Definition .....	5-3
5.2.1.2.1	Configuring Parameters .....	5-4
5.2.1.2.2	Configuring System Properties .....	5-4
5.2.1.2.3	Configuring Access Control.....	5-10
5.2.2	Managing Job Sets .....	5-10
5.2.2.1	Viewing Job Sets .....	5-11
5.2.2.2	Creating or Editing a Job Set.....	5-11
5.2.2.3	Deleting a Job Set.....	5-13
5.2.3	Managing Incompatibilities .....	5-13
5.2.3.1	Viewing Incompatibilities .....	5-13
5.2.3.2	Creating or Editing an Incompatibility .....	5-14
5.2.3.3	Deleting an Incompatibility .....	5-15

5.3	Managing Work Assignments and Workshifts .....	5-15
5.3.1	Managing Work Assignments .....	5-16
5.3.1.1	Creating or Editing a Work Assignment.....	5-17
5.3.1.2	Deleting a Work Assignment.....	5-18
5.3.2	Managing Workshifts.....	5-18
5.3.2.1	Creating or Editing a Workshift .....	5-19
5.3.2.2	Deleting a Workshift .....	5-20
5.3.3	Managing Schedules .....	5-20
5.3.3.1	Creating or Editing a Schedule.....	5-21
5.3.3.2	Deleting a Schedule.....	5-21
5.4	Managing Metadata Security for Jobs.....	5-21
5.4.1	Metadata Security Actions .....	5-22
5.4.2	How to Create Metadata Policies for Oracle Enterprise Scheduling Service Resources... 5-22	

## 6 Monitoring Oracle Enterprise Scheduling Service

6.1	Overview of Monitoring Oracle Enterprise Scheduling Service.....	6-1
6.2	Monitoring Oracle Enterprise Scheduling Service Request Activity .....	6-1
6.2.1	Viewing Top Ten Long Running Requests and Top Ten Ready Requests .....	6-2
6.2.1.1	Viewing Top Ten Long Running Oracle Enterprise Scheduling Service Requests .... 6-2	
6.2.1.2	Viewing Top Ten Oracle Enterprise Scheduling Service Jobs Ready to Be Executed 6-3	
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 Scheduling Service Instance	6-4
6.3.1	Viewing a Summary of Oracle Enterprise Scheduling Service Activity.....	6-4
6.3.2	Viewing Current Oracle Enterprise Scheduling Service Activity .....	6-5
6.3.3	Viewing Historical Oracle Enterprise Scheduling Service Reports.....	6-6
6.3.4	Managing Chart Sets .....	6-7
6.3.4.1	Saving a Chart Set.....	6-7
6.3.4.2	Deleting a Chart Set.....	6-7
6.3.4.3	Reordering Charts .....	6-7
6.3.4.4	Reverting to an Old Chart Set.....	6-8
6.3.5	Selecting the Performance Time Period .....	6-8
6.4	Monitoring Key Performance Metrics for Oracle Enterprise Scheduling Service.....	6-8
6.5	Viewing General Oracle Enterprise Scheduling Service Instance Information.....	6-9
6.6	Viewing Scheduler Component Status .....	6-9
6.7	Managing Logging for Oracle Enterprise Scheduling Service .....	6-10
6.7.1	Viewing Log Information for the Domain .....	6-10
6.7.2	Viewing Job Request Logs.....	6-10
6.7.3	Viewing Log Messages for Scheduled Job Requests .....	6-11
6.7.4	Setting Oracle Enterprise Scheduling Service Log Levels .....	6-11
6.7.5	Tracing Oracle Enterprise Scheduling Service Jobs.....	6-12
6.7.6	Saving Job Request Logs.....	6-14

## A High Availability for Oracle Enterprise Scheduling Service

6.8	Introduction to High Availability for Oracle Enterprise Scheduling Service .....	A-1
6.9	Oracle Enterprise Scheduling Service Concepts.....	A-2
6.9.1	Oracle Enterprise Scheduling Service Architecture.....	A-2
6.9.2	Oracle Enterprise Scheduling Service Components .....	A-4
6.9.3	Oracle Enterprise Scheduling Service Life Cycle .....	A-4
6.9.4	Oracle Enterprise Scheduling Service Life Cycle Tools .....	A-6
6.10	Configuring High Availability for Oracle Enterprise Scheduling Service .....	A-6
6.10.1	Oracle Enterprise Scheduling Service Configuration and Deployment Artifacts.....	A-7
6.10.2	Oracle Enterprise Scheduling Service Logging .....	A-7
6.10.3	Oracle Enterprise Scheduling Service Cluster Architecture.....	A-7
6.10.4	Failover Requirements .....	A-8
6.10.4.1	Request Processor Failover .....	A-9
6.10.4.2	External Component Failover .....	A-9
6.10.5	Scalability .....	A-9
6.10.6	Backup and Recovery.....	A-9
6.10.7	Load Balancing.....	A-10
6.11	Managing an Oracle Enterprise Scheduling Service Cluster.....	A-10
6.11.1	Starting and Stopping the Cluster .....	A-10
6.11.2	Propagating Configuration Changes to the Cluster .....	A-10
6.11.3	Deploying Applications to the Cluster .....	A-11
6.11.4	Failures and Expected Behavior .....	A-11
6.11.4.1	Retries.....	A-11
6.11.4.2	Death Detection and Restart .....	A-11
6.11.4.3	Oracle Java Transaction API Migration and Oracle Java Message Service.....	A-12

## B Troubleshooting Oracle Enterprise Scheduling Service

B.1	Introduction to Troubleshooting Oracle Enterprise Scheduling Service.....	B-1
B.2	Getting Started with Troubleshooting Oracle Enterprise Scheduling Service Jobs .....	B-2
B.2.1	Troubleshooting Asynchronous Scheduled Jobs .....	B-2
B.2.1.1	Troubleshooting Asynchronous Java Jobs.....	B-3
B.2.1.2	Troubleshooting Asynchronous PL/SQL Jobs .....	B-4
B.2.2	Troubleshooting Process Jobs .....	B-5
B.2.3	Steps for Manual Recovery.....	B-5
B.2.3.1	Handling Synchronous Java Jobs Requiring Manual Recovery .....	B-5
B.2.3.2	Handling Stuck Asynchronous Jobs Requiring Manual Recovery .....	B-6
B.2.4	Job Diagnostics.....	B-7
B.3	Getting Started with Troubleshooting an Oracle Enterprise Scheduling Service Cluster .....	B-11
B.3.1	Finding Performance and Scalability Issues .....	B-11
B.3.2	Using a Shared Database .....	B-12
B.3.3	Tuning Oracle Enterprise Scheduling Service System Performance.....	B-12
B.3.3.1	Tuning Clusters.....	B-12
B.3.3.2	Processor Bindings .....	B-13
B.3.3.3	Using Job Incompatibility to Manage Performance .....	B-13
B.3.3.4	Tuning Oracle Enterprise Scheduling Service for Optimal Performance .....	B-13
B.3.3.5	Tuning Dead Database Connections .....	B-15

B.4	Problems and Solutions .....	B-15
B.4.1	Job Remains in WAIT State .....	B-16
B.4.2	Synchronous Job Continues in RUNNING State for Too Long.....	B-16
B.4.3	Asynchronous Jobs Remain in RUNNING State and Do Not Complete .....	B-17
B.4.4	Asynchronous Java SOA Job Remains In RUNNING State .....	B-17
B.4.5	Asynchronous Java Oracle ADF Business Components Job Remains In RUNNING State B-18	
B.4.6	Asynchronous PL/SQL Job Remains in RUNNING State .....	B-19
B.4.7	Job Does Not Execute at Scheduled Time .....	B-20
B.4.8	Asynchronous Java Job Requires Manual-Error Recovery.....	B-20
B.4.9	Spawned (Process Type) Job Requires Manual Error Recovery .....	B-20
B.4.10	Job Remains in CANCELLING State .....	B-21
B.4.11	Performance and Scalability Goes Down When Two Very Database Intensive Jobs Run at the Same Time B-22	
B.4.12	Newly Added Server Is Not Being Utilized or Running Inappropriate Jobs .....	B-23
B.4.13	Oracle Enterprise Scheduling Service Run-Time System Is Throwing Errors .....	B-23
B.4.14	Oracle Enterprise Scheduling Service Is Running Out Of Database Connections..	B-24
B.4.15	Job Queue Full Due to a Hanging Job.....	B-24
B.5	Using My Oracle Support for Additional Troubleshooting Information .....	B-25





## List of Tables

4-1	Executable Requests and Cancellation States .....	4-9
4-2	Pre-processor States and States Following Cancellation Initiation .....	4-10
4-3	Returned Synchronous Java Job States and States Following Cancellation Initiation..	4-10
4-4	Returned Asynchronous Java Job States and States Following Cancellation Initiation	4-11
4-5	Returned PL/SQL Job States and States Following Cancellation Initiation .....	4-11
4-6	Returned Spawned Job States and States Following Cancellation Initiation.....	4-12
4-7	Grant Actions for Data Security.....	4-16
5-3	Grant Actions for Metadata Security .....	5-22
5-4	Sample Permission Grants for Security .....	5-22
6-1	Response and Load Statistics .....	6-4
6-2	Performance Statistics .....	6-4
6-9	Oracle Enterprise Scheduling Service External Component Failover.....	A-9



---

---

# Preface

This document describes how to configure and manage Oracle Enterprise Scheduling Service.

Oracle Enterprise Scheduling Service 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 Scheduling Service 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 Scheduling Service.

## 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>.

### Accessibility of Code Examples in Documentation

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

### Accessibility of Links to External Web Sites in Documentation

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

### Access to Oracle Support

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

## Related Documents

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

- *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Scheduling Service*

## Conventions

The following text conventions are used in this document:

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

---

---

# Introduction to Oracle Enterprise Scheduling Service

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

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

This chapter includes the following topics:

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

## 1.1 Introduction to Administering Oracle Enterprise Scheduling Service

With Oracle Enterprise Scheduling Service, 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 Scheduling Service:

- Java job type for executing Java code.
- 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 Scheduling Service. 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 Scheduling Service 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 [Section 1.2, "Basic Tasks for Configuring and Managing Oracle Enterprise Scheduling Service"](#).

## 1.2 Basic Tasks for Configuring and Managing Oracle Enterprise Scheduling Service

You administer Oracle Enterprise Scheduling Service 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).

The following provides a summary of steps you might take to get set up use the service:

### 1.2.1 Set Up and Configuration

1. Create a domain and configure a cluster. Oracle Enterprise Scheduling Service is installed by a product that includes it as a service to manage schedules. However, it's possible that the embedding product didn't configure domains that support Oracle Enterprise Scheduling Service for deploying. See [Section 3.2](#).
2. Configure a request processor and request dispatcher. A job request processor is bound to a particular Oracle Enterprise Scheduling Service server, and is responsible for allocating threads for job requests. A job request dispatcher polls for job requests. See [Section 3.4](#).
3. Start the instance. See [Section 3.5](#).

### 1.2.2 Creating Jobs to Run

1. Create job definitions. You define and manage job definitions that specify values specific to the request at hand. See [Section 5.2.1](#).
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 [Section 4.2.1.2](#), [Section 4.2.3](#) and [Section 5.3.3](#).
3. Create and submit job requests. A job request pulls together a job definition with a schedule on which the job will run. See [Section 4.2.1](#).
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 [Section 4.3](#).

### 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 [Section 5.2.2](#).
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 [Section 5.2.3](#).
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 [Section 5.3.1](#).

## 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 [Chapter 6, "Monitoring Oracle Enterprise Scheduling Service"](#).





---

---

# Getting Started Managing Oracle Enterprise Scheduling Service

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

This chapter includes the following topics:

- [Logging into Oracle Enterprise Manager Fusion Middleware Control Console](#)
- [Navigating to Oracle Enterprise Scheduling Service 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 Scheduling Service 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**.

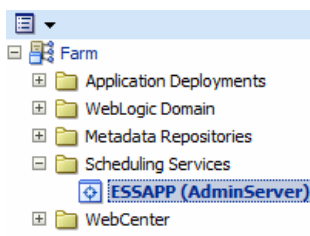
The farm home page is displayed. From there, you can navigate to the Scheduling Service.

## 2.2 Navigating to Oracle Enterprise Scheduling Service Administration Tasks

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

To navigate to Oracle Enterprise Scheduling Service pages in Fusion Middleware Control:

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



The Scheduling Service home page displays. See [Section 2.3](#) 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 <a href="#">Section 2.3</a> .
Control	<p>Displays the following options:</p> <ul style="list-style-type: none"> <li>▪ <b>Start Up:</b> This option runs the Oracle Enterprise Scheduling Service server. For more information, see <a href="#">Section 3.5</a>.</li> <li>▪ <b>Shut Down:</b> This option shuts down the Oracle Enterprise Scheduling Service server. For more information, see <a href="#">Section 3.5</a>.</li> </ul>
Logs	<p>The <b>View Log Messages</b> 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.</p> <p>The log files are stored in the following directories:</p> <ul style="list-style-type: none"> <li>▪ <code>DOMAIN_HOME/servers/SERVER_HOME/logs</code></li> <li>▪ <code>DOMAIN_HOME/servers/SERVER_HOME/logs/apps</code></li> <li>▪ <code>DOMAIN_HOME/servers/SERVER_HOME/logs/owsm/msglogging/diagnostic.log</code></li> <li>▪ <code>DOMAIN_HOME/servers/SERVER_HOME/sysman/log/emoms.log</code></li> </ul>
Job Requests	<p>Displays the following options:</p> <ul style="list-style-type: none"> <li>▪ <b>Search Job Requests:</b> Allows you to search for job requests according to particular parameters. For more information, see <a href="#">Section 4.2.2</a>.</li> <li>▪ <b>Submit Job Requests:</b> Allows you to submit an execution request for a particular job. For more information, see <a href="#">Section 4.2.1</a>.</li> </ul>
Performance	<p>Displays the following options:</p> <ul style="list-style-type: none"> <li>▪ <b>Service Summary:</b> This option displays a brief overview of the service performance. For more information, see <a href="#">Section 6.3.1</a>.</li> <li>▪ <b>Current Activity:</b> This option displays current Oracle Enterprise Scheduling Service activity. For more information, see <a href="#">Section 6.3.2</a>.</li> <li>▪ <b>Historical Reports:</b> This option displays a chart or table depicting previous Oracle Enterprise Scheduling Service data. For more information, see <a href="#">Section 6.3.3</a>.</li> </ul>
Request Processor	<p>Displays the following options:</p> <ul style="list-style-type: none"> <li>▪ <b>Start/Stop:</b> This option causes the request processor to start or stop running. For more information, see <a href="#">Section 3.5</a>.</li> <li>▪ <b>Configure:</b> 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 <a href="#">Section 3.4</a>.</li> </ul>
Request Dispatcher	<p>Displays the following options:</p> <ul style="list-style-type: none"> <li>▪ <b>Start/Stop:</b> This option causes the request dispatcher to start or stop running. For more information, see <a href="#">Section 3.5</a>.</li> <li>▪ <b>Configure:</b> 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 <a href="#">Section 3.4</a>.</li> </ul>

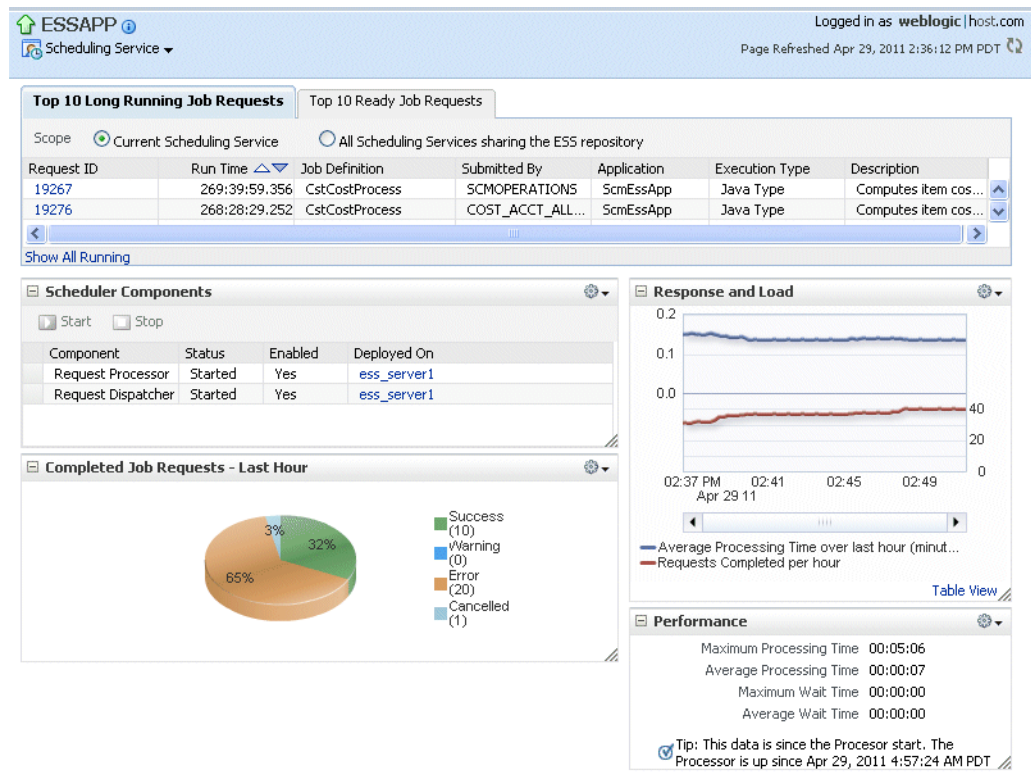
Menu Option	Description
Purge Policies	Allows you to configure and manage job request purge policies including job request purge criteria, retention criteria and purge policy schedules. For more information, see <a href="#">Section 4.3</a> .
Work Allocation	Displays the following options: <ul style="list-style-type: none"> <li>▪ <b>Work Assignments:</b> This option displays the Work Assignments page which allows you to create, edit and delete work assignments. For more information, see <a href="#">Section 5.3.1</a>.</li> <li>▪ <b>Workshifts:</b> This option displays the Workshifts page which allows you to create, edit and delete workshifts. For more information, see <a href="#">Section 5.3.2</a>.</li> <li>▪ <b>Schedules:</b> 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 <a href="#">Section 5.3.3</a>.</li> </ul>
Job Metadata	Displays the following options: <ul style="list-style-type: none"> <li>▪ <b>Job Definitions:</b> This option displays the Job Definitions page, which allows you to manage job definitions. See <a href="#">Section 5.2.1</a> for further information.</li> <li>▪ <b>Job Sets:</b> This option displays the Job Sets page, which allows you to create and edit job sets for a given application. For more information, see <a href="#">Section 5.2.2</a>.</li> <li>▪ <b>Incompatibilities:</b> This option displays the Incompatibilities page, which allows you to create and edit incompatibilities for a given application. For more information, see <a href="#">Section 5.2.3</a>.</li> </ul>
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 <a href="#">Section 3.8</a> .
Security	Displays the following options: <ul style="list-style-type: none"> <li>▪ <b>Application Policies:</b> This option displays the authorization policies that an application relies upon for controlling access to its resources. See <a href="#">Section 3.7</a> for further information.</li> <li>▪ <b>Application Roles:</b> This option displays the roles used by security aware applications specific to the application. See <a href="#">Section 3.7</a> for further information.</li> </ul>
General Information	Displays general details about the Oracle Enterprise Scheduling Service instance. For more information, see <a href="#">Section 6.5</a> .

## 2.3 The Scheduling Service Home Page

The Scheduling Service home page provides an overview of the performance of Oracle Enterprise Scheduling Service 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 Scheduling Service. [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 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 Scheduling Service repository. For more information, see "Viewing Top Ten Long Running Oracle Enterprise Scheduling Service Requests" in [Chapter 6, "Monitoring Oracle Enterprise Scheduling Service"](#).
- **Scheduler Components Region**

The Scheduler Components region displays the components of Oracle Enterprise Scheduling Service, 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 Top Ten Oracle Enterprise Scheduling Service Jobs Ready to Be Executed" in [Chapter 6, "Monitoring Oracle Enterprise Scheduling Service"](#).
- **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 [Chapter 6, "Monitoring Oracle Enterprise Scheduling Service"](#).
- **Response and Load Region**

The Response and Load region displays performance monitoring statistics regarding the time required to process to job requests. For more information, see

"Viewing Job Request Response and Load" in [Chapter 6, "Monitoring Oracle Enterprise Scheduling Service"](#).

- 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 [Chapter 6, "Monitoring Oracle Enterprise Scheduling Service"](#).

## 2.4 Logging Out of Oracle Enterprise Manager Fusion Middleware Control

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

To log out of Oracle Enterprise Manager Fusion Middleware Control:

1. Note the following details about logging out.
  - If multiple pages are open, 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** link.

---

---

## Performing Basic Administration Tasks

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

This chapter includes the following topics:

- [Introduction to Performing Basic Administration Tasks](#)
- [Creating a Domain That Includes Oracle Enterprise Scheduling Service Components Using the Configuration Wizard](#)
- [Configuring Perl to Support Process Jobs](#)
- [Configuring Oracle Enterprise Scheduling Service](#)
- [Starting and Stopping Oracle Enterprise Scheduling Service Components](#)
- [Managing Application Properties](#)
- [Managing Application and Role Policies](#)
- [Managing Oracle Enterprise Scheduling Service Web Services](#)

### 3.1 Introduction to Performing Basic Administration Tasks

Oracle Enterprise Scheduling Service is installed by a 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 Scheduling Service explicitly during domain creation, or to configure aspects of Oracle Enterprise Scheduling Service before you can get the most out of the service.

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

- Creating a domain that includes Oracle Enterprise Scheduling Service components.
- Configuring Oracle Enterprise Scheduling Service by configuring a cluster, request processor, and request dispatcher. While a cluster is optional (Oracle Enterprise Scheduling Service can run on a single node), a processor and dispatcher are required in order for Oracle Enterprise Scheduling Service 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 Creating a Domain That Includes Oracle Enterprise Scheduling Service Components Using the Configuration Wizard

Oracle Enterprise Scheduling Service is installed with products that embed it to support scheduling product jobs. However, during deployment, the embedding product might not have automatically deployed Oracle Enterprise Scheduling Service. In that case, Oracle Enterprise Scheduling Service needs to be explicitly selected for deployment during domain creation. When this is the case, you can use the following steps to create a domain that supports Oracle Enterprise Scheduling Service applications.

1. Change directory to the location of the Configuration Wizard. This is within the home directory of the product that includes Oracle Enterprise Scheduling Service. Oracle recommends having all database instances up.

```
SOAHOST1> cd ORACLE_HOME/common/bin
```

2. Start the Configuration Wizard.

```
SOAHOST1> ./config.sh
```

3. In the Welcome screen, select **Create a new WebLogic domain**, and click **Next**.

4. In the Select Domain Source screen, do the following:

- Select **Generate a domain configured automatically to support the following products**.
- Select the following products:
  - **Oracle Enterprise Scheduler Service Basic**
  - **Oracle WSM Policy Manager** (this should be selected automatically)
  - **Oracle JRF** (this should be selected automatically)
  - Select the appropriate templates for the product that includes Oracle Enterprise Scheduling Service. Refer to that product's documentation on deploying it.
  - If the embedding product installed Fusion Middleware Control and ESS-EM plugin, then select these templates also.

Click **Next**.

The remaining steps do not describe actions specific to deploying the product that includes Oracle Enterprise Scheduling Service. Refer to that product's documentation for details. The steps here describe actions specific to deploying Oracle Enterprise Scheduling Service.

5. In the Specify Domain Name and Location screen, enter the domain name you want.

Make sure that the domain directory matches the directory and shared storage mount point recommended in *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite*.

Click **Next**.

6. In the Configure Administrator Username and Password screen, enter the username and password to be used for the domain's administrator.

Click **Next**.



7. In the Configure Server Start Mode and JDK screen, for Weblogic Domain Startup Mode, select **Production Mode**.

Click **Next**.

8. In the Configure JDBC Component Schema screen, do the following:
  1. If the database is RAC, select **Convert to GridLink** or **Convert to RAC multi data source** as appropriate, then and click **Next** and configure accordingly. Otherwise, continue to the next step.
  2. Select all the Component Schemas.

Select Driver: \*Oracle's Driver (Thin) for Service connections

For the Oracle Enterprise Scheduling Service Schema and the Oracle Enterprise Scheduling Service MDS, enter the database SID instead of the database service name.

Fill in the host name where the database resides.

Fill in the database listener port.

3. Deselect the two MDS schemas, leaving only the ESS Schema selected. Type the Schema Owner and Schema Password for ESS.
  4. Deselect "ESS Schema" and select "MDS Schema" and "OWSM MDS Schema". Type the Schema Owner and Schema Password for MDS.
9. In the Test JDBC Data Sources screen, the connections should be tested automatically. The **Status** column displays the results. Ensure that all connections were successful. If not, click **Previous** to return to the previous screen and correct your entries.

Click **Next** when all the connections are successful.

10. In the Select Advanced Configuration screen, do one of the following:
  - If ESS is to be configured on a single server, `ess_server1`, then click **Next**, and go to step 15.
  - Otherwise, select **Manage Servers, Clusters and Machines and Deployments and Services**.

Click **Next**.

11. In the Configure Managed Servers screen, optionally rename `ess_server1` as desired. If you delete `ess_server1` in this screen, Oracle Enterprise Scheduling Service will be deployed to the admin server automatically. You can also create more managed servers by clicking **Add**. Choose unique listen ports between 13000 and 13500 for each new server added.

Click **Next**.

12. In the Configure Clusters screen, if you want to create a cluster, click **Add**, then name your cluster. Leave the other entries as they are.

Click **Next**.

13. In the Assign Servers to Clusters screen, if you have created a cluster, add your servers to the cluster.

Click **Next**.

14. In the Configure Machines screen, add machines if needed.

Click **Next**.

15. In Target Deployments to Clusters or Servers screen, verify the targeting of components. No changes should be necessary.

- The following applications should be targeted only to the ESS server or cluster: **ESSAPP**, **wsm-pm**.
- The following libraries should be targeted only to the ESS server or cluster: **oracle.ess.client\***, **oracle.ess\***.
- **FMW Welcome Page Application** will be targeted only to the admin server.
- All other components should be targeted to both the admin server and ESS server or cluster.

Click **Next**.

16. On the Target Services to Clusters or Servers screen, verify the following targeting of services. No changes should be necessary.
  - JDBC System Resources **EssDS**, **EssXADS**, **EssInternalDS**, **mds-ESS\_MDS\_DS** should all be targeted to the ESS server or cluster.
  - All the other services should be targeted to both the admin server and the ESS server or cluster.

Click **Next**.

17. In the Configuration Summary screen, review and click **Create**.
18. In the Creating Domain screen, wait for the domain creation to complete, then click **Done**.

After these steps, the domain should be configured with both Oracle Enterprise Scheduling Service and the product that includes it.

### 3.3 Configuring Perl to Support Process Jobs

To complete domain setup, you need to configure Perl to support process jobs. Oracle Enterprise Scheduling Service internally uses Perl for supporting process jobs.

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 5.8 or higher.
4. ESS locates perl in the following order:
  1. From the configuration above.
  2. If WL\_HOME is defined, it looks here: \$WL\_HOME.../../webtier\_mwhome/webtier/perl/bin/perl.
  3. Otherwise, it uses it from the container's path.

### 3.4 Configuring Oracle Enterprise Scheduling Service

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

Configuring Oracle Enterprise Scheduling Service involves the following main steps:

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

This section contains the following topics:

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

### 3.4.1 Expanding an Oracle Enterprise Scheduling Service Cluster

An Oracle Enterprise Scheduling Service 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 Scheduling Service cluster from the Oracle WebLogic Server Console.

When scaling out an Oracle Enterprise Scheduling Service 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.

Oracle Enterprise Scheduling Service provides protection for a new server so that work allocations can be configured before the server begins processing jobs. When a newly added server starts for the first time, Oracle Enterprise Scheduling Service determines whether the default work assignment can be used. If it cannot, Oracle Enterprise Scheduling Service binds a pre-seeded internal work assignment (`ESSInternalWA`) which includes a health check job. You can use the health check job to check the server, and unbind the internal work assignment and then bind your own work assignment as required.

---

---

**Note:** When determining whether the default work assignment can be used, Oracle Enterprise Scheduling Service considers all running servers in the group without regard to the applications that are deployed to the servers. Servers that are down are not taken into account.

---

---

For more information about work assignments, see [Section 5.3.1](#).

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

A work assignment must be bound to a request processor before it can process job requests. 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.

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 [Section 5.3.1](#).

To configure a request processor:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduling Service 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 Scheduling Service 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 once 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 Scheduling Service 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 Starting and Stopping Oracle Enterprise Scheduling Service Components

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

The following components can be started and stopped:

- Oracle Enterprise Scheduling Service instance.
- Job request processor and dispatcher.

---



---

**Note:** Stopping an Oracle Enterprise Scheduling Service instance or component is not recommended. Shutting down an Oracle Enterprise Scheduling Service component does not stop job requests from accumulating in the queue.

---



---

This section contains the following topics:

- [Starting and Stopping an Oracle Enterprise Scheduling Service Service Instance](#)

- [Starting and Stopping a Request Processor or Dispatcher](#)

### 3.5.1 Starting and Stopping an Oracle Enterprise Scheduling Service Service Instance

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

---

---

**Note:** Stopping an Oracle Enterprise Scheduling Service instance or component is not recommended.

---

---

To start up an instance of Oracle Enterprise Scheduling Service:

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

To shut down an instance of Oracle Enterprise Scheduling Service:

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

### 3.5.2 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 Scheduling Service instance or component is not recommended.

---

---

To stop a request processor or dispatcher:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduling Service application.
2. Start the request processor or request dispatcher as follows.
  - To start the request processor: From the Scheduling Services menu, select **Request Processor** and then select **Start**.
  - To start the request dispatcher: From the Scheduling Services menu, select **Request Dispatcher** and then select **Start**.
3. Stop the request processor or request dispatcher as follows.
  - To stop the request processor: From the Scheduling Services menu, select **Request Processor** and then select **Stop**.
  - To stop the request dispatcher: From the Scheduling Services menu, select **Request Dispatcher** and then select **Stop**.
4. When prompted as to whether or not to stop the processor or dispatcher, click **OK**.

## 3.6 Managing Application Properties

You can configure Oracle Enterprise Scheduling Service 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 Scheduling Service or properties specified for the application in its deployed configuration.

Oracle Enterprise Scheduling Service 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**: Specifies the binary path of Perl. The default is "/usr/bin/perl".
- **EssCallbackClientSecurityPolicyURI**: Specifies the security policy URI used in the WS-Security headers for web service invocations from Oracle Enterprise Scheduling Service 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.

To edit application properties:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduling Service 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. From the **Type** dropdown, select the new property's type.
  1. Select **ESS** to choose a property defined by Oracle Enterprise Scheduling Service, then select the property's name from the **Name** dropdown. Enter the property's value in the **Value** text field.
  2. Select another type to enter the name and value for another property.
  3. Click **OK** to save the new property setting.
7. On the Application Properties page, click **Apply** to save application properties changes.

## 3.7 Managing Application and Role Policies

To control access to Oracle Enterprise Scheduling Service data and metadata, you can create application roles to represent users and user groups, then grant to 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 [Section 5.4 in Chapter 5, "Managing the Work of Oracle Enterprise Scheduling Service Jobs"](#).

- For more on granting access to data, see [Section 4.4](#) in [Chapter 4](#), "Managing Oracle Enterprise Scheduling Service Requests".

### 3.7.1 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**.

## 3.8 Managing Oracle Enterprise Scheduling Service Web Services

Oracle Enterprise Scheduling Service web services are the same as other application 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*.



---

---

# Managing Oracle Enterprise Scheduling Service Requests

This chapter describes how to manage Oracle Enterprise Scheduling Service job requests, creating and submitting them, as well as holding and cancelling them as needed.

Job requests to 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 cancelling them, as well as doing clean-up by purging requests from the database.

This chapter includes the following topics:

- [Introduction to Managing Oracle Enterprise Scheduling Service Job Requests](#)
- [Managing Oracle Enterprise Scheduling Service Requests](#)
- [Managing Purge Policies](#)

## 4.1 Introduction to Managing Oracle Enterprise Scheduling Service Job Requests

Using Fusion Middleware Control Console, you can manage the job requests that do work on behalf of applications for which Oracle Enterprise Scheduling Service 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 cancelling requests.
- Purging job requests from the database.

## 4.2 Managing Oracle Enterprise Scheduling Service Requests

An application uses Oracle Enterprise Scheduling Service to execute jobs on the application's behalf. When the application wants to execute a job, it sends a job request to Oracle Enterprise Scheduling Service. 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 cancelling requests. You can do these tasks with Fusion Middleware Control Console.

This section contains the following topics:

- [Submitting an Oracle Enterprise Scheduling Service Job Request](#)
- [Searching for Oracle Enterprise Scheduling Service Job Requests](#)
- [Viewing Job Request Details](#)
- [Holding and Resuming Oracle Enterprise Scheduling Service Job Requests](#)
- [Cancelling Oracle Enterprise Scheduling Service Job Requests](#)

## 4.2.1 Submitting an Oracle Enterprise Scheduling Service 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 [Section 5.2.1.2](#).
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 **Description** 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 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. [Table 5–1](#) lists the names and descriptions of system properties.  
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 icon.
    - **Use existing schedule:** Click the search button to search for and select an existing schedule. For more on defining saved schedules, see [Section 4.2.3](#).
    - **Specify schedule:** Create your own schedule on the fly. Follow the instructions in [Section 4.2.1.2](#)
  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 [Section 5.2.1.2](#).

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 will have 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.
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 will appear when you're assigning the schedule to a job request.
  - **Package:** Optionally enter the name of a Java package related to this schedule if it will help you find or group schedules.
  - **Description:** Optionally enter a description.

## 4.2.2 Searching for Oracle Enterprise Scheduling Service Job Requests

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

- 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 that returns the same information as the simple search, as well as the date and time of execution, the run time or wait time of the job request, the number of retries and any error type that may have occurred during execution.
- 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 Scheduling Service run-time database that holds jobs requests -- can be shared across multiple domains with Oracle Enterprise Scheduling Service. 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 need 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. Statuses are listed in the following table.

Statuses		
BLOCKED	EXPIRED	SCHEDULE_ENDED
CANCELLED	FINISHED	SUCCEEDED
CANCELLING	HOLD	UNKNOWN
COMPLETED	PAUSED	VALIDATION_FAILED
ERROR	PENDING_VALIDATION	WAIT
ERROR AUTO RETRY	READY	WARNING
ERROR MANUAL RECOVERY	RUNNING	

- **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**, **Last 31 Days**.
  - **Submitted By:** In the text field, enter the name of the user who submitted the job request you want to find.
  - **Job Definition:** Click the search button next to the text field and select the relevant job definition name.
  - **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 [Section 4.2.2.2](#).
  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 Scheduling Service run-time database that holds jobs requests -- can be shared across multiple domains with Oracle Enterprise Scheduling Service. 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 [Section 4.2.2.1](#).
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 icon 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. 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 Scheduling Service Request search page and display the search interface.
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 don't.

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. Optionally, reorder the columns by selecting the relevant column names and using the vertical arrows on the right to move them up or down.
4. 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. To create a new schedule:
  1. Click **Create**.
  2. Enter the following information:

**Name:** Enter a name for the schedule. This is the name that will appear when you're assigning the schedule to a job request.

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

**Description:** Optionally enter a description.

3. From the **Frequency** dropdown, select how often you want job requests using this schedule to run. Each frequency option will have 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.
4. Click **OK**.
3. To edit an existing schedule:
  1. In the **Name** and **Package** fields, enter values to search for.
  2. Click **Go** to search.
  3. In the **Results** area, locate the schedule you want to edit.
  4. Select the schedule you want to edit, then click **Edit**.
  5. On the Edit Schedule page, edit details about the schedule.
  6. Click **OK**.
4. Click **Search** to submit the job request search.

#### 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 [Section 4.2.2](#).

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 [Section 4.2.2.1](#)
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.
- **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 will be 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 Scheduling Service 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. For more information, see [Section 4.2.2.1](#).
2. In the table of job request search results, select the job request you want to hold.
3. Click the **Hold** button.
4. To resume the paused job request, select the job request and click **Resume**.

## 4.2.6 Cancelling Oracle Enterprise Scheduling Service 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
BLOCKED	CANCELLED
PENDING_VALIDATION	CANCELLED
ERROR_AUTO_RETRY	CANCELLED
PAUSED	CANCELLING
RUNNING	CANCELLING
COMPLETED	CANCELLING
Any terminal state	No state change

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

<b>Job State When Initiating Cancellation</b>	<b>New Cancellation State</b>
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 cancelled 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](#)
- [Cancelling Parent Job Requests](#)
- [Cancelling 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**

<b>State Returned by the Pre-Processor</b>	<b>State Following Cancellation Initiation</b>
PROCEED	CANCELLED
DELAY	CANCELLED
WARN	CANCELLED
SYSTEM_ERROR	ERROR
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**

<b>State Returned by the Job</b>	<b>State Following Cancellation Initiation</b>
Normal return (success)	CANCELLED
ExecutionWarningException	CANCELLED
ExecutionPausedException	CANCELLED
ExecutionBizErrorException	ERROR

**Table 4–3 (Cont.) Returned Synchronous Java Job States and States Following Cancellation Initiation**

State Returned by the Job	State Following Cancellation Initiation
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 Scheduling Service 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 Scheduling Service 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 Cancelling Parent Job Requests

A parent job request that is not executable must be in `WAIT` state to be canceled. It then proceeds to `CANCELLING`. 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 Cancelling a Child Job Request

A child job request may be an executable or a parent job request. When cancelling 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 cancelling 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 scheduled jobs according to specified criteria. For example, a purge policy might specify the retention of all Java type job requests using a particular job definition submitted executed by a given application for three days. Another purge policy might retain a particular type of job request, say, all SQL job requests in a successful state, for only one day. You can also specify the frequency at which the purge policy is to run.

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 [Section 4.3.2](#).

---

To set up a new purge policy:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduling Service 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 icon 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 Scheduling Service 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 Scheduling Service 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 Scheduling Service schema is created or updated. The stored procedure is shown in [Example 4-1](#).

#### **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 );

```

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

#### **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);

```

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);
```

### 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 Scheduling Service 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.

To update a purge policy:

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

To delete a purge policy:

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

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 may see the request output.
- If a user can see a request, and if the request run as elevated user, the user is not able to 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 Scheduling Service. 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.



**Table 4–7 (Cont.) Grant Actions for Data Security**

<b>Action</b>	<b>Effect</b>
ESS_REQUEST_DELETE	Delete a request.
ESS_REQUEST_PURGE	Purge 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 Scheduling Service Components

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

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.RuntimeDataPermission**.
11. Click the **Search** button.
12. Under Search Results, select **Scope=APPLICATION**, then click **Continue**.
13. In the Add Permission dialog, in the Permission Actions field, edit the comma-separated list of permissions so that the list reflects the permissions you want to grant.
14. Click **OK**.



---

---

# Managing the Work of Oracle Enterprise Scheduling Service Jobs

This chapter describes how you can add metadata to refine Oracle Enterprise Scheduling Service 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 topics:

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

## 5.1 Introduction to Managing the Work of Oracle Enterprise Scheduling Service 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 Scheduling Service 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're 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 at the same time.
- 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 Scheduling Service 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.

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, duplicate, delete and search for job definitions.

This section contains the following topics:

- [Viewing Job Definitions](#)
- [Creating 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 Scheduling Service 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.
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.
Package	This column displays the name of the Java package associated with the job definition.
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 a Job Definition

You can create a new job definition, which you can then use to 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.

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 run time, such as boolean, number and string values.

This section contains the following topics:

- [Configuring Parameters](#)
- [Configuring System Properties](#)
- [Configuring Access Control](#)

To create a job definition:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduling Service 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.
5. For the new job definition, configure the following properties:

- **Name** (required): Enter a name for the job definition.
- **Package**: Enter the package of the job to be run.
- **Description**: Enter a description for the job.
- **Job Type** (required): From the dropdown list, select the job type you want to use for the job definition. Select the **Overwrite** check box to

The Job Type you choose will determine 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)

The class name that is associated with the job definition's job type. If you can edit the field, select **OverWrite** 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 **OverWrite** 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.

- **Command Line (for ProcessJobType)**

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

6. Optionally, configure parameters, system properties, and access control. For more information, see [Section 5.2.1.2.1](#), [Section 5.2.1.2.2](#), and [Section 5.2.1.2.3](#).
7. Click **OK**.

**5.2.1.2.1 Configuring Parameters** 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 **Parameters** section.
2. Click the **Add** icon.

The Add Parameter dialog box displays.

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

To configure system properties for a job definition:

1. On the Create Job Definition page, expand the **System Properties** section.
2. Click the **Add** icon.

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](#).

**Table 5–1 System Properties**

<b>System Property</b>	<b>Description</b>
<code>SYS_EXT_cmdLine.Unix</code>	For a process job type, this specifies a string representing the command line command on Unix.  If the <code>SYS_cmdLine</code> property is defined, this property is ignored.
<code>SYS_EXT_cmdLine.Windows</code>	For a process job type, this specifies a string representing the command line command on Windows.  If the <code>SYS_cmdLine</code> property is defined, this property is ignored.
<code>SYS_EXT_executableDir.Unix</code>	For a process job, this specifies the directory where the process executable is located on Unix.
<code>SYS_EXT_executableDir.Windows</code>	For a process job, this specifies the directory where the process executable is located on Windows.
<code>SYS_EXT_executableName</code>	For a process job, this specifies the name of the process executable.
<code>SYS_EXT_executableSuffix.Unix</code>	Specifies the suffix for <code>SYS_EXT_executableName</code> on Unix. The default is ".sh".
<code>SYS_EXT_executableSuffix.Windows</code>	Specifies the suffix for <code>SYS_EXT_executableName</code> on Windows. The default is ".cmd".
<code>SYS_EXT_executableAutoExport</code>	For process and Java job types, this specifies a value that prompts Oracle Enterprise Scheduling Service 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 needs 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_processArguments</code>	For a process job, this specifies the arguments.
<code>SYS_EXT_requestLogLevel</code>	Specifies the log level for Java and PL/SQL request loggers. The default is "INFO".
<code>SYS_EXT_resolvedCmdLine</code>	For a process job type, after the job is successfully started, this property provides a place for Oracle Enterprise Scheduling Service to save the actual command line (with substitution performed). This property is for diagnostic purposes only. It is for use by Oracle Enterprise Scheduling Service and other values might be overwritten by the service.
<code>SYS_EXT_supportOutputFiles</code>	Specifies a string indicating whether the job will create files in the file system. The following values are supported: <ul style="list-style-type: none"> <li>■ "work" to create a temporary file rather than an output file.</li> <li>■ "output" to use the file system to create output files.</li> <li>■ "none" (or no value) to not create any files in the file system.</li> </ul>
<code>SYS_EXT_uploadContentToRepository</code>	Set this to "copy" to have Oracle Enterprise Scheduling Service upload request log and output files from its content store to a separate repository (UCM). The request log and output content will remain in the content store, but will also be available to user interfaces and tools that expect log/output in UCM.
<code>SYS_EXT_useAlternateEnvironment</code>	Set this to "true" to get environment properties from Fusion Applications-style files. The default is "false".
<code>SYS_EXT_useExtendedSetup</code>	Set this to "true" to have Oracle Enterprise Scheduling Service set up extended functionality, such as <code>AppSession</code> , for each stage of the request if that functionality is available.

**Table 5–1 (Cont.) System Properties**

<b>System Property</b>	<b>Description</b>
<code>SYS_EXT_userFileDirShared</code>	<p>For process and Java job types that may create files in the file system, this will prompt Oracle Enterprise Scheduling Service to save in the request the configured <code>RequestFileDirectory</code> and <code>RequestFileDirectoryShared</code> from the <code>ess-config.xml</code> file. These values are saved as the properties <code>SYS_userFileDir</code> and <code>SYS_EXT_userFileDirShared</code>, respectively. Oracle Enterprise Scheduling Service uses these properties to create the request work and output directories, depending on <code>SYS_EXT_supportOutputFiles</code>.</p> <p>This for use by Oracle Enterprise Scheduling Service and other values might be overwritten by the service.</p>
<code>SYS_allowMultPending</code>	<p>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.</p>
<code>SYS_application</code>	<p>Specifies the logical name of the J2EE application used for request processing. This property is automatically set by Oracle Enterprise Scheduler during request submission.</p>
<code>SYS_bizErrorExitCode</code>	<p>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.</p> <p>This property is optional for a process job type. It is not used for other job types.</p> <p>For more information about business errors, see <a href="#">Section 4.2.2.2</a></p>
<code>SYS_className</code>	<p>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.</p>
<code>SYS_cmdLine</code>	<p>Specifies the command line used to invoke an external program for a process job request.</p> <p>This property is required for a Process job type. It is not used for other job types.</p>
<code>SYS_effectiveApplication</code>	<p>Specifies specifies the logical name of the J2EE application that will be 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.</p>
<code>SYS_environmentVariables</code>	<p>Specifies 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 (<code>name=value</code>) representing the environment variables to be set.</p> <p>This property is optional for a Process job type. It is not used for other job types.</p>
<code>SYS_executePast</code>	<p>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 nor 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.</p> <p>Values for this property are:</p> <ul style="list-style-type: none"> <li>■ <code>TRUE</code>: All instances specified by a schedule are generated regardless of the time of generation.</li> <li>■ <code>FALSE</code>: Instances with a scheduled execution time in the past (that is, before the time of generation) will not be generated.</li> </ul> <p>If this property is not specified, the system defaults to <code>TRUE</code>.</p>
<code>SYS_extensionListener</code>	<p>For internal use only.</p>



**Table 5–1 (Cont.) System Properties**

<b>System Property</b>	<b>Description</b>
<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 Scheduling Service. 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 Scheduling Service.
<code>SYS_groupName</code>	Specifies the name of the Oracle Enterprise Scheduling Service isolation group to which this request is bound. This property is automatically set by Oracle Enterprise Scheduling Service 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 Scheduling Service 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 Scheduling Service 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_logWorkDirectory</code>	Specifies the logging working directory. This is for use by Oracle Enterprise Scheduling Service 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.
<code>SYS_outputWorkDirectory</code>	Specifies the working directory used during job request processing for output files. Oracle Enterprise Scheduling Service sets the value of this property during job request processing. This is for use by Oracle Enterprise Scheduling Service 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 <code>schema.name</code> 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.

**Table 5–1 (Cont.) System Properties**

<b>System Property</b>	<b>Description</b>
<code>SYS_redirectedOutputFile</code>	<p>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.</p> <p>This is for use by Oracle Enterprise Scheduling Service and other values might be overwritten by the service.</p>
<code>SYS_reprocessDelay</code>	<p>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.</p> <p>Default: If this property is not specified, the system default used is 5. Integer type.</p>
<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 needs.
<code>SYS_requestedProcessor</code>	<p>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.</p> <p>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 will never be executed. If the specified node is not running, the request will remain in <code>READY</code> state and will not be executed until the node is restarted.</p>
<code>SYS_requestExpiration</code>	<p>Specifies the expiration time for a request. This represents the time, in minutes, that a request will expire after its scheduled execution time. A expiration value of zero (0) means that the request never expires. If this property is not specified, the system default value used is 0.</p> <p>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.</p>
<code>SYS_retries</code>	<p>Specifies the retry limit for a failed request. If request execution fails, the request will be retried up to the number of times specified by this property until the request succeeds. If retry limit is zero (0), a failed request will not be retried.</p> <p>If this property is not specified, the system default used is 0.</p>
<code>SYS_runasApplicationID</code>	<p>This property enables elevating access privileges for completing a scheduled job.</p> <p>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.</p>
<code>SYS_selectState</code>	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 will be included in the determination of the state of the job set. Boolean type.

**Table 5–1 (Cont.) System Properties**

<b>System Property</b>	<b>Description</b>
<code>SYS_sqlJobClass</code>	<p>Specifies an Oracle Enterprise Scheduling Service 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.</p> <p>If this property is not specified, a default Oracle Enterprise Scheduling Service 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 Scheduling Service SQL job request to use that default job class.</p> <p>This property is optional for a SQL job type. It is not used for other job types.</p>
<code>SYS_submittingApplication</code>	<p>Specifies the logical name of the J2EE application for the submitted (absolute parent) request. This property is automatically set by Oracle Enterprise Scheduling Service during request submission.</p>
<code>SYS_substitutionHandlers</code>	<p>This property has been deprecated.</p>
<code>SYS_successExitCode</code>	<p>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.</p> <p>This property is optional for a process job type. It is not used for other job types.</p>
<code>SYS_userFileDir</code>	<p>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.</p> <p>Oracle Enterprise Scheduling Service supports a configuration parameter that specifies a file directory where requests may store files. At request submission, a <code>USER_FILE_DIR</code> property is automatically added for the request if the configuration parameter is currently set and <code>USER_FILE_DIR</code> property was not specified for the request. If the property is added, it will be initialized to the value of the configuration parameter. The property will not be added if the configuration parameter is not set at the time of request submission.</p> <p>This is for use by Oracle Enterprise Scheduling Service and other values might be overwritten by the service.</p>
<code>SYS_userName</code>	<p>Specifies the name of the user used to execute the request. Normally, this is the submitting user unless the <code>RUNAS_APPLICATIONID</code> property was set in the job metadata. This property is automatically set by Oracle Enterprise Scheduling Service during request submission.</p>
<code>SYS_warningExitCode</code>	<p>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.</p> <p>This property is optional for a process job type. It is not used for other job types.</p>
<code>SYS_workDirectoryRoot</code>	<p>Specifies the working directory for the process of a process job request.</p> <p>This is for use by Oracle Enterprise Scheduling Service and other values might be overwritten by the service.</p>

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**.

**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 you visible here are defined in a jazn.xml file used by the application.

1. On the Create Job Definition page, expand the **Access Control** section.
2. Click the **Add** icon.  
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 will 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**.

## 5.2.2 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.

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

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

Name	Package	Execution Mode	Description
ShortAndLongJob	/mypackage	Serial	

Total Rows : 1

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 Scheduling Service 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.
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.

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

### 5.2.2.2 Creating or Editing a Job Set

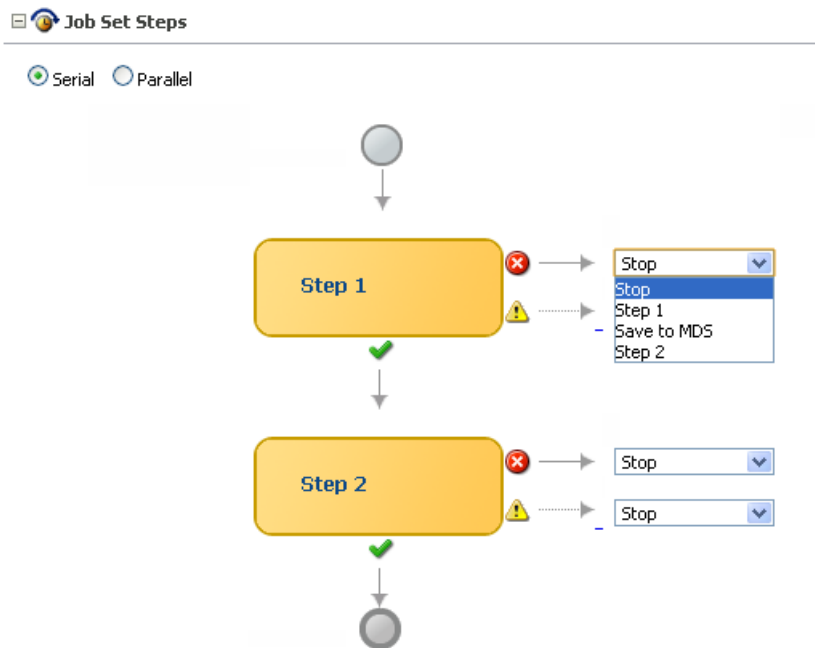
You define a job set as either serial or parallel. At run time, Oracle Enterprise Scheduling Service runs parallel job set steps together, in parallel. When a serial job set runs, on the other hand, Oracle Enterprise Scheduling Service runs the steps one after another in a specified sequence. With a serial job set, Oracle Enterprise Scheduling Service 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:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduling Service application.
2. From the Scheduling Services menu, select **Job Metadata** and then select **Job Sets**. The Job Sets page displays.
3. Click **Create** to define a new job set, or **Edit** to modify an existing job set.
4. In the Name field, enter the name of the job set. Optionally, add a description in the **Description** text field, and the name of the relevant job set Java package in the **Package** text field.

5. In the Job Set Steps section, select **Serial** or **Parallel** to create a serial or parallel job set.
6. Add steps as required by clicking the **Add** icon. Define each step as required.
  - a. In the Step tab, in the **Step ID** field, enter a meaningful ID for the step.  
 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. In the Effective Application region, 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 Parameters tab, click the **Add** icon to define any required parameters and enter their initial value in the field provided. For more information about defining parameters, see [Section 5.2.1.2.1](#).
  - d. In the System Properties tab, click the **Add** icon to select system properties and enter their initial value in the field provided. For more information about selecting system properties, see [Section 5.2.1.2.2](#).
  - 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.



7. Continue to define steps as required for the job set.
8. If your job set requires them, define parameters and system properties in sections toward the bottom of the job set window.
9. Configure access control for the job set. For more on defining access control, see [Section 5.2.1.2.3](#).

10. 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 [Section 5.2.2.1](#).
2. In the Results table, select the job set you want to delete and click the **Delete** icon.

## 5.2.3 Managing Incompatibilities

An Oracle Enterprise Scheduling Service 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 Scheduling Service application.
2. From the Scheduling Services menu, select **Job Metadata** and then select **Incompatibilities**.

The Incompatibilities page displays.

3. 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.
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.

4. 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 two or more job definitions configured as incompatible, and optionally the resource over which they need to 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 Scheduling Service 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 Scheduling Service 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.

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 Scheduling Service application.
2. From the Scheduling Services menu, select **Job Metadata** and then select **Incompatibilities**.

The Incompatibilities page displays.

3. Click **Create** to define a new incompatibility, or **Edit** to modify an existing incompatibility.

The Create Incompatibilities page displays.

4. Enter the following information:
  - **Name:** Enter a name for the incompatibility.
  - **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**.
- 5. In the Entities section, click the Add icon to add jobs to the incompatibility. The Add Entity dialog box displays.
- 6. 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 icon, 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 [Section 5.2.1.2.3](#).
- 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 [Section 5.2.3.1](#).
2. In the Results table, select the incompatibility you want to delete and click the **Delete** icon.

## 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 will be started to process all jobs using all available resources.

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 run time, work assignments can be attached to a request processor and can limit the period during which job requests of a certain type can be processed.
- At run time, a request processor can be controlled and a work assignment can limit the resources that are available to process all job requests.

#### 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 will be active using all threads and having no limits on asynchronous workers.

Only one workshift can be active for a given work assignment at any point in time. If a work assignment has more than one active workshift, Oracle Enterprise Scheduling Service chooses the most specific of these to be the actual active workshift. The most specific workshift is the one that ends soonest, or, given two workshifts that have ended at the same time, the workshift that started most recently.

#### 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 needed to satisfy the allocations of all active work assignments exceeds the configured thread count. In this case, Oracle Enterprise Scheduling Service 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 Scheduling Service allocates 14 threads to work assignment 1 and six 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.

---

---

#### Processing Active Work Assignments

After determining work assignments and thread allocations, Oracle Enterprise Scheduling Service 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.

This section contains the following topics:

- [Creating or Editing a Work Assignment](#)
- [Deleting a Work Assignment](#)

### 5.3.1.1 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 5–2](#) shows the properties that you can define for specialization rules.

**Table 5–2 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 a label defined by the system administrator allowing administrators to group job requests for specific needs.

The following operators can be used to join the conditions of a rule: AND, OR (both of type binary), CONTAINS and NOT (unary).

[Example 5–1](#) shows sample specialization rules that can be used in a 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')
```

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.

To create or edit a work assignment:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduling Service 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 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** icon to add one or more workshifts to the work assignment.
8. Click **OK** to save the work assignment.

### 5.3.1.2 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 Scheduling Service 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 run time, Oracle Enterprise Scheduling Service determines the resource allocation for the workshifts within the work assignment.

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 would share the ten PL/SQL asynchronous workers, for a maximum of ten PL/SQL jobs active for that work assignment.

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.

To create or edit a workshift:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduling Service 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 Workshift page displays.

4. Enter the following information for the workshift.
  - **Name:** Enter a name for the workshift.
  - **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 removes 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.  
From the Time Zone dropdown list, select a time zone for the schedule.  
In the Start region, select **Immediately** or **Later**. If you choose Later, select the date and time using the calendar icon.  
In the Repeating region, use the Repeat dropdown list to select the frequency at which the workshift schedule will repeat: **Do not repeat, Every N minutes/hours/days/weeks, Specific days of the week, Specific days of the month**. Enter the number of minutes, hours, days or weeks. Alternatively, select the days of the week or dates of the month on which the schedule is to run.  
In the End Date region, select **No End Date** or **Specified End Date**. When selecting Specified End Date, use the calendar icon to set an end date in the **Date** text field.  
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.

To delete a workshift:

1. From the navigation pane, expand the **Scheduling Services** folder and select the Oracle Enterprise Scheduling Service 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, purge policies and job requests.

This section contains the following topics:

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

### 5.3.3.1 Creating or Editing a Schedule

A job schedule specifies a predefined time or recurrence for scheduling and executing a job request, work allocation or purge policy. Schedules are defined independent of jobs and can be associated with one or more jobs.

To create or edit a schedule:

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

The Schedules page displays.

3. Click **Create** to define a new schedule, or **Edit** to modify an existing 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.
  - **Frequency:** From the dropdown, select how often you want items using this schedule to run. Each frequency option will have its own way to specify details for the option. As you specify details, note that you can click calendar icons to specify time zones as well.
5. Click **OK** to save the schedule.

### 5.3.3.2 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 Scheduling Service application.
2. From the Scheduling Services menu, select **Work Allocation** and then select **Schedules**.

The Schedules page displays.

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

## 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 5-3](#) lists possible actions.

**Table 5-3 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 entities' containing package. That list includes items you or application developers might have defined as well as items Oracle Enterprise Scheduling Service has defined. [Table 5-4](#) lists a few examples of the effect of granting permission for certain actions to certain resources.

**Table 5-4 Sample Permission Grants for Security**

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

## 5.4.2 How to Create Metadata Policies for Oracle Enterprise Scheduling Service 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 5-4](#) 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 5-3](#) lists possible actions.
15. Click **OK**.



---

# Monitoring Oracle Enterprise Scheduling Service

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

This chapter includes the following topics:

- [Overview of Monitoring Oracle Enterprise Scheduling Service](#)
- [Monitoring Oracle Enterprise Scheduling Service Request Activity](#)
- [Monitoring the Performance of a Single Oracle Enterprise Scheduling Service Instance](#)
- [Monitoring Key Performance Metrics for Oracle Enterprise Scheduling Service](#)
- [Viewing General Oracle Enterprise Scheduling Service Instance Information](#)
- [Viewing Scheduler Component Status](#)
- [Managing Logging for Oracle Enterprise Scheduling Service](#)

## 6.1 Overview of Monitoring Oracle Enterprise Scheduling Service

Much of the monitoring you'll do is related to job requests, where Oracle Enterprise Scheduling Service 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 Scheduling Service 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 Scheduling Service.

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

## 6.2 Monitoring Oracle Enterprise Scheduling Service 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 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 Scheduling Service 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 Scheduling Service 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 Scheduling Service Requests](#)
- [Viewing Top Ten Oracle Enterprise Scheduling Service Jobs Ready to Be Executed](#)

### 6.2.1.1 Viewing Top Ten Long Running Oracle Enterprise Scheduling Service 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 Scheduling Service 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 Scheduling Service 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 will all 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 Scheduling Service application.
2. At the top of the Oracle Enterprise Scheduling Service home page, click the **Top 10 Long Running 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 Scheduling Service 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 Scheduling Service. 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 will all 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 Scheduling Service application.
2. At the top of the Oracle Enterprise Scheduling Service page, click the **Top 10 Ready 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 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 Scheduling Service Instance

You can monitor the performance of an Oracle Enterprise Scheduling Service 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 Scheduling Service Activity](#)
- [Viewing Current Oracle Enterprise Scheduling Service Activity](#)
- [Viewing Historical Oracle Enterprise Scheduling Service Reports](#)
- [Managing Chart Sets](#)
- [Selecting the Performance Time Period](#)

### 6.3.1 Viewing a Summary of Oracle Enterprise Scheduling Service Activity

The Oracle Enterprise Scheduling Service 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 Scheduling Service 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 Scheduling Service.

2. You can change the duration for which performance data display.
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 **Go** 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 [Section 6.3.4.1, "Saving a Chart Set"](#).
  - Reorder the chart set, as described in [Section 6.3.4.3, "Reordering Charts"](#).
  - Revert to an old chart set, as described in [Section 6.3.4.4, "Reverting to an Old Chart Set"](#).
  - Delete a chart set, as described in [Section 6.3.4.2, "Deleting a Chart Set"](#).

## 6.3.2 Viewing Current Oracle Enterprise Scheduling Service Activity

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

To view current Oracle Enterprise Scheduling Service 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 [Section 5.3.1.1, "Creating or Editing a Work Assignment."](#)

### 6.3.3 Viewing Historical Oracle Enterprise Scheduling Service Reports

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

To view historical Oracle Enterprise Scheduling Service 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: <b>Seconds, Minutes, Hours</b> or <b>Days</b> .
Completed between	Use the calendar icons 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: <b>Seconds, Minutes, Hours</b> or <b>Days</b> .
Completed between	Use the calendar icons 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 icon 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 icons 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 Scheduling Service 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 Scheduling Service 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 Scheduling Service Performance Summary page.
2. Choose **View > Save Chart Set**.
3. In the Save Chart Set dialog box, enter a name for the chart set and click **OK**.
4. Once 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 Scheduling Service Performance Summary page.

To delete a chart set:

1. Display the Oracle Enterprise Scheduling Service Performance Summary page.
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 Scheduling Service Performance Summary page. Charts can be moved up or down, or removed entirely.

To reorder chart sets:

1. Display the Oracle Enterprise Scheduling Service Performance Summary page.
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 **Delete**.

4. Click **OK**.

#### 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 Scheduling Service Performance Summary page.
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 Scheduling Service 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 Scheduling Service Performance Summary page.
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 icon to select a time period for which you want to display job requests.

## 6.4 Monitoring Key Performance Metrics for Oracle Enterprise Scheduling Service

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

**Table 6–7 Oracle Enterprise Scheduling Service Key Performance Metrics**

<b>Metric Name</b>	<b>Location in Fusion Middleware Control</b>
Number of active processor threads for each instance	<ol style="list-style-type: none"> <li>1. From the navigation pane, expand <b>Scheduling Services</b>.</li> <li>2. Select the Oracle Enterprise Scheduling Service component.</li> <li>3. From the <b>Scheduling Service</b> menu, choose <b>Performance &gt; Current Activity</b>.</li> <li>4. From the <b>View</b> list, choose <b>Workload by Work Assignment</b>.</li> <li>5. In the <b>Work Assignments with active Workshifts</b> report, view the <b>Thread Allocation</b> columns.</li> </ol>
Processing Time (for jobs) Wait Time	<ol style="list-style-type: none"> <li>1. From the navigation pane, expand <b>Scheduling Services</b>.</li> <li>2. Select the Oracle Enterprise Scheduling Service component.</li> <li>3. From the <b>Scheduling Service</b> menu, choose <b>Performance &gt; Historical Reports</b>.</li> <li>4. In the Historical Reports page, from the <b>View</b> list, choose <b>Request Metrics By Work Assignment</b>.</li> <li>5. In the <b>Results</b> section, view the <b>Wait Time</b> and <b>Processing Time</b> columns.</li> </ol>
Job running time	<ol style="list-style-type: none"> <li>1. From the navigation pane, expand <b>Scheduling Services</b>.</li> <li>2. Select the Oracle Enterprise Scheduling Service component.</li> <li>3. From the Scheduling Service home page, in the <b>Top 10 Long Running Request</b> tab, select a job.</li> <li>4. In the Request Detail page, under <b>Execution Trail</b>, view the <b>Run Time</b> metric.</li> </ol>

## 6.5 Viewing General Oracle Enterprise Scheduling Service Instance Information

You can view information about a particular Oracle Enterprise Scheduling Service 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 Scheduling Service:

1. From the Scheduling Service menu, select **General Information**.
2. Fusion Middleware Control displays the following information.
  - **Target Name:** The location of the target application, in this case, Oracle Enterprise Scheduling Service, 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 Scheduling Service. 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's 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 Scheduling Service

You can search for and view log data for individual job requests, as well as set log levels for Oracle Enterprise Scheduling Service. 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 Scheduling Service Log Levels](#)
- [Tracing Oracle Enterprise Scheduling Service Jobs](#)
- [Saving Job Request Logs](#)

---

---

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

---

---

### 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 Scheduling Service 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.

To view job request logs:

1. Search for the relevant job requests as described in [Section 4.2.2.1](#)
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 Scheduling Service 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 view the logs for a request, Fusion Middleware Control displays only messages logged in the scope of the Oracle Enterprise Scheduling Service cluster. If the Oracle Enterprise Scheduling Service application is not deployed to a cluster, Fusion Middleware Control displays the messages logged in the managed server scope. However, Oracle Enterprise Scheduling Service 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 **Broaden Target Scope** and select the target whose log messages you want to view.

### 6.7.4 Setting Oracle Enterprise Scheduling Service Log Levels

It is possible to configure Oracle Enterprise Scheduling Service server logging for a 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 Scheduling Service. Oracle Enterprise Scheduling Service 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 Scheduling Service 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 Scheduling Service logger names, log levels and a description for each level.

**Table 6–8** *Loggers and log levels for Oracle Enterprise Scheduling Service*

Logger Name	Log Level	Description
oracle.as.ess	SEVERE	Problems encountered by Oracle Enterprise Scheduling Service 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 Scheduling Service code.  Problems encountered by Oracle Enterprise Scheduling Service runtime outside the context of request processing, such as dispatching, system event handling, and so on.
	WARNING	Less severe problems encountered by Oracle Enterprise Scheduling Service 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 Scheduling Service resource adapter.
	CONFIG	Application endpoint activation and de-activation for Oracle Enterprise Scheduling Service resource adapter.

- Request Logging: Oracle Enterprise Scheduling Service 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 [Section 6.7.2](#).

For information about setting the log levels for Oracle WebLogic Server, see [Section 6.7.5](#).

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

## 6.7.5 Tracing Oracle Enterprise Scheduling Service Jobs

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

This section contains the following topics:

- [Navigate to the Specify Loggers Region](#)
- [Define the Loggers](#)
- [Attach the Loggers](#)
- [Download the Trace File](#)

### 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 Scheduling Service jobs:

### Task 1 Navigate to the Specify Loggers Region

Navigate to the Log Configuration pane and expand the Specify Loggers region.

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.

### Task 2 Define the Loggers

Create the following loggers:

```
oracle.as.scheduler.security.internal.SecurityExecutorFactory  
and 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.

### Task 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**.

### Task 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 Scheduling Service 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 Scheduling Service server name>-diagnostic.log`.



---

---

# High Availability for Oracle Enterprise Scheduling Service

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

This appendix includes the following topics:

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

## 6.8 Introduction to High Availability for Oracle Enterprise Scheduling Service

A highly available cluster of Oracle Enterprise Scheduling Service 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 Scheduling Service expects the job to send a status upon completion using a web services callback. If Oracle Enterprise Scheduling Service 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 Scheduling Service 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 Scheduling Service 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 Scheduling Service cluster, see [Chapter B](#).

---

---

## 6.9 Oracle Enterprise Scheduling Service Concepts

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

This section includes the following topics:

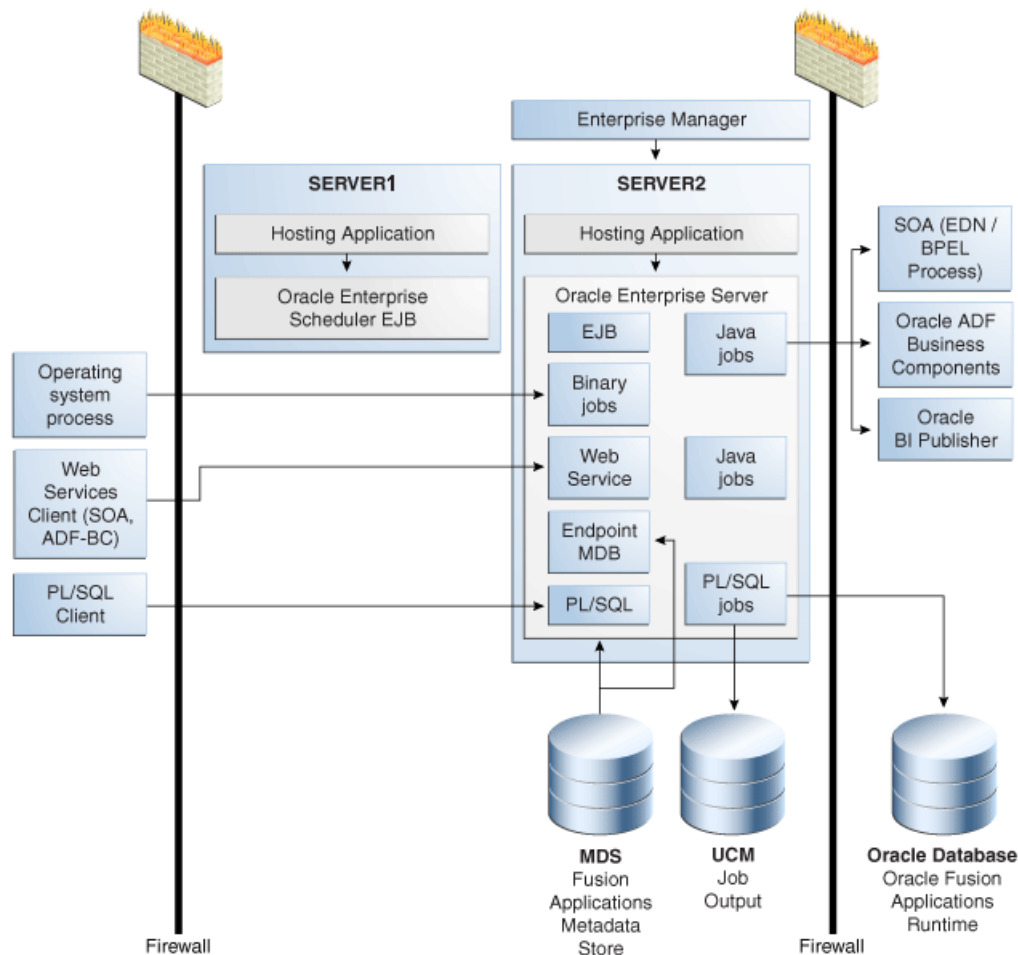
- [Oracle Enterprise Scheduling Service Architecture](#)
- [Oracle Enterprise Scheduling Service Components](#)
- [Oracle Enterprise Scheduling Service Life Cycle](#)
- [Oracle Enterprise Scheduling Service Life Cycle Tools](#)

### 6.9.1 Oracle Enterprise Scheduling Service Architecture

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

[Figure 6–1](#) shows the Oracle Enterprise Scheduling Service run time architecture in the context of Oracle Fusion Middleware components.

Figure 6–1 Oracle Enterprise Scheduling Service Run Time Architecture



The components of the Oracle Enterprise Scheduling Service run time architecture are as follows:

- **Oracle Enterprise Scheduling Service** 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 Scheduling Service:** Fusion Middleware Control enables you to manage Oracle Enterprise Scheduling Service clusters, services and jobs. Oracle Enterprise Scheduling Service accesses metadata using MDS. Scheduled job output is saved to Oracle WebCenter Content. Oracle Enterprise Scheduling Service includes interfaces and APIs that enable interaction with applications and external components. For example, a PL/SQL client uses the Oracle Enterprise Scheduling Service PL/SQL API to request a scheduled job.
- Components accessed by **Oracle Enterprise Scheduling Service:** Oracle Enterprise Scheduling Service 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 Scheduling Service over RMI, whereas client applications using Oracle Enterprise Scheduling Service web

services use HTTP. Connections from client applications to the server are persistent, short-lived asynchronous interactions that sometimes use callback functions.

## 6.9.2 Oracle Enterprise Scheduling Service Components

Oracle Enterprise Scheduling Service components are as follows:

- Oracle ADF Server (Oracle Enterprise Scheduling Service client): `RuntimeServiceEJB` and `MetadataServiceEJB` are deployed as shared libraries. These libraries are imported in the ADF client applications (ears).
- Oracle Enterprise Scheduling Service Server (Oracle Enterprise Scheduling Service Runtime): The Oracle Enterprise Scheduling Service service component manages all scheduled jobs.
- Core runtime: This is an Oracle Enterprise Scheduling Service 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 Scheduling Service hosting application submits job requests using the Oracle Enterprise Scheduling Service libraries or an integrated job request submission interface.
- Oracle Database Scheduler: The standard Oracle Database Scheduler is used to execute Oracle Enterprise Scheduling Service PL/SQL jobs.
- Oracle Enterprise Scheduling Service uses Java process APIs to spawn native binary jobs.

Oracle Enterprise Scheduling Service relies on the following data sources:

- Oracle Enterprise Scheduling Service runtime (XA)
- Oracle Enterprise Scheduling Service runtime (non-XA)
- Oracle Enterprise Scheduling Service 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.

## 6.9.3 Oracle Enterprise Scheduling Service Life Cycle

The Oracle Enterprise Scheduling Service engine starts up as part of the standard J2EE application initialization by Oracle WebLogic Server. The Oracle Enterprise Scheduling Service 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 Scheduling Service.

1. A client application submits a job request.
2. The client application sends the job request to Oracle Enterprise Scheduling Service.

3. Oracle Enterprise Scheduling Service reads the metadata for the job request.
4. Oracle Enterprise Scheduling Service places the job request and the job metadata in a queue in the Oracle Enterprise Scheduling Service run-time data store.
5. Based on schedule and request processor availability, Oracle Enterprise Scheduling Service sends a message to the hosted application, which includes all the job request parameters and metadata captured at the time of submission.
6. The hosting application executes the job and returns a status. Job output and logs are written to Oracle WebCenter Content.
7. Oracle Enterprise Scheduling Service updates the history with the job request status.

Figure 6–2 shows the changes in job state during the life cycle of an executed job request.

**Figure 6–2 Job Request Changes in State During Run Time**

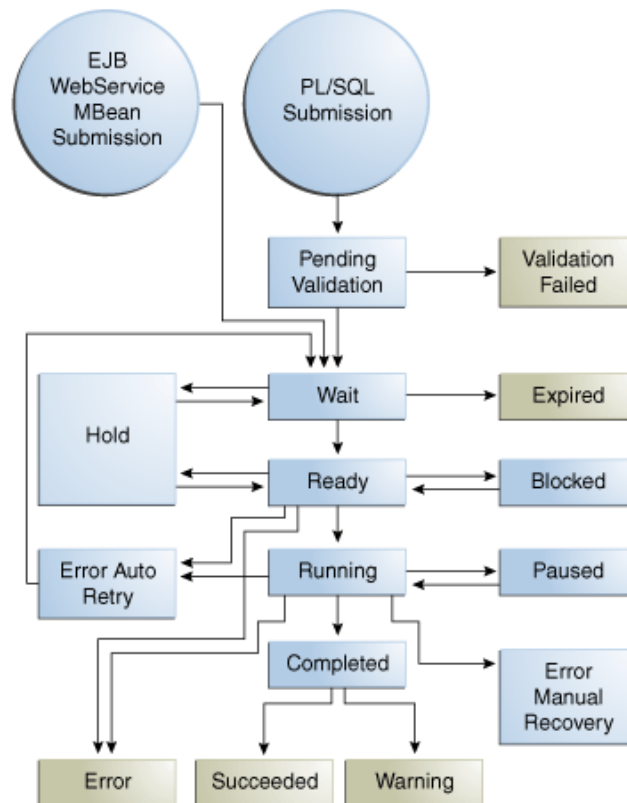


Figure 6–3 displays the changes in state for an executable job request for which the executing user has canceled the request.

**Figure 6–3 Changes in Job State Following Cancellation**

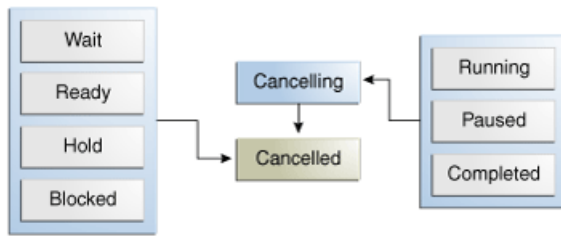
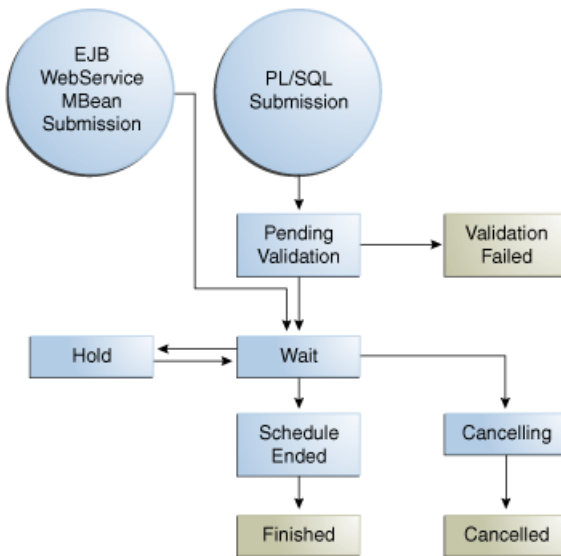


Figure 6–4 displays state transitions for a job request submitted with a schedule.

**Figure 6–4 State Transitions for a Job Request Submitted with a Schedule**



### 6.9.4 Oracle Enterprise Scheduling Service Life Cycle Tools

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

Oracle Enterprise Scheduling Service jobs can be hosted on the same Oracle WebLogic Server instance (or a remote Oracle WebLogic Server instance), database and binary processes. Oracle Enterprise Scheduling Service controls the life cycle of Oracle Enterprise Scheduling Service jobs. Use Oracle Enterprise Manager to monitor and manage Oracle Enterprise Scheduling Service jobs. For more information about managing Oracle Enterprise Scheduling Service jobs, see [Chapter 4](#).

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*.

## 6.10 Configuring High Availability for Oracle Enterprise Scheduling Service

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

This section includes the following topics:

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

### 6.10.1 Oracle Enterprise Scheduling Service Configuration and Deployment Artifacts

Configuration files are as follows:

- `ess.xml`: This file is part of the Oracle Enterprise Scheduling Service EAR file deployed to the Oracle Enterprise Scheduling Service cluster.
- MDS repository: The Oracle Metadata repository stores Oracle Enterprise Scheduling Service job metadata. Oracle Enterprise Scheduling Service 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.

### 6.10.2 Oracle Enterprise Scheduling Service Logging

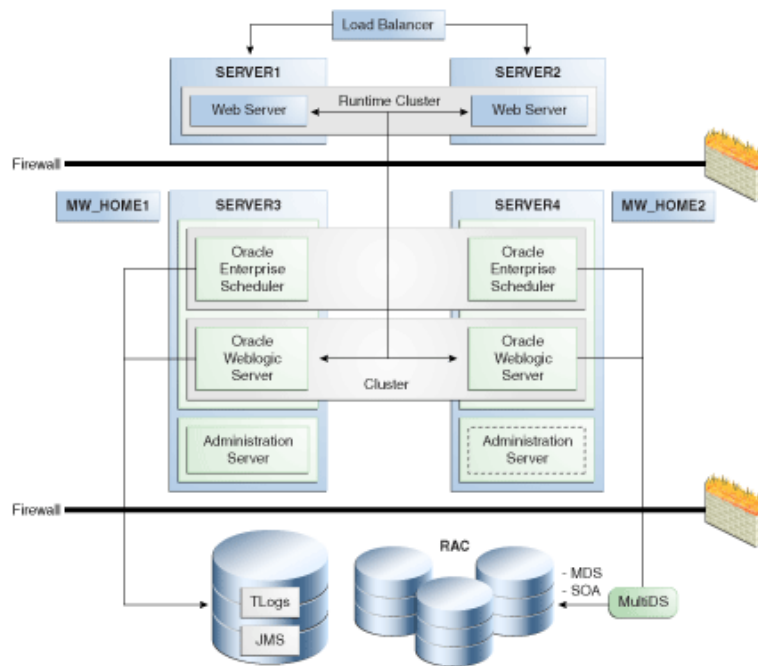
Use standard Oracle WebLogic Server logging for an Oracle Enterprise Scheduling Service cluster. Use logs in Oracle WebCenter Content to examine Oracle Enterprise Scheduling Service behavior. Oracle Enterprise Scheduling Service logging is configured by default in Oracle WebLogic Server.

The default location for log files for Oracle Enterprise Scheduling Service process jobs on UNIX servers is `/tmp/ess/requestFileDirectory`. Oracle Enterprise Scheduling Service 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.

### 6.10.3 Oracle Enterprise Scheduling Service Cluster Architecture

[Figure 6–5](#) is an architectural diagram of a two node Oracle Enterprise Scheduling Service cluster.

**Figure 6–5 Oracle Enterprise Scheduling Service 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 Scheduling Service 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
- A Metadata Store
- An HTTP load balancer is required for web service interactions.

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*.

### 6.10.4 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 Scheduling Service web service interface. This load balancing occurs independently of where the job is scheduled to execute.



As Oracle Enterprise Scheduling Service does not use JMS, no JMS failover is needed. Oracle Enterprise Scheduling Service uses stateless session beans and JCA RAR files rather than stateful session beans. Therefore, no EJB state failover is required.

This section contains the following topics:

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

#### 6.10.4.1 Request Processor Failover

Oracle Enterprise Scheduling Service includes a request processor component, which represents a single Managed Server in the Oracle Enterprise Scheduling Service 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.

#### 6.10.4.2 External Component Failover

Oracle Enterprise Scheduling Service 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 6–9](#) lists the external components that may fail, along with the steps to take to prevent failed Oracle Enterprise Scheduling Service jobs.

**Table 6–9 Oracle Enterprise Scheduling Service 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.
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.

### 6.10.5 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 [Section 5.3.2.1](#) and [Section 5.3.1.1](#).

### 6.10.6 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. Run time 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.

### 6.10.7 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.

## 6.11 Managing an Oracle Enterprise Scheduling Service Cluster

Managing an Oracle Enterprise Scheduling Service 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](#)

### 6.11.1 Starting and Stopping the Cluster

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

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

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

### 6.11.2 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.

### 6.11.3 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.

### 6.11.4 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 Scheduling Services. 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](#)

#### 6.11.4.1 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 [Section 4.2.1.1](#).

#### 6.11.4.2 Death Detection and Restart

In order to enable death detection and recovery, each Oracle Enterprise Scheduling Service 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`.

#### **6.11.4.3 Oracle Java Transaction API Migration and Oracle Java Message Service**

Oracle JTA migration may be necessary if two-phase commits are used. Two phase commit is used if a client with a transaction invokes the Oracle Enterprise Scheduling Service EJB to submit a job and propagates the transaction to the EJB. If this pattern is not used, then JTA migration is not necessary.

Oracle Enterprise Scheduling Service does not use Oracle Java Message Service (JMS) such that JMS recovery is not needed.

---

---

## Troubleshooting Oracle Enterprise Scheduling Service

This appendix describes common problems that you might encounter when using Oracle Enterprise Scheduling Service and explains how to solve them.

- [Introduction to Troubleshooting Oracle Enterprise Scheduling Service](#)
- [Getting Started with Troubleshooting Oracle Enterprise Scheduling Service Jobs](#)
- [Getting Started with Troubleshooting an Oracle Enterprise Scheduling Service 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 Scheduling Service

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

#### Guidelines

When using the information in this chapter, Oracle recommends:

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

## B.2 Getting Started with Troubleshooting Oracle Enterprise Scheduling Service Jobs

You may want to troubleshoot the following typical issues that can arise when running Oracle Enterprise Scheduling Service jobs.

- Asynchronous jobs remain in running state indefinitely.
- An asynchronous job hangs or crashes.
- Oracle Enterprise Scheduling Service is down when the remote scheduled job completes or there are network problems such that Oracle Enterprise Scheduling Service does not receive the completion status from the remote job.
- A scheduled job is ready to execute, but does not execute.
- A scheduled job is placed in manual error recovery state where troubleshooting is needed.
- Oracle Enterprise Scheduling Service is throwing errors.
- A scheduled job ends in error.

For troubleshooting Oracle Enterprise Scheduling Service, use the standard Oracle WebLogic Server system log. For information about viewing job request logs, see [Section 6.7](#)

This section contains the following topics:

- [Troubleshooting Asynchronous Scheduled 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 Scheduling Service 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 Scheduling Service 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 Scheduling Service 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 [Section B.2.1.1](#) and [Section B.2.1.2](#)
- Cancel the job. For more information, see [Section 4.2.4](#).
- Configure a timeout for asynchronous jobs. For more information, see [Section 4.2.1](#).

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 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 [Section B.4.4](#).

[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
```

#### **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: 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]
```

*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 will not be 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: 0000If99plA76EWLHyo2yf1CVJ^i000000,0] [APP:
ADFBCAsyncInvalidCallbackCreds] [MessageID: urn:uuid:7e355428-f68f-4c3f-a368-baf529048323]
```

For more information about configuring parameters for a job, see [Section 4.2.1](#). For more information about viewing the server-diagnostic.log file, see [Section 6.7.6](#).

### 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 [Section 4.2.4](#).

## B.2.2 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 [Section 4.2.4](#).

## B.2.3 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 Scheduling Service 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 Scheduling Service 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 Scheduling Service 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 will continue 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 Scheduling Service, which goes down, along with the associated Perl agent. If the first instance of Oracle Enterprise Scheduling Service will not be back up and running for a while, Oracle Enterprise Scheduling Service 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.2.3.1 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 [Section 4.2.6](#). 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 Scheduling Service server on which the request is running by finding the server and cluster names.

Determine the cluster name from the Oracle Enterprise Scheduling Service 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 [Section 4.2.2](#) for information on performing searches.
3. If canceling the job is not effective, restart the relevant Oracle Enterprise Scheduling Service server:
  - a. From the navigation pane, expand the farm and then **WebLogic Domain**.
  - b. Expand the Oracle Enterprise Scheduling Service cluster and select the Oracle Enterprise Scheduling Service 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.2.3.2 Handling Stuck Asynchronous Jobs Requiring Manual Recovery

When an asynchronous job request requires manual recovery, follow these basic steps. Additional steps will 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 [Section 4.2.4](#) 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 non-Oracle Enterprise Scheduling Service 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. Once the remote job is no longer running, terminate the job request in Oracle Enterprise Scheduling Service, so that Oracle Enterprise Scheduling Service is no longer 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 will be in `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 need to 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.2.4 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 need for manual recovery.

For more information about viewing job request details, see [Section 4.2.4](#).

[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 will contain 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 cancelling a job request, see <a href="#">Section 4.2.6</a> .
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 <a href="#">Section 4.2.5</a> .
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 Scheduling Service, see <a href="#">Section 3.5</a> .
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.	
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 <a href="#">Section 3.5.2</a> .
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 Scheduling Service, see <a href="#">Section 3.5</a> .
READY	NO_BOUND_WORK_ASSIGNMENT	You must bind a specialized work assignment to the job request.	For information about binding a work assignment, see <a href="#">Section 3.4.2</a> .
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 <a href="#">Section 5.3.1.1</a> .

**Table B-1 (Cont.) Job Request States and Associated Diagnostic Codes**

State	Diagnostic Code	Description	Related Documentation
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 <a href="#">Section 5.3.1.1</a> .
RUNNING	PREPROCESS_DELAY	The request is delayed by the pre-processor. Includes the time at which the delay ends.	
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 Scheduling Service server is running (see <a href="#">Section 6.2</a> ). If necessary, restart one of the Oracle Enterprise Scheduling Service components, as described in <a href="#">Section 3.5</a> .
RUNNING	REQUESTED_PROCESSOR_NOT_ACTIVE	Job request processing cannot continue because the server specified by the job request <code>SYS_requestedProcessor</code> property is not available. Includes the name of the requested processor.	Verify that the Oracle Enterprise Scheduling Service server is running (see <a href="#">Section 6.2</a> ). If necessary, restart one of the Oracle Enterprise Scheduling Service components, as described in <a href="#">Section 3.5</a> .
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 Scheduling Service server is running (see <a href="#">Section 6.2</a> ). Start a request processor, as described in <a href="#">Section 3.5.2</a> .
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 Scheduling Service server is running (see <a href="#">Section 6.2</a> ). Restart the request processor, as described in <a href="#">Section 3.5.2</a> .
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.	

**Table B-1 (Cont.) Job Request States and Associated Diagnostic Codes**

State	Diagnostic Code	Description	Related Documentation
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 <a href="#">Section 5.3.1.1</a> .
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 Scheduling Service server is running (see <a href="#">Section 6.2</a> ). If necessary, restart one of the Oracle Enterprise Scheduling Service components, as described in <a href="#">Section 3.5</a> .
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 <a href="#">Section 3.4.2</a> .
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 <a href="#">Section 5.3.1.1</a> .
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 <a href="#">Section 4.2.5</a> .

**Table B-1 (Cont.) Job Request States and Associated Diagnostic Codes**

State	Diagnostic Code	Description	Related Documentation
WAIT	DEFERRED	The job request is an instance of a recurrence for which the previous recurrence instance is still running. Oracle Enterprise Scheduling Service prevents concurrent execution of recurrent instances, and the next recurrence remains in WAIT state while the current recurrence is active.	
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 <a href="#">Section 3.5.2</a>
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 Scheduling Service components. For more information, see <a href="#">Section 3.5.2</a> and <a href="#">Section 3.5.1</a> .
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 Scheduling Service*.

---

## B.3 Getting Started with Troubleshooting an Oracle Enterprise Scheduling Service Cluster

Troubleshooting an Oracle Enterprise Scheduling Service clustered environment involves the following.

- [Finding Performance and Scalability Issues](#)
- [Using a Shared Database](#)
- [Tuning Oracle Enterprise Scheduling Service 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 Scheduling Service 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 Scheduling Service 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 [TODO: new link. was 9.7].

### 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 Scheduling Service systems.

### B.3.3 Tuning Oracle Enterprise Scheduling Service System Performance

The following potential performance and scalability issues may occur in the context of job requests or Oracle Enterprise Scheduling Service runtime.

- Jobs are saturating the CPU of Oracle Enterprise Scheduling Service 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 Scheduling Service 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 Scheduling Service 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. Otherwise, the job will 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 Scheduling Service instance. One Oracle Enterprise Scheduling Service 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 [Section 5.3.2.1](#)

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 [Section 5.2.3.2](#).

### B.3.3.4 Tuning Oracle Enterprise Scheduling Service for Optimal Performance

You can tune the following Oracle Enterprise Scheduling Service components:

- Request dispatcher
- Request processor
- Connection pool size
- RDBMS Scheduler

### Tuning the Dispatcher

The dispatcher tuning parameters apply to the Oracle Enterprise Scheduling Service request dispatcher. The request dispatcher manages requests that are awaiting their scheduled execution. The request processor handles the job requests once they have executed.

Parameters are as follows.

- **Dispatcher Enabled:** Indicates whether the request dispatcher is enabled on the Oracle Enterprise Scheduling Service server. When disabled, that Oracle Enterprise Scheduling Service server will 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 Scheduling Service 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 Scheduling Service server. If disabled, the Oracle Enterprise Scheduling Service server will 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 Scheduling Service 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 will be 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 Scheduling Service 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 Scheduling Service Internal Data Source

The connection pool size for the Oracle Enterprise Scheduling Service 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 Scheduling Service spawned jobs connect to the database using SQL\*Net. If the spawned jobs are canceled, Oracle Enterprise Scheduling Service 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 Scheduling Service. 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 Scheduling Service Run-Time System Is Throwing Errors](#)
- [Oracle Enterprise Scheduling Service Is Running Out Of Database Connections](#)
- [Job Queue Full Due to a Hanging Job](#)

In addition to the recommended solutions, consider reviewing [Section B.3.3](#) 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 Scheduling Service 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 [Section 3.5.2](#).
  - 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 [Section 3.5.2](#).
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 [Section 3.5.1](#).
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 [Section 3.4.2](#).
  - From the **Scheduling Service** menu, choose **Work Allocation > Work Assignments** to review the configuration in the Work Assignments page. See [Section 5.3.1](#).

## 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 Scheduling Service server has crashed and recovery has not taken place.

To resolve this problem, determine the current status of Oracle Enterprise Scheduling Service 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 [Section 3.5.2](#).
2. Verify the Oracle Enterprise Scheduling Service 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 Scheduling Service cluster.
  - c. From the WebLogic Cluster page, in the **Servers** section, view the **Status** column to determine if the Oracle Enterprise Scheduling Service 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 [Section 4.2.4](#).  
When an Oracle Enterprise Scheduling Service 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 Scheduling Service 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 [Section B.2.1](#).

### 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 Scheduling Service 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 [Section 4.2.2](#).
2. On the Request Details page, from the **Action** menu, select **Request Log** to view the log message. See [Section 4.2.4](#) 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 Scheduling Service 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 Scheduling Service 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. [Section 6.7.4](#).
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 Scheduling Service, manually complete the job in the Request Details page. For related information, see [Section B.2.1](#).

## 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 Scheduling Service 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 [Section 4.2.2](#).
2. On the Request Details page, from the **Action** menu, select **Request Log** to view the log message. See [Section 4.2.4](#) 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 Scheduling Service 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 Scheduling Service 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 Scheduling Service, manually complete the job in the Request Details page. For related information, see [Section B.2.1](#).

## 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 Scheduling Service 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 Scheduling Service by their job names. Job definition names are available from the Request Details page in the Fusion Middleware Control associated with the request. See [Section 4.2.4](#).

To resolve this issue, troubleshoot the native job. For more information, see [Section B.2.1.2](#).



**Problem 2**

Oracle Enterprise Scheduling Service 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 Scheduling Service 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 [Section 5.3.2](#).

## 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 fail with an error, the Request Details page shows the reason and provides access to the job request log from the **Action** menu. See [Section 4.2.4](#).

For more information on job diagnostics, see [Section B.2.4](#).

## 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 [Section B.2.3](#).

**Solution**

To resolve this issue, manually update the job status to complete it. See [Section B.2.3.1](#) and [Section B.2.3.2](#).

## 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 [Section 4.2.4](#) 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 will be subject to auto-retries if configured. From Fusion Middleware Control, perform the following steps:
  - a. Search for the request, as described in [Section 4.2.2](#). 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 [Section 4.2.4](#) 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 will be shown on Request Details page.
  - c. Manually update the job status to complete it. See [Section B.2.3.1](#) and [Section B.2.3.2](#).

For more information on manually recovering a job, see [Section B.2.3](#)

## 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 Scheduling Service will attempt to kill the RDBMS scheduler job. For spawned process, Oracle Enterprise Scheduling Service will try to kill the running process. If the job is successfully killed, the request will transition to CANCELLED state. If the job completes before it can be killed, the state to which the request will transition depends on the result of the job execution. For these type of jobs, this issue should not occur.
- Asynchronous Java job: The request was canceled, but the remote job never contacted Oracle Enterprise Scheduling Service with its terminal status. This could happen if the job is still executing because either `AsyncCancellable` 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 `Cancellable()` interface was not implemented or the job's `Executable.execute()` method still did not return after `Cancellable.cancel` was invoked. For more information about the `Executable` interface, see the chapter "Use Case Oracle Enterprise Scheduling Service 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 will remain in 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 Scheduling Service 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 [Section 4.2.2](#).
2. On the Request Details page, from the **Action** menu, select **Recover Stuck Request**. See [Section 4.2.4](#) 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 [Section 5.2.3](#).
- Schedule the jobs at different times. See [Section 4.2.1](#).

**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 will 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 will see in the job history over time that no jobs are running on this server.
4. Search for the request, as described in [Section 4.2.2](#), 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 Scheduling Service 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 [Section 3.4.1](#).

**B.4.13 Oracle Enterprise Scheduling Service Run-Time System Is Throwing Errors****Problem**

Oracle Enterprise Scheduling Service 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 Scheduling Service system logs to troubleshoot this issue. From Fusion Middleware Control, perform the following:

1. Search for the request, as described in [Section 4.2.2](#).
2. On the Request Details page, from the **Action** menu, select **Request Log** to view the log message. See [Section 4.2.4](#) 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 Scheduling Service cluster, and select the Oracle Enterprise Scheduling Service 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 Scheduling Service 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 Scheduling Service and it will inherit 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 Scheduling Service 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 [Section 6.7.4](#).

---

---

**Note:** The logger only shows logs written by Oracle Enterprise Scheduling Service job running in Oracle WebLogic Server. Once Oracle Enterprise Scheduling Service 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 Scheduling Service logs as they run in a separate process.

---

---

## B.4.14 Oracle Enterprise Scheduling Service Is Running Out Of Database Connections

### Problem

If Oracle Enterprise Scheduling Service is running out of database connections, there could be a problem with connection leaks in Oracle Enterprise Scheduling Service.

### 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 [Section 3.4.2](#).

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 Scheduling Service server, leaving the other servers to process the rest of the jobs. See [Section 5.3.1.1](#).

4. Restart the Oracle Enterprise Scheduling Service server:
  - a. From the navigation pane, expand the farm and then **WebLogic Domain**.
  - b. Expand the Oracle Enterprise Scheduling Service cluster and select the Oracle Enterprise Scheduling Service 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 Scheduling Service 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 [Section 4.2.6](#).

- At times, heavy database inserts and updates will 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 minutes 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 Scheduling Service server will process only 25 PL/SQL jobs concurrently. Since each PL/SQL job is taking one to two hours to complete, the remaining PL/SQL jobs will be 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>.