

## **Oracle® Fusion Middleware**

Scheduling Jobs Guide for Oracle Business Intelligence  
Enterprise Edition

11g Release 1 (11.1.1)

**E18562-02**

December 2011

Explains how to manage Oracle Business Intelligence Enterprise Edition Scheduler and Job Manager, including how to configure Scheduler, how to use Job Manager and its menus, and how to program scripts and Java Jobs. Includes information about the SA System subject area.

Oracle Fusion Middleware Scheduling Jobs Guide for Oracle Business Intelligence Enterprise Edition, 11g Release 1 (11.1.1)

E18562-02

Copyright © 2010, 2011, Oracle and/or its affiliates. All rights reserved.

Primary Author: Nick Fry

Contributors: Oracle Business Intelligence development, product management, and quality assurance teams.

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

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

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

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

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

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

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd.

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> .....	vii
Audience .....	vii
Documentation Accessibility .....	vii
Related Documents .....	viii
System Requirements and Certification .....	viii
Conventions .....	viii
<b>New Features in Scheduling Jobs for Oracle Business Intelligence</b> .....	ix
New Features for Oracle BI EE 11g Release 1 (11.1.1.6) .....	ix
New Features for Oracle BI EE 11g Release 1 (11.1.1.5) .....	ix
New Features for Oracle BI EE 11g Release 1 (11.1.1.3) .....	ix
<b>1 Introducing Oracle BI Scheduler</b>	
1.1 What is Oracle BI Scheduler? .....	1-1
1.2 About Oracle BI Scheduler Server Components .....	1-2
1.3 About Agent Support in Oracle BI Scheduler .....	1-2
1.4 About Database Support in Oracle BI Scheduler .....	1-2
1.5 About Oracle BI Scheduler Schemas .....	1-3
1.6 Topics of Interest in Other Guides .....	1-4
<b>2 Configuration Tasks for Oracle BI Scheduler</b>	
2.1 Creating an Oracle BI Scheduler Database and Tables Using RCU .....	2-1
2.2 Configuring a Database for the Oracle BI Scheduler .....	2-1
2.2.1 Using Fusion Middleware Control to Configure a Database for the Oracle BI Scheduler .....	2-2
2.2.2 Configuring a System DSN entry for SQL Server Databases .....	2-3
2.3 Configuring Oracle BI Scheduler Instances in a Clustered Environment .....	2-3
2.4 Changing Oracle BI Scheduler Table Names .....	2-4
<b>3 Using Oracle BI Scheduler Job Manager</b>	
3.1 Opening Job Manager .....	3-1
3.2 Adding Oracle BI Scheduler Jobs in Job Manager .....	3-2
3.3 Modifying Oracle BI Scheduler Jobs in the Job Manager .....	3-2
3.4 About Adding Agent Jobs .....	3-2

3.5	Modifying Agents in Job Manager .....	3-3
3.6	Re-Running a Job Instance.....	3-3
3.7	Managing Oracle BI Scheduler Job Instances .....	3-4
3.7.1	Viewing or Refreshing Oracle BI Scheduler Job Instances .....	3-4
3.7.2	Cancelling Oracle BI Scheduler Job Instances .....	3-4
3.7.3	Purging Oracle BI Scheduler Job Instances.....	3-5
3.7.3.1	Using the Job Manager Instances List .....	3-5
3.7.3.2	Using the Job Manager Purge Instances Window .....	3-5

## 4 Programming BI Scheduler VBScript and JScript Jobs

4.1	Configuring Custom Script Properties for Oracle BI Scheduler .....	4-2
4.2	Creating a Custom Script Example - Cache Clearance.....	4-2
4.3	Configuring Custom Script Properties for Oracle BI Delivers .....	4-4
4.4	Creating a Custom Script Example - Copy Results to the File System .....	4-5
4.5	Oracle BI Scheduler Read-Only Script Object Properties .....	4-5
4.6	Oracle BI Scheduler Read/Write Script Object Properties .....	4-6
4.7	Oracle BI Scheduler Script-Defined Constants.....	4-7
4.7.1	Severity Constants .....	4-7
4.7.2	DayEnum Constants.....	4-7
4.7.3	DayOfWeekEnum Constants .....	4-7
4.7.4	JobFlagsEnum Constants.....	4-8
4.7.5	MonthEnum Constants .....	4-8
4.7.6	OccurrenceEnum Constants .....	4-9
4.8	Oracle BI Scheduler Script Object Methods and Events .....	4-9
4.8.1	CreateArray Method .....	4-10
4.8.2	DeregisterCancelCommand Method .....	4-11
4.8.3	GetConfigurationValue Method.....	4-11
4.8.4	GetTempFileName Method .....	4-11
4.8.5	LaunchProcess Method.....	4-12
4.8.6	RegisterCancelCommand Method .....	4-12
4.8.7	ScheduleJobDaily Method.....	4-13
4.8.8	ScheduleJobMonthlyDate Method .....	4-13
4.8.9	ScheduleJobMonthlyDOW Method .....	4-14
4.8.10	ScheduleJobNow Method.....	4-15
4.8.11	ScheduleJobOnce Method .....	4-16
4.8.12	ScheduleJobWeekly Method .....	4-16
4.8.13	OnError Event .....	4-17
4.9	Troubleshooting JScript and VBScript Job Failures .....	4-18

## 5 Programming BI Scheduler Java Jobs

5.1	Using Oracle BI Scheduler Java Jobs.....	5-1
5.1.1	Adding and Configuring Custom Java Jobs in Job Manager .....	5-2
5.1.2	Example: Creating a Java Program for Agents.....	5-2
5.1.3	Example: Configuring a Java Program.....	5-3
5.2	Oracle BI Scheduler Java Jobs .....	5-4
5.3	Adding Java Jobs for Oracle BI Scheduler.....	5-4
5.4	Oracle BI Scheduler Custom Java Program Package.....	5-5

5.5	SchedulerJavaExtension Interface .....	5-5
5.6	SchedulerJobInfo Interface .....	5-5
5.7	SchedulerJobException Class .....	5-6
5.8	Oracle BI Scheduler Java Extension Example .....	5-7

## 6 Oracle BI Scheduler Job Manager Menus

6.1	About the Job Manager .....	6-1
6.2	Toolbar Menus in Job Manager.....	6-1
6.2.1	File Menu in Job Manager .....	6-1
6.2.2	Service Management Menu in Job Manager.....	6-2
6.2.3	Jobs Menu in Job Manager .....	6-2
6.2.4	Instances Menu in Job Manager .....	6-3
6.2.5	Instance Properties in Job Manager .....	6-3
6.3	General Oracle BI Scheduler Job Properties.....	6-4
6.4	Job Action Properties Available in Job Manager.....	6-5
6.5	Job Triggers in Job Manager .....	6-6
6.5.1	Single-Run Triggers.....	6-7
6.5.2	Recurrent Triggers .....	6-7
6.5.2.1	Recurrent Trigger Types.....	6-8

## A Setting Up the SA System Subject Area

A.1	About the SA System Subject Area .....	A-1
A.1.1	About Group and Application Role Resolution.....	A-2
A.2	Setting Up the Data Source for the SA System Subject Area.....	A-2
A.3	Importing SA System Data Into the Repository .....	A-4
A.4	Setting Configuration Options for the SA System Subject Area .....	A-5
A.4.1	Managing the Case of Login Names for the SA System Subject Area .....	A-6

## Index



---

---

# Preface

The Oracle Business Intelligence Foundation Suite is a complete, open, and integrated solution for all enterprise business intelligence needs, including reporting, ad hoc queries, OLAP, dashboards, scorecards, and what-if analysis. The Oracle Business Intelligence Foundation Suite includes Oracle Business Intelligence Enterprise Edition.

Oracle Business Intelligence Enterprise Edition (Oracle BI EE) is a comprehensive set of enterprise business intelligence tools and infrastructure, including a scalable and efficient query and analysis server, an ad-hoc query and analysis tool, interactive dashboards, proactive intelligence and alerts, and an enterprise reporting engine.

The components of Oracle BI EE share a common service-oriented architecture, data access services, analytic and calculation infrastructure, metadata management services, semantic business model, security model and user preferences, and administration tools. Oracle BI EE provides scalability and performance with data-source specific optimized request generation, optimized data access, advanced calculation, intelligent caching services, and clustering.

This guide contains information about configuring the scheduler, using scripts and Java jobs and understanding the Job Manager menu.

## Audience

This guide is intended for administrators who are responsible for managing Job scheduling in Oracle Business Intelligence.

## Documentation Accessibility

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

### Access to Oracle Support

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

## Related Documents

For more information, see the following documents in the Oracle Business Intelligence Enterprise Edition 11g Release 1 (11.1.1) documentation set:

- the Oracle Business Intelligence chapter in the *Oracle Fusion Middleware Release Notes*
- *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*
- *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Developer's Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Integrator's Guide for Oracle Business Intelligence Enterprise Edition*

## System Requirements and Certification

Refer to the system requirements and certification documentation for information about hardware and software requirements, platforms, databases, and other information. Both of these documents are available on Oracle Technology Network (OTN).

The system requirements document covers information such as hardware and software requirements, minimum disk space and memory requirements, and required system libraries, packages, or patches:

[http://www.oracle.com/technology/software/products/ias/files/fusion\\_requirements.htm](http://www.oracle.com/technology/software/products/ias/files/fusion_requirements.htm)

The certification document covers supported installation types, platforms, operating systems, databases, JDKs, and third-party products:

[http://www.oracle.com/technology/software/products/ias/files/fusion\\_certification.html](http://www.oracle.com/technology/software/products/ias/files/fusion_certification.html)

## Conventions

The following text conventions are used in this document:

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

---

---

# New Features in Scheduling Jobs for Oracle Business Intelligence

This preface describes changes to scheduling jobs features for Oracle Business Intelligence Enterprise Edition 11g Release 1 (11.1.1).

If you are upgrading to Oracle BI EE from a previous release, read the following information carefully, because there might be significant differences in features, tools, and procedures. For more information about upgrading to Oracle BI EE 11g, see *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence Enterprise Edition*.

This preface contains the following topics:

- [New Features for Oracle BI EE 11g Release 1 \(11.1.1.6\)](#)
- [New Features for Oracle BI EE 11g Release 1 \(11.1.1.5\)](#)
- [New Features for Oracle BI EE 11g Release 1 \(11.1.1.3\)](#)

## New Features for Oracle BI EE 11g Release 1 (11.1.1.6)

There are no new scheduling jobs features in Oracle BI EE 11g Release 1 (11.1.1.6).

## New Features for Oracle BI EE 11g Release 1 (11.1.1.5)

There are no new scheduling jobs features in Oracle BI EE 11g Release 1 (11.1.1.5).

## New Features for Oracle BI EE 11g Release 1 (11.1.1.3)

New scheduling jobs features in Oracle BI EE 11g Release 1 (11.1.1.3) include:

- [Creating an Oracle BI Scheduler Database and Tables Using RCU](#)
- [Using Fusion Middleware Control to Configure a Database for the Oracle BI Scheduler](#)

### **Creating an Oracle BI Scheduler Database and Tables Using RCU**

In this release, you can create an Oracle BI Scheduler database and tables using the Oracle Repository Creation Utility (RCU).

For information about using RCU, see [Section 2.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU"](#).

## **Using Fusion Middleware Control to Configure a Database for the Oracle BI Scheduler**

In this release, you can perform many system administration tasks using the centralized management capabilities of Fusion Middleware Control.

For more information about using Fusion Middleware Control to configure a database for the Oracle BI Scheduler, see [Section 2.2.1, "Using Fusion Middleware Control to Configure a Database for the Oracle BI Scheduler"](#).

---

---

# Introducing Oracle BI Scheduler

This chapter provides general information about Oracle BI Scheduler. It contains the following topics:

- [Section 1.1, "What is Oracle BI Scheduler?"](#)
- [Section 1.2, "About Oracle BI Scheduler Server Components"](#)
- [Section 1.3, "About Agent Support in Oracle BI Scheduler"](#)
- [Section 1.4, "About Database Support in Oracle BI Scheduler"](#)
- [Section 1.5, "About Oracle BI Scheduler Schemas"](#)
- [Section 1.6, "Topics of Interest in Other Guides"](#)

## 1.1 What is Oracle BI Scheduler?

Oracle BI Scheduler is a server that manages and schedules jobs. Oracle BI Scheduler supports two kinds of jobs:

- Scripted jobs.

Scripted jobs are configured and submitted using the Job Manager feature of the Oracle BI Administration Tool. For example, a scripted job could periodically load the Oracle BI Server usage statistics into a back-end database. In this example Oracle BI Scheduler communicates with the BI Server. However, scripted jobs might not access the BI Server, for example by saving the output of an agent to a shared drive. Scripted jobs can also be configured through agents and actions.

Oracle BI Scheduler supports jobs that are written in the Java programming language or in the VBScript and JScript scripting languages.

---

---

**Note:** Scripting for agents and scripts that are defined by the Oracle BI Scheduler Job Manager are supported only under Windows platforms. The Java interfaces support all platforms.

---

---

- Agents.

Agents deliver content to end users. Content can be analyses, dashboards, briefing books, or alerts. After delivering content, agents can also execute actions. Actions include Java actions, URL actions, Web service actions, and server script actions. Agents can also run other agents, creating chains of agents. Agents are configured and submitted for execution using Oracle BI Delivers. Oracle BI Scheduler communicates with Oracle BI Presentation Services for unscripted jobs.

---



---

**Note:** There are thus two different forms of custom Java that can be executed: Java scripted jobs run in their own right and Java actions run as part of an agent.

---



---

## 1.2 About Oracle BI Scheduler Server Components

Oracle BI Scheduler consists of the following components:

- Oracle BI Scheduler Job Manager
- Oracle BI Scheduler Service process:
  - Windows operating systems: `nqscheduler.exe`
  - UNIX operating systems: `nqscheduler`
- Scheduler tab in Fusion Middleware Control
- Command line job invocation tool:
  - Windows operating systems: `saschinvoke.exe`
  - UNIX operating systems: `saschinvoke`

## 1.3 About Agent Support in Oracle BI Scheduler

When a user creates and schedules an agent, Oracle BI Presentation Services gathers information about the agent such as its priority, the intended recipients, and the devices to which content should be delivered. Presentation Services packages this information and other characteristics into a job, then informs Oracle BI Scheduler when to execute the job.

Agents can run in parallel on different threads. The number of agents that can run in parallel depends on the size of the Scheduler thread pool (a configurable setting) and the number of threads used up by each agent. Queueing might occur if too many agents are triggered at the same time.

For information about agents, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

## 1.4 About Database Support in Oracle BI Scheduler

Oracle BI Scheduler uses a single back-end database to store pertinent information about a job, its instances, and its parameters. For information, see "[System Requirements and Certification](#)".

The Oracle BI Scheduler service starts only if the back-end database satisfies the following requirements:

Back-End Database Requirement	Notes
Configured	There is a one-to-one relationship between the back-end database schema and Oracle BI Scheduler. Do not configure multiple Oracle BI Scheduler applications to use a single database schema.
Operational	For information about the specific back-end databases that are supported by Oracle BI Scheduler, see " <a href="#">System Requirements and Certification</a> ".

Back-End Database Requirement	Notes
Mixed Authentication Mode	<p>For example, if the database is SQL Server, then the security mode should be set to allow both SQL Server and Windows Authentication for logon.</p> <p>Do not use operating system authentication for the back-end database login. If you do, then the Oracle BI Scheduler service might not start in some cases.</p>

## 1.5 About Oracle BI Scheduler Schemas

You create the database schemas that are associated with Oracle BI Scheduler using the Repository Creation Utility (RCU). (For information, see [Section 2.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU."](#)) You can also create tables in Oracle BI Scheduler's schema for use with other major commercial databases, by running the SQL scripts in the following directory:

```
\oraInst\bifoundation\OracleBISchedulerComponent\coreapplication_
obisch1\schema\.
```

---

**Note:** To upgrade Scheduler tables, see *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence Enterprise Edition*.

---

[Table 1–1](#) gives brief descriptions of the database tables that are used by Oracle BI Scheduler.

If usage tracking is enabled in the NQSCfg.INI file, then the Oracle BI Server generates Usage Tracking data files. A sample JavaScript is provided, which extracts information from the Usage Tracking files and loads them to a table in the relational database. The S\_NQ\_ACCT table stores all the information regarding Accounting Data. For information, see *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition "Managing Usage Tracking"*.

**Table 1–1 Tables Used by Oracle BI Scheduler**

Table Name	Table Description
S_NQ_JOB	Stores information about scheduled jobs.
S_NQ_INSTANCE	Stores information about scheduled job instances.
S_NQ_ERR_MSG	Stores error messages about Oracle BI Scheduler job instances that do not complete successfully. Also stores information that coordinates restarting an agent, if there is a Scheduler failover during agent execution.
S_NQ_JOB_PARAM	Stores information about Oracle BI Scheduler job parameters for scheduled jobs. For example, agent path.

The following are characteristics of Oracle BI Scheduler schema tables:

- The contents of the schema tables are implementation-specific. For information about creating these tables, see *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*.
- Schemas store path and file names to job scripts, which provide easy updates to several jobs if they share a script.

## 1.6 Topics of Interest in Other Guides

Some topics that might be of interest to security administrators are covered in other guides. [Table 1-2](#) lists these topics and indicates where to go for more information.

**Table 1-2 Topics Covered in Other Guides**

<b>Topic</b>	<b>Where to Go for More Information</b>
Configuration settings affecting agents	<i>Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition</i>
Configuring data sources	<i>Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition</i>
Information about security, including configuring SSO and SSL	<i>Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition</i>
Installing and upgrading	<i>Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence</i> <i>Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence Enterprise Edition</i>

---

---

## Configuration Tasks for Oracle BI Scheduler

Depending on your specific deployment, you must perform the following Oracle BI Scheduler configuration tasks:

- [Section 2.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU"](#)
- [Section 2.2, "Configuring a Database for the Oracle BI Scheduler"](#)
- [Section 2.3, "Configuring Oracle BI Scheduler Instances in a Clustered Environment"](#)
- [Section 2.4, "Changing Oracle BI Scheduler Table Names"](#)

Keep the following points in mind:

- If you are not using Oracle BI Scheduler, then you do not need the information in this chapter.
- Configuration that is required for running agents (as opposed to running jobs in general) is described in "Configuring and managing Agents" in *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.
- If you are migrating an Oracle Business Intelligence environment to a new system, then ensure that you also migrate the Oracle Business Intelligence Server repository file and the Oracle BI Scheduler tables. For information, see "Moving Between Environments" in *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*. The Oracle BI Scheduler tables are required for agents.

### 2.1 Creating an Oracle BI Scheduler Database and Tables Using RCU

You use the Repository Creation Utility (RCU) to create Oracle BI Scheduler database schemas and tables. The RCU is a graphical tool that enables you to create and manage Oracle Fusion Middleware database schemas in your database.

The RCU enables you to use either an existing database schema or to create a database schema in which to store the Oracle BI Scheduler tables.

For information about using the RCU, see *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*.

### 2.2 Configuring a Database for the Oracle BI Scheduler

You can use Fusion Middleware Control to configure common settings that are used by agents with Oracle BI Scheduler.

The following procedures describe how to configure the back-end database and tables:

- [Section 2.2.1, "Using Fusion Middleware Control to Configure a Database for the Oracle BI Scheduler."](#)
- [Section 2.2.2, "Configuring a System DSN entry for SQL Server Databases."](#)

## 2.2.1 Using Fusion Middleware Control to Configure a Database for the Oracle BI Scheduler

The Data Source Name that is specified in the Fusion Middleware Control configuration must match the database service alias that was created in [Section 2.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU."](#)

The following procedure configures database connection details. Before you begin this procedure, ensure that you are familiar with the information in "Using Fusion Middleware Control to Update Oracle Business Intelligence Configuration Settings" *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*

### To use Fusion Middleware Control to configure a database for the Oracle BI Scheduler:

1. Go to the Business Intelligence Overview page, as described in "Using Fusion Middleware Control to Manage Oracle Business Intelligence System Components" *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.
2. Display the Scheduler tab of the Deployment page.
3. Click **Lock and Edit Configuration** to enable changes to be made.
4. Complete the elements using the descriptions in the Help topic for the page. Click the **Help** button on the page to access the page-level help for the following options:
  - **Database list**
  - **Call Interface list**

The Call Interface is updated automatically according to the Database chosen.
  - **Data Source option**

For SQL Server databases, you must have a system DSN configured. For information, see [Section 2.2.2, "Configuring a System DSN entry for SQL Server Databases."](#)
  - **Username option**

Enter the user name that was created in the procedure that is described in [Section 2.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU."](#)
  - **Password option**

Enter the password of the existing scheduler schema user. For information, see [Section 2.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU."](#)
  - **Confirm Password option**

Password details are not checked against the database.
5. Click **Apply**, then click **Activate Changes**.

6. Return to the Business Intelligence Overview page and click **Restart**.

See *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition* "Configuring and Managing Agents" for information about advanced configuring settings for agents.

For information on using Oracle BI Systems Management API methods to change configuration settings, see "Introducing the Oracle BI Systems Management API" *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*

## 2.2.2 Configuring a System DSN entry for SQL Server Databases

For SQL Server databases, the Data Source Name (DSN) that is used in the Fusion Middleware Control Scheduler configuration must match an existing ODBC DSN for the SQL Server S\_NQ\_SCHED database that is used in [Section 2.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU."](#)

If you do not have a System DSN entry, then create a new one as described in the following procedure.

**To configure the SQL Server database DSN entry:**

1. From the Windows Start menu, select **Settings**, then **Control Panel**, then **Administrative Tools**, then **Data Sources (ODBC)**.
2. Start the ODBC Data Source Administrator.
3. Select the System DSN tab, and click **Add**.
4. Select the driver SQL Server, and click **Finish**.
5. In the Create a New Data Source to SQL Server wizard, do the following:
  - a. Enter a name and description for the data source.
  - b. Select the SQL Server from the Server list, and click **Next**.
  - c. For server verification of the login ID authenticity, select the appropriate authentication for the S\_NQ\_SCHED SQL Server database schema that was created in [Section 2.1, "Creating an Oracle BI Scheduler Database and Tables Using RCU."](#) Click **Next**.
6. Select the **Change the default database to** field and select the S\_NQ\_SCHED database from the list. Click **Next**.
7. Update any language or log file settings if appropriate, and click **Finish**.
8. To verify the connection settings, click the **Test Data Source** button, and click **OK**.
9. Click **OK** to exit ODBC Data Source Administrator.
10. Use Fusion Middleware Control to configure the SQL Server database as described in [Section 2.2.1, "Using Fusion Middleware Control to Configure a Database for the Oracle BI Scheduler."](#)

## 2.3 Configuring Oracle BI Scheduler Instances in a Clustered Environment

In a clustered environment you can have zero, one, or two scheduler instances. Configure zero if you do not require support for scheduled jobs or agents. Configure one if you do not require high availability. Configure two to support high availability.

For information, see "Using Fusion Middleware Control to Configure Primary and Secondary Instances" in *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

## 2.4 Changing Oracle BI Scheduler Table Names

You can change the names of the tables that the Oracle BI Scheduler uses. This step is required only if the default table names are incompatible with the database setup. Custom table names are configured by adding settings to the Oracle BI Scheduler configuration files for each Scheduler instance. You must make the same changes for each Scheduler instance. Using different table names for the two Scheduler instances results in inconsistent behavior when the active Scheduler changes.

Use various elements in the `instanceconfig.xml` file to change these settings.

The following procedure describes how to change the Oracle BI Scheduler table names. For this procedure, a new tag, `DB_Column_NamesType`, has been created as an example.

When using database scripts to create the Scheduler tables, the changed table names should be included in the scripts.

Before you begin this procedure, ensure that you are familiar with the information in "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings" in *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

### To manually change the settings for the Oracle BI Scheduler table names:

1. Open the Oracle BI Scheduler version of the `instanceconfig.xml` file for editing, as described in "Where are Configuration Files Located" in *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.
2. Create a new element named `DB_Column_NamesType` as shown in the following example from an `instanceconfig.xml` file:

```
<xs:complexType name="DB_Column_NamesType">
<xs:all>
<xs:element name="TABLE_JOBS" type="xs:string" default="S_NQ_JOB"
minOccurs="0">
<xs:annotation>
<xs:documentation xml:lang="en">
The name of the table used as the jobs table in the back-end DB.
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="TABLE_INSTANCES" type="xs:string" default="S_NQ_INSTANCE"
minOccurs="0">
<xs:annotation>
<xs:documentation xml:lang="en">
The name of the table used as the instances table in the back-end DB.
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="TABLE_PARAMS" type="xs:string" default="S_NQ_JOB_PARAM"
minOccurs="0">
<xs:annotation>
<xs:documentation xml:lang="en">
The name of the table used as the job parameters table in the back-end DB.
</xs:documentation>
</xs:annotation>
</xs:element>
```

```

</xs:element>
<xs:element name="TABLE_ERRMSG" type="xs:string" default="S_NQ_ERR_MSG"
minOccurs="0">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      The name of the table used to store information about job instances that do
      not complete successfully in the back-end DB.
    </xs:documentation>
  </xs:annotation>
</xs:element>

```

3. For each of the entries under the tag `DB_Column_NamesType`, add the parameter and string values that are shown in [Table 2-1](#). The values that are created in the data string become the values that are used for the Oracle BI Scheduler table names.

**Table 2-1 Parameters for DB Column Names in `instanceconfig.xml`**

Parameter Name	Type	String Value
TABLE_JOBS	REG_SA	S_NQ_JOB
TABLE_INSTANCES	REG_SA	S_NQ_INSTANCE
TABLE_PARAMS	REG_SA	S_NQ_JOB_PARAM
TABLE_ERRMSG	REG_SA	S_NQ_ERR_MSG

---

**Note:** The data types for each column should remain true to the intent of the schema. For example, if the job ID is defined as an integer type, do not change it to a varchar type. However, increasing the number of characters in a varchar column is an acceptable change.

---

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.



---

---

## Using Oracle BI Scheduler Job Manager

This chapter describes how to manage Oracle BI Scheduler using the Job Manager feature in the Oracle BI Administration Tool. This chapter contains the following topics:

- [Section 3.1, "Opening Job Manager"](#)
- [Section 3.2, "Adding Oracle BI Scheduler Jobs in Job Manager"](#)
- [Section 3.3, "Modifying Oracle BI Scheduler Jobs in the Job Manager"](#)
- [Section 3.4, "About Adding Agent Jobs"](#)
- [Section 3.5, "Modifying Agents in Job Manager"](#)
- [Section 3.6, "Re-Running a Job Instance"](#)
- [Section 3.7, "Managing Oracle BI Scheduler Job Instances"](#)

### 3.1 Opening Job Manager

The Job Manager is a Windows tool that is the interface with the Oracle BI Scheduler. Through Job Manager, you can connect to, start and stop the Oracle BI Scheduler, add and manage jobs, and manage job instances. You create and edit agents using Oracle BI Presentation Services. For information, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

The following procedure describes how to open Job Manager from the Oracle BI Administration Tool.

**To open the Job Manager:**

1. Open the Oracle BI Administration Tool.
2. From the toolbar, click the **Manage** menu, and select **Jobs**.  
Job Manager opens in a new window.
3. In Job Manager, from the toolbar, click the **File** menu, and select **Open Scheduler Connection**.
4. In the Machine Name dialog box, enter the machine name and port number where the Oracle BI Scheduler is located, and click **OK**.

You can learn the machine name (host) and port number by displaying the Availability tab of the Capacity Management page in Fusion Middleware Control. For information, see "Using Fusion Middleware Control to Configure Primary and Secondary Instances" in *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

## 3.2 Adding Oracle BI Scheduler Jobs in Job Manager

Job Manager enables you to add new scripted jobs. Agents can only be created in Oracle BI Presentation Services.

Add an Oracle BI Scheduler job in the Job Manager using the following procedure.

**To add an Oracle BI Scheduler job in the Job Manager:**

1. In the Job Manager, from the **Jobs** menu, select **Add New Job**.
2. Enter the appropriate information in the dialog box.

See the following sections for field descriptions:

- [Section 6.3, "General Oracle BI Scheduler Job Properties"](#)
- [Section 6.4, "Job Action Properties Available in Job Manager"](#)
- [Section 6.5, "Job Triggers in Job Manager"](#)

## 3.3 Modifying Oracle BI Scheduler Jobs in the Job Manager

You can modify an Oracle BI Scheduler job in the Job Manager using the following procedure.

---

**Note:** If, while adding or modifying a job in the Job Manager, you enter a script in the Script field, then the Oracle BI Scheduler creates a file with an SCS extension in the following directory:

```
\oraInst\bifoundation\\coreapplication_obisch1\scripts\scheduler
```

Where oraInst is the install location for Oracle Business Intelligence.

Oracle BI Scheduler's job scripts are stored in this location (not in the back-end database), so do not remove scripts from here.

---

**To modify an Oracle BI Scheduler job in the Job Manager:**

1. In the Job Manager, select the job to modify.
2. From the **Jobs** menu, select **Modify Job**.
3. In the Modify Job dialog box, change the job properties.

For field descriptions, see [Section 6.3, "General Oracle BI Scheduler Job Properties,"](#) [Section 6.4, "Job Action Properties Available in Job Manager,"](#) and [Section 6.5, "Job Triggers in Job Manager."](#)

## 3.4 About Adding Agent Jobs

You cannot add agent jobs using the Job Manager. Agents are most commonly added through Oracle BI Presentation Services. For information, see [Section 1.1, "What is Oracle BI Scheduler?"](#)

You can, however, modify an agent job using the Modify Job dialog box. For information, see [Section 3.5, "Modifying Agents in Job Manager."](#)

## 3.5 Modifying Agents in Job Manager

You can modify an individual agent using the Modify Job dialog box in the Job Manager.

### To modify an agent in Job Manager:

1. In Job Manager, select the agent to modify.
2. From the **Jobs** menu, select **Modify Job**.
3. In the Modify Job dialog box, change the agent job properties.

For field descriptions, see [Section 6.3, "General Oracle BI Scheduler Job Properties,"](#) [Section 6.4, "Job Action Properties Available in Job Manager,"](#) and [Section 6.5, "Job Triggers in Job Manager."](#)

Agent-specific job properties are described in the following table.

---



---

**Note:** A default value in these fields indicates that the value that is specified in "Agent Scheduler Configuration Settings" *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition* is active.

---



---

Agent Property	Description
BI Presentation Server	Specifies the Web server that this agent contacts when it runs. Do not change this setting, because the agent might not exist on a different Web server. This feature was added for debugging purposes only.
Debug Log	Determines whether debugging information is written to a log. This overrides the Scheduler's Debug configuration setting for just this one agent. If set to true, then log files named Agent- <i>&lt;jobID&gt;</i> - <i>&lt;InstanceID&gt;</i> .log are created in the active Scheduler's log directory when the agent is executed.

## 3.6 Re-Running a Job Instance

You can re-run any type of job with failed job instances to deliver results to recipients which haven't received the agent (for example, agent jobs with status Failed, Warning, Cancelled, Timed Out)). When you re-run a failed agent job, only the failed items are delivered. For example, if there are 1,000 recipients of an agent, but 200 of those recipients belonged to a group that does not have the appropriate request viewing permissions, then the administrator can correct the permission settings and re-run the job. In this example, only the 200 users in the modified group receive the agent. The agent is not re-delivered to the other 800 users who successfully received the agent.

### To re-run a job instance:

1. Open the Oracle BI Administration Tool in Online mode.
2. From the **Manage** menu, select **Jobs** from the toolbar.  
The Job Manager window is displayed.
3. Go to the Instance list.
4. Locate the job instance to re-run.
5. Right-click the job instance and select **Re-run Instance(s) Now**.

The **Re-run Instance(s) Now** menu option is not available for successfully completed jobs.

## 3.7 Managing Oracle BI Scheduler Job Instances

An *instance* in the Oracle BI Scheduler is a record that stores information regarding a specific execution of an Oracle BI Scheduler job.

You can perform the following tasks using Oracle BI Scheduler job instances:

- [Section 3.7.1, "Viewing or Refreshing Oracle BI Scheduler Job Instances"](#)
- [Section 3.7.2, "Cancelling Oracle BI Scheduler Job Instances"](#)
- [Section 3.7.3, "Purging Oracle BI Scheduler Job Instances"](#)

**To work with Oracle BI Scheduler job instances:**

1. Click the Instances tab in the lower-left corner of the Job Manager window.
2. When instances are present, use the tree in the left pane to locate instances and view information about them.
  - The Instances menu is described in the table in [Section 6.2.4, "Instances Menu in Job Manager."](#)
  - Instances properties are described in the table in [Section 6.2.5, "Instance Properties in Job Manager."](#)

### 3.7.1 Viewing or Refreshing Oracle BI Scheduler Job Instances

You can view Oracle BI Scheduler job instance information using the following procedure.

---

---

**Note:** In some environments, if numerous instances have run and instances have not been purged in some time, then this process can take a few seconds.

---

---

**To view Oracle BI Scheduler job instance information:**

1. Open the Oracle BI Administration Tool in Online mode.
2. From the **Manage** menu, select **Jobs** from the toolbar.

The Job Manager window is displayed.
3. Go to the Instance list.
4. Select a particular job instance and from the **Instances** menu, select **View Instance**.

A description of the Instance properties shown in the Instance window is given in [Section 6.2.4, "Instances Menu in Job Manager."](#)

**To refresh Oracle BI Scheduler job instances:**

1. In the Instance List, from the **Instances** menu, select **Refresh Instance List**.

### 3.7.2 Cancelling Oracle BI Scheduler Job Instances

Registered canceled instances are described in [Section 4.8.6, "RegisterCancelCommand Method."](#) The cancel event is issued to the Oracle BI Scheduler and the instance is marked as canceled when its registered cancel methods are called.

**To cancel an Oracle BI Scheduler job instance:**

1. Open the Oracle BI Administration Tool in Online mode.

2. From the **Manage** menu, select **Jobs** from the toolbar.  
The Job Manager window is displayed.
3. Go to the Instance list.
4. Select a particular job instance, and from the **Instances** menu, select **Cancel Instance(s)**.

### 3.7.3 Purging Oracle BI Scheduler Job Instances

Purging a job instance involves removing it from the back-end database using one of the following methods:

- [Section 3.7.3.1, "Using the Job Manager Instances List"](#)
- [Section 3.7.3.2, "Using the Job Manager Purge Instances Window"](#)

#### 3.7.3.1 Using the Job Manager Instances List

The following procedure purges Oracle BI Scheduler job instances through the Instances List.

**To purge Oracle BI Scheduler job instances through the Instances List:**

1. Open the Oracle BI Administration Tool in online mode.
2. From the **Manage** menu, select **Jobs** from the toolbar.  
The Job Manager window is displayed.
3. Go to the Instance list.
4. Select the instances from the Instance List and click **Delete**.

#### 3.7.3.2 Using the Job Manager Purge Instances Window

The following procedure purges job instances through the Purge Instances window.

**To purge Oracle BI Scheduler job instances through the Purge Instances window:**

1. Open the Oracle BI Administration Tool in online mode.
2. From the **Manage** menu, select **Jobs** from the toolbar.  
The Job Manager window is displayed.
3. Go to the Instance list.
4. Click the Purge Instance(s) icon on the toolbar or from the **Instances** menu, select **Purge Instances** to open the Purge Instances window.

You can purge instances by JobID, by UserID, or by End Time.

If you select the End Time method, then all jobs with an End Time less than or equal to the given time are purged.

5. Select the purge method to use.
6. Click **OK** when you have finished to return to the Job Manager window.



---

---

## Programming BI Scheduler VBScript and JScript Jobs

You can use the Oracle BI Scheduler to schedule general purpose scripts that extend the functionality of Oracle Business Intelligence.

Scripts can either be standalone Script Jobs (in Job Manager), or Script Actions tagged onto the end of agents. (For more information on agents, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*). Both types of script use the same script facilities, with a few exceptions. For example Script Actions can access the result set that is being delivered by the agent, while standalone Script Jobs have no access to result sets.

Oracle BI Scheduler includes a Script object that encapsulates a running script. The Script object represents a script and exposes the properties and methods of a script. You can access its methods and properties directly because its name is implied. For example, to access the JobID property, you can specify JobID, not Script.JobID.

---

---

**Note:** Scripting for agents and scripts defined by Oracle BI Scheduler Job Manager are supported only under Windows platforms and are not supported under UNIX.

---

---

This chapter describes how to configure custom script properties, includes scripting examples, and provides detailed information about script job properties. It contains the following topics:

- [Section 4.1, "Configuring Custom Script Properties for Oracle BI Scheduler"](#)
- [Section 4.2, "Creating a Custom Script Example - Cache Clearance"](#)
- [Section 4.3, "Configuring Custom Script Properties for Oracle BI Delivers"](#)
- [Section 4.4, "Creating a Custom Script Example - Copy Results to the File System"](#)
- [Section 4.5, "Oracle BI Scheduler Read-Only Script Object Properties"](#)
- [Section 4.6, "Oracle BI Scheduler Read/Write Script Object Properties"](#)
- [Section 4.7, "Oracle BI Scheduler Script-Defined Constants"](#)
- [Section 4.8, "Oracle BI Scheduler Script Object Methods and Events"](#)
- [Section 4.9, "Troubleshooting JScript and VBScript Job Failures"](#)

## 4.1 Configuring Custom Script Properties for Oracle BI Scheduler

Use the following procedure to modify the properties of an existing Oracle BI Scheduler script.

To add this job as a standalone job in Job Manager, see [Section 3.2, "Adding Oracle BI Scheduler Jobs in Job Manager."](#)

---



---

**Note:** The script has to exist on the Oracle BI Scheduler server computer before you can configure the properties.

---



---

**To configure custom Oracle BI Scheduler script properties:**

1. Set the custom properties according to [Section 6.4, "Job Action Properties Available in Job Manager."](#)

For example, for the `purgeSASCACHE.js` script, use the values that are shown in the following table. To view an example of the `SASCACHE.js` script, see [Section 4.2, "Creating a Custom Script Example - Cache Clearance."](#)

Field	Value or Setting
Script Type	JScript
Script / Script File check boxes	Script File
Script	<code>purgeSASCACHE.js</code>
Parameter(0): User	Administrator
Parameter(1): Password	<i>your_password</i>

2. Click OK.

## 4.2 Creating a Custom Script Example - Cache Clearance

You can use the `purgeSASCACHE.js` script to periodically purge all of the cache from the Oracle BI Server. The file must be saved in the following directory:

`ORACLE_INSTANCE\bifoundation\OracleBISchedulerComponent\coreapplication_obisch\scripts\common`

```

////////////////////////////////////
//purgeSASCACHE.js
//
//Purges the cache on SAS.
//Parameter(0) - The user name to pass in to NQCMD.
//Parameter(1) - The password for the aforementioned user.
////////////////////////////////////
//The full path to nqcmd.exe
var nqCmd = "[${INSTALLDIR}]\server\Bin\nqcmd.exe";
//The data source name
var dsn = "BI Web";
//The user to execute the queries
var user = Parameter(0);
//The password of the aforementioned user
var pswd = Parameter(1);
//The ODBC procedure call for purging the cache
var sqlStatement = "{call SAPurgeAllCache()}";
////////////////////////////////////

```

```

//Returns a string from the file name
////////////////////////////////////
function GetOutput(fso, fileName)
{
    var outStream = fso.OpenTextFile(fileName, 1);
    var output = outStream.ReadAll();
    outStream.Close();
    return output;
}
////////////////////////////////////
// Get WshShell object and run nqCmd. Capture the output
// so that we can handle erroneous conditions.
var wshShell = new ActiveXObject("WScript.Shell");
// Create a temp file to input the SQL statement.
var fso = new ActiveXObject("Scripting.FileSystemObject");
var tempFolder = fso.GetSpecialFolder(2);
var tempInFileName = fso.GetTempName();
var tempOutFileName = fso.GetTempName();
tempInFileName = tempFolder + "\\ " + tempInFileName;
tempOutFileName = tempFolder + "\\ " + tempOutFileName;
var tempInFile = fso.CreateTextFile(tempInFileName, true);
tempInFile.WriteLine(sqlStatement);
tempInFile.Close();
try
{
    // execute
    var dosCmd = nqCmd + " -d \"" + dsn + "\" -u \"" + user
        + "\" -p \"" + pswd + "\" -s \"" + tempInFileName + "\" +
        " -o \"" + tempOutFileName + "\"";
    wshShell.Run(dosCmd, 0, true);
    var output = GetOutput(fso, tempOutFileName);
    // Remove the temp files
    fso.DeleteFile(tempInFileName);
    if (fso.FileExists(tempOutFileName)) {
        fso.DeleteFile(tempOutFileName);
    }

    // Check the output for any errors
    if (output.indexOf("Processed: 1 queries") == -1) {
        ExitCode = -1;
        throw Error(-1, output);
    }
    else if (output.indexOf("Encountered") != -1) {
        ExitCode = -2;
        throw Error(-2, output);
    }
    ExitCode = 0;
} catch (e) {
    if (fso.FileExists(tempInFileName)) {
        fso.DeleteFile(tempInFileName);
    }
    if (fso.FileExists(tempOutFileName)) {
        fso.DeleteFile(tempOutFileName);
    }
    throw e;
}

```

## 4.3 Configuring Custom Script Properties for Oracle BI Delivers

You set script properties on the Actions tab of an agent in Oracle BI Delivers. See *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition* for details. Refer also to [Section 3.5, "Modifying Agents in Job Manager."](#)

---



---

**Note:** The script has to exist on the Oracle BI Scheduler server computer before you can create the agent. Create the custom script, and then create the agent to call the script. See [Section 4.4, "Creating a Custom Script Example - Copy Results to the File System."](#)

---



---

### To configure custom script properties for agents:

1. On the Home page in Oracle BI EE, click the New menu and select the **Agent** option.
2. Display the Conditions tab and select the **Use a condition** box.
3. Click **Create** and **Browse** to select an analysis.
4. Click **OK**.
5. Click **OK**.
6. Display the Actions tab.
7. Click the **Add New Action** icon and select the **Invoke Server Script** menu option. The New Action - Invoke Server Script dialog is displayed.
8. Click the **Add Document Parameter** icon.
9. Select the first row of parameters and click the **Delete** button.
10. Enter properties for the parameter displayed.

For example, for the script that is shown in [Section 4.4, "Creating a Custom Script Example - Copy Results to the File System,"](#) you use the values that are described in the following table.

Field	Value or Setting
Language	JavaScript
Script Path	createResultfile.js
Name	Enter a name for the parameter.
Prompt	Enter a prompt. For example "Result Set:"
Value	Select Value from the list. This maps a fixed value, for example, PDF.
	Select Condition Analysis
Optional	Clear the check box.

11. Use the plus icon to display a new row for a second parameter.
12. Enter Result Set Extension in the **Prompt** field.
13. Enter .pdf into the **Value** field.
14. Click **OK**.
15. Save the agent.

This script runs after the Conditional Request of the agent.

## 4.4 Creating a Custom Script Example - Copy Results to the File System

This example configures a script for the Oracle BI Scheduler that copies the results of an agent to another directory. The script copies the temporary file that contains the results of the Conditional Request to the agent log directory. The JobID, InstanceID, and UserID are used in the file name to guarantee that the result sets do not overwrite each other with each execution of the agent, for each user, or for other agents that share this script.

To add this job in an agent, see [Section 4.3, "Configuring Custom Script Properties for Oracle BI Delivers."](#)

The example script uses the following values:

- The agent log directory on the Oracle BI Scheduler computer is \$OracleBI\instances\logs\Agents (where \$OracleBI is the location in which the Oracle Business Intelligence software is installed).
- The agent is run as Administrator.
- The Custom Script properties are set according to the table in [Section 4.3, "Configuring Custom Script Properties for Oracle BI Delivers."](#)
- If the job ID is 101 and instance ID is 1208, then you see a file called 101-1208-Administrator-AgentScript1.pdf.

The output of this example, after the agent is run, is a file on the Oracle BI Scheduler computer called

D:\OracleBI\Log\Agents\101-1208-Administrator-AgentScript1.PDF.

This file contains the results of the Conditional Request in PDF format.

For all script jobs from chained agents, the full path name to the temporary file is specified in Parameter(0).

```

////////////////////////////////////
//
// createResultFile.js
//
// Copies the results in the temporary file to a new file name
//
// Parameter(0) = Agent Result File Path
// Parameter(1) = Last Part of Output File Name (no path)
//
////////////////////////////////////
var FSO = new ActiveXObject("Scripting.FileSystemObject");
var fileName = GetConfigurationValue("Log Dir", "Agents") +
  "\\\" + JobID + "-" + InstanceID + "-" + UserID + "-" +
  Parameter(1);
var fooFile = FSO.CopyFile(Parameter(0), fileName, true);

```

## 4.5 Oracle BI Scheduler Read-Only Script Object Properties

The Oracle BI Scheduler supports the read-only script object properties that are described in [Table 4-1](#).

**Table 4–1 Oracle BI Scheduler Read-Only Script Object Properties**

Object Property	Description	Return Value	Syntax
JobID	Returns the job identification number that is associated with this instance.	long	NA
InstanceID	Returns the instance identification number that is associated with this instance.	long	NA
ParameterCount	Returns the number of job parameters that is associated with the job script.	long	NA
Parameter (index)	Returns a specific parameter that is associated with the script. Parameter (index) returns an error if the given index is less than zero or greater than ParameterCountminus 1.	string	Parameter(index) Index is the zero-based index of the parameter.
Script	Returns the Script object that represents the current script. This object implements the COM IDispatch interface and can be passed as arguments to methods of other objects that exist on the system. Implementing the COM IDispatch is particularly useful when handling cancel events to a running instance. See <a href="#">Section 4.8.6, "RegisterCancelCommand Method"</a> .	script object	NA
UserID	Returns the user identification number that is associated with the instance.	string	NA

## 4.6 Oracle BI Scheduler Read/Write Script Object Properties

The Oracle BI Scheduler supports the read/write script object properties that are shown in [Table 4–2](#).

**Table 4–2 Oracle BI Scheduler Read/Write Script Object Properties**

Object Property	Description	Return Value
Message	<p>Sets or returns the Message property of the running instance. The Message property can convey meaningful error information. Setting this value changes the Message field of a Job Instance without stopping execution of the current Job Script.</p> <p>If the JScript throw() method is called and this property has been set, then the value is appended to the message description in the JScript or VBScript Error object.</p> <p>COM objects that implement the IDispatch interface can be accessed from within Job Scripts. If any method fails and properly provides error information through the SetErrorInfo() method, then that information is contained in the Message field of the Job Instance. If the Message property is set before the COM object error is generated, then that string value is appended to the COM object error information.</p>	string
Severity	Sets the instance status. You can set it to any of the Severity Constants, as described in <a href="#">Section 4.7.1, "Severity Constants"</a> . By default, it is set to nqSeverityInformation.	string
ExitCode	Sets or returns the Exit Code property that is associated with the instance. The default is 0 (zero). See the description of ExitCode instance properties in <a href="#">Section 6.2.5, "Instance Properties in Job Manager"</a> .	long

## 4.7 Oracle BI Scheduler Script-Defined Constants

The Oracle BI Scheduler supports the following script-defined constants. These constants are used by the methods to schedule new jobs.

- [Section 4.7.1, "Severity Constants"](#)
- [Section 4.7.2, "DayEnum Constants"](#)
- [Section 4.7.3, "DayOfWeekEnum Constants"](#)
- [Section 4.7.4, "JobFlagsEnum Constants"](#)
- [Section 4.7.5, "MonthEnum Constants"](#)
- [Section 4.7.6, "OccurrenceEnum Constants"](#)

### 4.7.1 Severity Constants

This topic is part of [Section 4.7, "Oracle BI Scheduler Script-Defined Constants."](#)

Severity constants are used in the severity property of a Message (error message) returned by a script to determine the status of a job instance. [Table 4–3](#) describes Severity values.

**Table 4–3 Severity Constant Values**

Value	Description
nqSeverityInformation	Set the Severity property to <i>Information</i> if the Message contains only information for the job instance; that is, no error condition is reflected. The status of the instance is set to Completed. This is the default if Severity is not set.
nqSeverityWarning	Set the Severity property to <i>Warning</i> if the Message contains text that describes a non-critical failure. The instance status is set to Warning.
nqSeverityError	Set the Severity property to <i>Error</i> if the Message contains text that indicates a critical failure. The instance status is set to Failed.

### 4.7.2 DayEnum Constants

This topic is part of [Section 4.7, "Oracle BI Scheduler Script-Defined Constants."](#)

The DayEnum values are used with the scheduling functions to identify days in a month, from Day 1 to Day 31. [Table 4–4](#) describes DayEnum values.

**Table 4–4 DayEnum Constant Values**

Value	Description
nqDay1	Day 1
nqDay2	Day 2
nqDay3	Day 3
...	...
nqDay31	Day 31

### 4.7.3 DayOfWeekEnum Constants

This topic is part of [Section 4.7, "Oracle BI Scheduler Script-Defined Constants."](#)

The DayOfWeekEnum values are used with the scheduling functions to identify days in a week. [Table 4–5](#) describes DayOfWeekEnum values.

**Table 4–5 DayOfWeekEnum Constant Values**

Value	Description
nqSunday	Sunday
nqMonday	Monday
nqTuesday	Tuesday
nqWednesday	Wednesday
nqThursday	Thursday
nqFriday	Friday
nqSaturday	Saturday

#### 4.7.4 JobFlagsEnum Constants

This topic is part of [Section 4.7, "Oracle BI Scheduler Script-Defined Constants."](#)

The JobFlagsEnum values are used with the scheduling methods of the Script object to control how a job behaves. [Table 4–6](#) describes JobFlagsEnum values.

**Table 4–6 JobFlagsEnum Constant Values**

Value	Description
nqJobNoFlags	This flag indicates that the job has no special behavior.
nqJobDeleteWhenDone	This flag indicates that the job is deleted when there are no more scheduled run times.
nqJobDisabled	This flag indicates that the job is disabled. This is useful for preventing a job from running at the scheduled time or times.
nqJobHasEndDate	This flag indicates that the job has a valid end date.
nqJobExecuteWhenMissed	If for some reason the Oracle BI Scheduler is down when the job is supposed to start, then this flag indicates that the job should run when the Oracle BI Scheduler starts again.
nqJobDeleteScriptWhenDone	When a job is removed and this flag is set, the script that is associated with the job is deleted. This is useful only with the nqJobScriptContainsPath flag.
nqJobScriptContainsPath	This flag indicates that the script that is associated with the job contains a path to a file that contains the actual script code.
nqJobStartNow	When this flag is set, the begin date and start time are ignored. Instead, these fields are set to the current time of the Oracle BI Scheduler.

#### 4.7.5 MonthEnum Constants

This topic is part of [Section 4.7, "Oracle BI Scheduler Script-Defined Constants."](#)

The MonthEnum values are used with the scheduling functions to identify months. [Table 4–7](#) describes MonthEnum values.

**Table 4–7 MonthEnum Constant Values**

Value	Description
nqJanuary	January
nqFebruary	February
nqMarch	March
nqApril	April
nqMay	May
nqJune	June
nqJuly	July
nqAugust	August
nqSeptember	September
nqOctober	October
nqNovember	November
nqDecember	December

### 4.7.6 OccurrenceEnum Constants

This topic is part of [Section 4.7, "Oracle BI Scheduler Script-Defined Constants."](#)

The OccurrenceEnum values are used with the scheduling functions to identify the occurrence of a given day. [Table 4–8](#) describes OccurrenceEnum values.

**Table 4–8 OccurrenceEnum Constant Values**

Value	Description
nqFirst	First occurrence
nqSecond	Second occurrence
nqThird	Third occurrence
nqFourth	Fourth occurrence
nqLast	Last occurrence

## 4.8 Oracle BI Scheduler Script Object Methods and Events

You use script object methods and events for the Oracle BI Scheduler when writing programs, as described in [Chapter 4, "Programming BI Scheduler VBScript and JScript Jobs."](#) The following sections describe methods and events:

- [Section 4.8.1, "CreateArray Method"](#)
- [Section 4.8.2, "DeregisterCancelCommand Method"](#)
- [Section 4.8.3, "GetConfigurationValue Method"](#)
- [Section 4.8.4, "GetTempFileName Method"](#)
- [Section 4.8.5, "LaunchProcess Method"](#)
- [Section 4.8.6, "RegisterCancelCommand Method"](#)
- [Section 4.8.7, "ScheduleJobDaily Method"](#)
- [Section 4.8.8, "ScheduleJobMonthlyDate Method"](#)

- [Section 4.8.9, "ScheduleJobMonthlyDOW Method"](#)
- [Section 4.8.10, "ScheduleJobNow Method"](#)
- [Section 4.8.11, "ScheduleJobOnce Method"](#)
- [Section 4.8.12, "ScheduleJobWeekly Method"](#)
- [Section 4.8.13, "OnError Event"](#)

## 4.8.1 CreateArray Method

This topic is part of [Section 4.8, "Oracle BI Scheduler Script Object Methods and Events."](#)

Creates an Array object.

**Usage:** This method is provided only for JScript because local JScript Array objects cannot be passed directly to the Script methods. This method is called to create an array object and to pass the array object to Script methods that accept an array as an argument.

**Syntax 1:** Set array = CreateArray ()

**Syntax 2:** Set array = CreateArray (size)

**Syntax 3:** Set array = CreateArray (element 0, element 1, ..., element *n*)

The different syntax versions create arrays as follows:

- Syntax 1 creates an array of size 0 (zero).
- Syntax 2 creates an array with the specified size.
- Syntax 3 creates an array filled with the specified elements.

### **Example 4–1 Example**

```
var i;
var array1= CreateArray(2);
for (i = 0; i < array1.Size; i++)
{
    array1(i) = i;
}

    array1.Resize(4);
for (i = 2; i < array1.Size; i++)
{
    array1(i) = i;
}

var array2 = CreateArray(0, 1, 2,3);
for (i = 0; i < array2.Size; i++)
{
    if (array1(i) != array2(i))
        break;
}
}
```

**Arguments:** See [Table 4–9](#) for CreateArray method arguments.

**Return Value:** Returns an Array object.

**Table 4–9 CreateArray Method Arguments**

Argument	Description
size	A long value that specifies the initial size of the array.
element0 ... elementn	The values to place in the array. This creates an array with the lower and upper bounds of 0 (zero) and n, respectively.

## 4.8.2 DeregisterCancelCommand Method

This topic is part of [Section 4.8, "Oracle BI Scheduler Script Object Methods and Events."](#)

Deregisters a previously registered cancel method.

**Usage:** Call this method to deregister the most recently registered cancel method after a long operation has completed successfully. You need not call this method if the script was canceled.

**Syntax:** DeregisterCancelCommand

## 4.8.3 GetConfigurationValue Method

This topic is part of [Section 4.8, "Oracle BI Scheduler Script Object Methods and Events."](#)

Returns the value in the configuration relative to the root registry entry of the Oracle BI Scheduler.

**Usage:** Returns the string value for a registry setting relative to the Oracle BI Scheduler. The `configKey` and `subkeyPath` strings must be identical to those in the registry.

**Syntax:** value = GetConfigurationValue(configKey [, subkeyPath])

**Arguments:** See [Table 4–10](#) for GetConfigurationValue method arguments.

**Return Value:** Returns a string value.

**Table 4–10 GetConfigurationValue Method Arguments**

Argument	Description
configKey	A string that specifies the registry key name to return.
subkeyPath	(Optional) A string value that specifies the registry path below the Oracle BI Scheduler's root path.

## 4.8.4 GetTempFileName Method

This topic is part of [Section 4.8, "Oracle BI Scheduler Script Object Methods and Events."](#)

Returns a temporary file name.

**Usage:** GetTempFileName() does not create a file. It only provides a temporary file name for use in creating a file. Files that are created in job scripts are not deleted automatically when the script terminates.

**Syntax:** tfname = GetTempFileName()

**Return Value:** Returns a string value.

## 4.8.5 LaunchProcess Method

This topic is part of [Section 4.8, "Oracle BI Scheduler Script Object Methods and Events."](#)

Executes a command line in a new process.

**Usage:** Call this method to execute a command line in a new process. If the wait argument is set to True, then this method returns the exit code that is returned by the process.

**Syntax:** `exitcode = LaunchProcess ( commandLine [, wait, terminateOnCancel] )`

**Arguments:** See [Table 4–11](#) for LaunchProcess method arguments.

**Return Value:** Returns a long value.

**Table 4–11 LaunchProcess Method Arguments**

Argument	Description
commandLine	A string that specifies the command line to execute.
wait	(Optional) A Boolean value that specifies whether the method should wait for the process to terminate. The default is True.
terminateOnCancel	(Optional) A Boolean value that specifies whether the method should terminate the process when the script is canceled. The default is True.

## 4.8.6 RegisterCancelCommand Method

This topic is part of [Section 4.8, "Oracle BI Scheduler Script Object Methods and Events."](#)

Registers a method to be called when the script is canceled.

**Usage:** Occasionally, an object's method takes a long time to complete. If the job is canceled before the call returns, then the script engine still must wait until the call returns. This could potentially take hours and limit resources. This method solves the problem by registering a method that is asynchronously called by the script engine if the script gets canceled.

Cancel methods should be registered before calling the method that executes a long operation. When the method returns, the cancel method should be deregistered by calling `DeregisterCancelCommand()`.

A good practice is to hide implementation details of a COM object from the caller, having the COM object itself handle all registration and deregistration of cancel commands. Pass an instance of the Script object to the COM object, then call the `RegisterCancelCommand()` and `DeregisterCancelCommand()` methods because the Script object implements the `IDispatch` interface.

**Syntax:** `RegisterCancelCommand source, methodName [, arguments]...`

**Arguments:** See [Table 4–12](#) for RegisterCancelCommand method arguments.

**Table 4–12 RegisterCancelCommand Method Arguments**

Argument	Description
source	An object whose method is being registered.
methodName	A string that specifies the method name.

**Table 4–12 (Cont.) RegisterCancelCommand Method Arguments**

Argument	Description
arguments	Optional arguments to be passed into the method.

### 4.8.7 ScheduleJobDaily Method

This topic is part of [Section 4.8, "Oracle BI Scheduler Script Object Methods and Events."](#)

Schedules a new job with a Daily trigger.

**Syntax:** ScheduleJobDaily name, description, scriptType, script, startDate, startTime, endTime, minutesInterval, daysInterval [, parameters, flags, maxRunTimeMS, maxConcurrentInstances, endDate]

**Arguments:** See [Table 4–13](#) for ScheduleJobDaily method arguments.

**Table 4–13 ScheduleJobDaily Method Arguments**

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type that is associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) that is associated with the job.
startDate	A date value that specifies the date that the job is activated.
startTime	A date value that specifies the time that the job is activated.
endTime	A date value that specifies the time that the job is deactivated.
minutesInterval	A long value that specifies the number of minutes between consecutive job executions.
daysInterval	An integer value that specifies the number of days between job invocations.
parameters	(Optional) A string array of parameter values that is passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags that are associated with the job. For valid settings, see <a href="#">Section 4.7.4, "JobFlagsEnum Constants"</a> . The default is nqJobNoFlags.
maxRunTimeMS	(Optional) A long value that specifies the maximum time in milliseconds that a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.
maxConcurrentInstances	(Optional) A long value that specifies the maximum number of concurrent running instances of this job. The default is 0 (zero), which means no limit.
endDate	(Optional) A date value that specifies the time that the job is deactivated.

### 4.8.8 ScheduleJobMonthlyDate Method

This topic is part of [Section 4.8, "Oracle BI Scheduler Script Object Methods and Events."](#)

Schedules a new job with a Monthly by Date trigger.

**Syntax:** ScheduleJobMonthlyDate name, description, scriptType, script, startDate, startTime, endTime, minutesInterval, whichDays, whichMonths [, parameters, flags, maxRunTimeMS, maxConcurrentInstances, endDate]

**Arguments:** See [Table 4–14](#) for ScheduleJobMonthlyDate method arguments.

**Table 4–14 ScheduleJobMonthlyDate Method Arguments**

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type that is associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) that is associated with the job.
startDate	A date value that specifies the date that the job is activated.
startTime	A date value that specifies the time that the job is activated.
endTime	A date value that specifies the time that the job is deactivated.
minutesInterval	A long value that specifies the number of minutes between consecutive job executions.
whichDays	An long value that specifies the days of the month on which the job runs. For valid settings, see <a href="#">Section 4.7.2, "DayEnum Constants."</a>
whichMonths	An integer value that specifies the months in which the job runs. For valid settings, see <a href="#">Section 4.7.5, "MonthEnum Constants."</a>
parameters	(Optional) A string array of parameter values that is passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags that are associated with the job. For valid settings, see <a href="#">Section 4.7.4, "JobFlagsEnum Constants."</a> The default is nqJobNoFlags.
maxRunTimeMS	(Optional) A long value that specifies the maximum time in milliseconds that a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.
maxConcurrentInstances	(Optional) A long value that specifies the maximum number of concurrent running instances of this job. The default is 0 (zero), which means no limit.
endDate	(Optional) A date value that specifies the time that the job is deactivated.

### 4.8.9 ScheduleJobMonthlyDOW Method

This topic is part of [Section 4.8, "Oracle BI Scheduler Script Object Methods and Events."](#)

Schedules a new job with a monthly by day of the week (DOW) trigger.

**Syntax:** ScheduleJobMonthlyDOW name, description, scriptType, script, startDate, startTime, endTime, minutesInterval, whichOccurrences, whichDays, whichMonths [, parameters, flags, maxRunTimeMS, maxConcurrentInstances, endDate]

**Arguments:** See [Table 4–15](#) for ScheduleJobMonthlyDOW method arguments.

**Table 4–15** *ScheduleJobMonthlyDOW Method Arguments*

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type that is associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) that is associated with the job.
startDate	A date value that specifies the date that the job is activated.
startTime	A date value that specifies the time that the job is activated.
endTime	A date value that specifies the time that the job is deactivated.
minutesInterval	A long value that specifies the number of minutes between consecutive job executions.
whichOccurrences	An integer value that specifies the occurrences of days of the week on which the job runs. For valid settings, see <a href="#">Section 4.7.2, "DayEnum Constants"</a> .
whichDays	An integer value that specifies the days of the week on which the job runs. For valid settings, see <a href="#">Section 4.7.3, "DayOfWeekEnum Constants"</a> .
whichMonths	An integer value that specifies the months in which the job runs. For valid settings, see <a href="#">Section 4.7.5, "MonthEnum Constants"</a> .
parameters	(Optional) A string array of parameter values that is passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags that are associated with the job. For valid settings, see <a href="#">Section 4.7.4, "JobFlagsEnum Constants"</a> . The default is nqJobNoFlags.
maxRunTimeMS	(Optional) A long value that specifies the maximum time in milliseconds that a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.
maxConcurrentInstances	(Optional) A long value that specifies the maximum number of concurrent running instances of this job. The default is 0 (zero), which means no limit.
endDate	(Optional) A date value that specifies the time that the job is deactivated.

#### 4.8.10 ScheduleJobNow Method

This topic is part of [Section 4.8, "Oracle BI Scheduler Script Object Methods and Events."](#)

Schedules a new job with a Run Now trigger.

**Syntax:** ScheduleJobNow name, description, scriptType, script [, parameters, flags, maxRunTimeMS]

**Arguments:** See [Table 4–16](#) for ScheduleJobNow method arguments.

**Table 4–16** *ScheduleJobNow Method Arguments*

Argument	Description
name	A string that specifies the name of the job.

**Table 4–16 (Cont.) ScheduleJobNow Method Arguments**

Argument	Description
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type that is associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) that is associated with the job.
parameters	(Optional) A string array of parameter values that is passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags that are associated with the job. For valid settings, see <a href="#">Section 4.7.4, "JobFlagsEnum Constants"</a> . The default is nqJobNoFlags.
maxRunTimeMS	(Optional) A long value that specifies the maximum time in milliseconds that a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.

### 4.8.11 ScheduleJobOnce Method

This topic is part of [Section 4.8, "Oracle BI Scheduler Script Object Methods and Events."](#)

Schedules a new job with a Run Once trigger.

**Syntax:** ScheduleJobOnce name, description, scriptType, script, startDate, startTime [, parameters, flags, maxRunTimeMS]

**Arguments:** See [Table 4–17](#) for ScheduleJobOnce method arguments.

**Table 4–17 ScheduleJobOnce Method Arguments**

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type that is associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) that is associated with the job.
startDate	A date value that specifies the date that the job is activated.
startTime	A date value that specifies the time that the job is activated.
parameters	(Optional) A string array of parameter values that is passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags that are associated with the job. For valid settings, see <a href="#">Section 4.7.4, "JobFlagsEnum Constants"</a> . The default is nqJobNoFlag.
maxRunTimeMS	(Optional) A long value that specifies the maximum time in milliseconds that a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.

### 4.8.12 ScheduleJobWeekly Method

This topic is part of [Section 4.8, "Oracle BI Scheduler Script Object Methods and Events."](#)

Schedules a new job with a Weekly trigger.

**Syntax:** ScheduleJobWeekly name, description, scriptType, script, startDate, startTime, endTime, minutesInterval, weeksInterval, whichDays [, parameters, flags, maxRunTimeMS, maxConcurrentInstances, endDate]

**Argument:** See [Table 4–18](#) for ScheduleJobWeekly method arguments.

**Table 4–18 ScheduleJobWeekly Method Arguments**

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type that is associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) that is associated with the job.
startDate	A date value that specifies the date that the job is activated.
startTime	A date value that specifies the time that the job is activated.
endTime	A date value that specifies the time that the job is deactivated.
minutesInterval	A long value that specifies the number of minutes between consecutive job executions.
weeksInterval	An integer value that specifies the number of weeks between job invocations.
whichDays	An integer value that specifies the days of the week on which the job runs. See <a href="#">Section 4.7.3, "DayOfWeekEnum Constants"</a> for valid settings.
parameters	(Optional) A string array of parameter values that is passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags that are associated with the job. For valid settings, see <a href="#">Section 4.7.4, "JobFlagsEnum Constants"</a> . The default is nqJobNoFlags.
maxRunTimeMS	(Optional) A long value that specifies the maximum time in milliseconds that a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.
maxConcurrentInstances	(Optional) A long value that specifies the maximum number of concurrent running instances of this job. The default is 0 (zero), which means no limit.
endDate	(Optional) A date value that specifies the time that the job is deactivated.

### 4.8.13 OnError Event

This topic is part of [Section 4.8, "Oracle BI Scheduler Script Object Methods and Events."](#)

Occurs when the script engine encounters a run-time error while executing the script. This is intended for cleanup purposes, but the creative use of try/catch blocks in JScript and appropriate Error Handling in VBScript are often superior alternatives to using this event.

**Usage:** The script engine calls this procedure when it encounters a run-time error while executing the script. Define this procedure in your script to perform cleanup

activities before the script terminates, such as deleting temporary files and releasing resources.

**Syntax:** OnError

**Example 4–2 Using VBScript:**

```
Public Sub OnError()
    LogFile.WriteLine "Encountered a runtime error in the script."
    LogFile.Close
End Sub
```

**Example 4–3 Using JScript:**

```
function OnError()
{
    LogFile.WriteLine("Encountered a runtime error in the
    script.");
    LogFile.Close();
}
```

## 4.9 Troubleshooting JScript and VBScript Job Failures

If a JScript or VBScript job fails with the error "nQSError: 66001] Failed to create the ActiveX scripting engine.", then the required script engine (VBScript or JScript) is not available or is broken on this computer.

You can reregister the script DLL files in Windows by running the command "regsvr32 vbscript.dll" for a VBScript failure or "regsvr32 jscript.dll" for a JScript failure. If these return the message "failed module could not be found", then you must repair the Windows installation to re-instate the missing DLL files. You can achieve this by rolling back to a restore point, or by carrying out a repair using the original operating system installation discs.

---

## Programming BI Scheduler Java Jobs

You can use the Oracle BI Scheduler to schedule Java programs, called Java jobs, that extend the functionality of Oracle Business Intelligence. Java jobs can be either standalone Scheduler Jobs in Job Manager (Windows), or existing Java Actions in Oracle BI Delivers that are added to the end of agents and upgraded from Release 10g (Windows or UNIX).

Note that existing Java actions that are upgraded from Release 10g can be executed, but not created in Release 11g. Instead, you should use the EJB-based Java actions that are available in this release. For more information on actions, see *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition*.

This chapter describes programming Java jobs for the Oracle BI Scheduler and contains the following topics:

- [Section 5.1, "Using Oracle BI Scheduler Java Jobs"](#)
- [Section 5.2, "Oracle BI Scheduler Java Jobs"](#)
- [Section 5.3, "Adding Java Jobs for Oracle BI Scheduler"](#)
- [Section 5.4, "Oracle BI Scheduler Custom Java Program Package"](#)
- [Section 5.5, "SchedulerJavaExtension Interface"](#)
- [Section 5.6, "SchedulerJobInfo Interface"](#)
- [Section 5.7, "SchedulerJobException Class"](#)
- [Section 5.8, "Oracle BI Scheduler Java Extension Example"](#)

### 5.1 Using Oracle BI Scheduler Java Jobs

Java jobs are Java programs that are executed by the JavaHost process on behalf of the Oracle BI Scheduler. Java jobs are different than Java EJB actions. A Java job is deployed in a JAR file, with the entry point defined by one class. That class must implement the SchedulerJavaExtension interface. The job's context is provided by the input SchedulerJobInfo parameter.

This section contains the following information:

- [Section 5.1.1, "Adding and Configuring Custom Java Jobs in Job Manager"](#)
- [Section 5.1.2, "Example: Creating a Java Program for Agents"](#)
- [Section 5.1.3, "Example: Configuring a Java Program"](#)

## 5.1.1 Adding and Configuring Custom Java Jobs in Job Manager

You add and configure custom Java jobs in the Modify Job and Add Jobs dialog boxes in Job Manager. Refer to [Section 3.2, "Adding Oracle BI Scheduler Jobs in Job Manager"](#), and [Section 3.3, "Modifying Oracle BI Scheduler Jobs in the Job Manager"](#).

---

**Note:** The Java program must exist on the Oracle BI Scheduler server computer before you can create the job in Job Manager. Create the Java program, and then create the job to call the Java program.

---

### To configure and add custom Java jobs in Job Manager:

1. Set the custom properties according to the descriptions in [Section 6.4, "Job Action Properties Available in Job Manager"](#).
2. In the Add New Job window, enter the properties.

For example, for the Java program filecopy.jar, use the values that are shown in the following table. To view an example of the filecopy.jar program, see [Section 5.1.2, "Example: Creating a Java Program for Agents"](#).

Field	Value or Setting
Script Type	Java
Class Name	sched.sched The Java class that you created in <a href="#">Section 5.1.2, "Example: Creating a Java Program for Agents."</a>
Class Path	filecopy.jar The JAR file that contains the Java class.
Parameters	c:\tmp\report.pdf <b>Note:</b> The owner of the JavaHost process must have write permissions on this directory point.

3. Click **OK**.

The Java program is run after the Conditional Request of the agent is run.

You cannot add a new Java job action to an agent in Oracle BI Delivers. You can use only existing ones that have been upgraded to this release. However, you can add a new Java job to a job in Job Manager. For information, see [Section 3.2, "Adding Oracle BI Scheduler Jobs in Job Manager."](#)

## 5.1.2 Example: Creating a Java Program for Agents

This example creates a Java program that copies the results of an agent to another directory. The example creates a Java class that contains filecopy logic.

### To create a Java program to be used with agents:

1. Create a Java program using a Java editor.
  - a. Create a new Java class called 'sched'.
  - b. Paste the following code into the Java editor:

```
package sched;
import java.io.*;
import java.lang.Thread;
```

```

import
com.siebel.analytics.scheduler.javahostpccalls.SchedulerJavaExtension;
import
com.siebel.analytics.scheduler.javahostpccalls.SchedulerJobException;
import
com.siebel.analytics.scheduler.javahostpccalls.SchedulerJobInfo;

public class sched implements SchedulerJavaExtension{
public void run(SchedulerJobInfo jobInfo) throws SchedulerJobException
{
    System.out.println("JobID is:" + jobInfo.jobID());
    System.out.println("Instance ID is:" + jobInfo.instanceID());
    System.out.println("JobInfo to string is:" + jobInfo.toString());
    try
    {
        // File outputFile = new File("D:\\JavaJob.txt");
        File attachFile = jobInfo.getResultSetFile();

        InputStream in = new FileInputStream(attachFile.getAbsolutePath());
        OutputStream out = new FileOutputStream(jobInfo.parameter(0));
        byte[] buf = new byte[1024];
        int len;
        while ((len = in.read(buf)) > 0)
        {
            out.write(buf, 0, len);
        }
        in.close();
        out.close();

    }
    catch(Exception ex)
    {
        throw new SchedulerJobException(1, 1, ex.getMessage());
    }
    }
    public void cancel()
    {
    }
}

```

- c. Add the schedulerrpccalls.jar file from the `\MW_HOME\ORACLE_HOME\bifoundation\javahost\lib\scheduler` directory into your classpath.
- d. Compile the Java Class without errors.
- e. Jar the compiled output to a file. For example, filecopy.jar.
- f. Note the location of the file and ensure that there are no errors.

### 5.1.3 Example: Configuring a Java Program

This example configures the Java program that you created in [Section 5.1.2, "Example: Creating a Java Program for Agents"](#) to enable the Java job to work with agents.

**To configure a Java program to be used with agents:**

1. Copy the filecopy.jar file that you created in [Section 5.1.2, "Example: Creating a Java Program for Agents"](#) to the following directory:

```
ORACLE_HOME\bifoundation\javahost\lib
```

2. Make the following changes to the JavaHost configuration file, which is called config.xml:

```
<Scheduler>
  <Enabled>True</Enabled>
  <DefaultUserJarFilePath>D:\<ORACLE_
HOME>\bifoundation\javahost\lib</DefaultUserJarFilePath>
</Scheduler>
```

See "Using the Javahost Service for Oracle BI Presentation Services" *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition* for information on working with the JavaHost configuration file.

If the JavaHost file is not configured correctly, then the agent log file can stop getting written to, although the agent and the Scheduler are still running. In this situation, you stop the Scheduler using the Windows Task Manager.

3. Restart the JavaHost service.

## 5.2 Oracle BI Scheduler Java Jobs

The Oracle BI Scheduler integrates with the JavaHost Service to support a custom Java program. The Oracle BI Scheduler provides two Java interfaces (SchedulerJavaExtension and SchedulerJobInfo) and one Java class (SchedulerJobException). You provide a class that implements the SchedulerJavaExtension interface.

---



---

**Note:** For more information about the JavaHost service, see, "Using the Javahost Service for Oracle BI Presentation Services" in *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

---



---

## 5.3 Adding Java Jobs for Oracle BI Scheduler

Use the following procedure to add a Java job for the Oracle BI Scheduler.

---



---

**Note:** The compiled Java class file has to exist on the JavaHost computer before you can configure the properties.

---



---

### To add a Java Job for Oracle BI Scheduler:

1. Access the Job Manager and from the Jobs menu, select **Add New Job**.  
The Add New job window appears.
2. In the Script Type field, select Java.
3. Specify the custom properties. For information about setting these values, see [Section 6.4, "Job Action Properties Available in Job Manager"](#).

Example values and settings for a Java job with the class name "sample.Test", file path "Sample", and no additional paths and parameters are included below.

Field	Value or Setting
Script Type	Java
Class Name	sample.Test

Field	Value or Setting
Class File (Jar File)	Sample

4. Click OK.

## 5.4 Oracle BI Scheduler Custom Java Program Package

The public interfaces and class for Oracle BI Scheduler Custom Java Program are packaged as *com.siebel.analytics.scheduler.javahostrpcalls*. There are two interfaces and one class, which are described in following topics:

- [Section 5.5, "SchedulerJavaExtension Interface"](#)
- [Section 5.6, "SchedulerJobInfo Interface"](#)
- [Section 5.7, "SchedulerJobException Class"](#)

## 5.5 SchedulerJavaExtension Interface

Your custom code must implement the following interface:

```
package com.siebel.analytics.scheduler.javahostrpcalls;
public interface SchedulerJavaExtension {
public void run(SchedulerJobInfo jobInfo) throws SchedulerJobException;
public void cancel();
}
```

This interface has two methods: run and cancel. The following table describes the methods:

Method	Description
run	This method is invoked by the JavaHost. It provides one SchedulerJobInfo object (described below), which contains instance-related properties such as user ID, Job ID, and Instance ID and parameters. The method is declared to throw SchedulerJobException, which is also described below.
cancel	This method is invoked if the Job instance is still running while Scheduler wants to cancel it. The cancel method is called concurrently by a different thread. Your implementation must therefore protect any data that is shared by the run and cancel methods by synchronization blocks. A typical implementation would be to set a 'cancelCalled' boolean in the cancel method implementation, and check this in any long running loops in the run implementation.

## 5.6 SchedulerJobInfo Interface

The SchedulerJobInfo interface provides information about the currently running job instance to the custom code:

```
package com.siebel.analytics.scheduler.javahostrpcalls;
import java.io.*;
public interface SchedulerJobInfo {
public final int kJavaJobInformation = 0;
public final int kJavaJobWarning = 1;
public final int kJavaJobError = 2;
int jobID();
int instanceID();
int parameterCount();
String parameter(int index);
}
```

```

boolean hasResultSet();
File getResultSetFile();
String userID();
int getExitCode();
void setExitCode(int exitCode);
int getStatus();
void setStatus(int status);
String getMessage();
void setMessage(String message);
void appendMessage(String message);
}

```

Three public final integers, *kJavaJobInformation*, *kJavaJobWarning*, and *kJavaJobError* are the suggested values that are used to set the status depending upon the circumstances. The following table describes the circumstances:

Members	Description
public final int kJavaJobInformation = 0	Contains an informational message.
public final int kJavaJobWarning = 1	Contains a warning message.
public final int kJavaJobError = 2	Contains an error message.

The following table describes all the methods that are declared in the interface:

Method	Description
int jobID()	Returns the job ID that is associated with the agent.
int instanceID()	Returns the instance ID that is associated with the agent.
int parameterCount()	Returns how many parameters are associated with the agent.
String parameter(int index)	Returns the indexed parameter for the agent.(1).
boolean hasResultSet()	Specifies if there is a result set for this agent.
File getResultSetFile()	Returns a file of result set for this agent (2).
String userID()	Returns the ID of the user who is running the agent.
int getExitCode()	Returns the exit code for the agent.
void setExitCode(int exitCode)	User can set the exit code for the agent.
int getStatus()	Returns the status code for the agent.
void setStatus(int status)	User can set the status code for the agent.
String getMessage()	Returns the message that is associated with the agent.
void setMessage(String message)	User can set the message that is associated with the agent. It replaces the existing message.
void appendMessage(String message)	User can append an additional message to the agent.

## 5.7 SchedulerJobException Class

If your custom code cannot complete successfully, then throw an instance of this exception class.

```

package com.siebel.analytics.scheduler.javahostrpcalls;
public final class SchedulerJobException extends Exception {

```

```

public SchedulerJobException(int exitCode, int status, String message) {
    m_exitCode = exitCode;
    m_status = status;
    m_message = message;
}
public int getExitCode() {
    return m_exitCode;
}
public int getStatus() {
    return m_status;
}
public String getMessage() {
    return m_message;
}
private int m_exitCode;
private int m_status;
private String m_message;
}

```

The *run* method of the SchedulerJavaExtension interface is declared to throw SchedulerJobException. The following table describes the three members:

Members	Description
int m_exitCode	The framework assigns this exit code to the agent.
int m_status	The framework assigns this status code to the agent.
String m_message	The framework assigns this message to the agent.

## 5.8 Oracle BI Scheduler Java Extension Example

The following example illustrates how to use the previously described interfaces and class to create a custom Java action. For more information, see [Section 5.1, "Using Oracle BI Scheduler Java Jobs."](#)

This example does not contain any long running code, so it is acceptable to do nothing in the cancel method.

When the compiled class runs, it collects the ID of the user who ran the agent, the job ID of the agent, the instance ID of the agent, and all possible parameters into an output file.

```

package sample;
import java.io.*;
import java.lang.Thread;
import com.siebel.analytics.scheduler.javahostrpcalls.SchedulerJavaExtension;
import com.siebel.analytics.scheduler.javahostrpcalls.SchedulerJobException;
import com.siebel.analytics.scheduler.javahostrpcalls.SchedulerJobInfo;
/**
 *
 * @author
 */
public class SimpleTest implements SchedulerJavaExtension
{
    public void run(SchedulerJobInfo jobInfo) throws SchedulerJobException
    {
        System.out.println("JobID is:" + jobInfo.jobID());
        System.out.println("Instance ID is:" + jobInfo.instanceID());
        System.out.println("JobInfo to string is:" + jobInfo.toString());
        try
        {
            File outputFile = new File("D:\\temp\\JavaJob.txt");

```

```
FileWriter out = new FileWriter(outputFile);
out.write("User ID:\t\t" + jobInfo.userID() + "\r\n");
out.write("Job ID:\t\t" + jobInfo.jobID() + "\r\n");
out.write("Instance ID:\t\t" + jobInfo.instanceID() + "\r\n");
out.write("Parameter Count:\t\t" + jobInfo.parameterCount() + "\r\n");
for(int i = 0; i < jobInfo.parameterCount(); ++i)
{
out.write("\tParameter ");
out.write(new Integer(i).toString());
out.write(":\t" + jobInfo.parameter(i) + "\r\n");
}
out.close();
}
catch(Exception ex)
{
throw new SchedulerJobException(1, 1, ex.getMessage());
}
}
public void cancel()
{
}
}
```

---

---

## Oracle BI Scheduler Job Manager Menus

This chapter describes the Job Manager feature of the Oracle BI Administration Tool. It is the first part of the process of configuring Oracle BI Scheduler from the Job Manager.

This chapter contains the following topics:

- [Section 6.1, "About the Job Manager"](#)
- [Section 6.2, "Toolbar Menus in Job Manager"](#)
- [Section 6.3, "General Oracle BI Scheduler Job Properties"](#)
- [Section 6.4, "Job Action Properties Available in Job Manager"](#)
- [Section 6.5, "Job Triggers in Job Manager"](#)

### 6.1 About the Job Manager

Use the Job Manager to add, remove, modify, or cancel Oracle BI Scheduler jobs. For example, you can perform the following tasks:

- Set options for an execution schedule, such as a start time, a start date, an interval between executions, and an optional end time and date.
- Add or modify jobs using the Add Job and Modify Job dialog boxes. These dialog boxes contain three types of information:
  - General job properties
  - A script area where you can specify the actions to perform
  - A trigger area where you can specify the job trigger

The trigger defines when the job is run.

### 6.2 Toolbar Menus in Job Manager

The Job Manager toolbar contains four menus, as described in the following topics:

- [Section 6.2.1, "File Menu in Job Manager"](#)
- [Section 6.2.2, "Service Management Menu in Job Manager"](#)
- [Section 6.2.3, "Jobs Menu in Job Manager"](#)
- [Section 6.2.4, "Instances Menu in Job Manager"](#)

#### 6.2.1 File Menu in Job Manager

[Table 6–1](#) describes the File menu options.

**Table 6–1 Job Manager File Menu Options**

Command	Description
Open Scheduler Connection	Opens the Machine Name dialog, which provides alternative connection mechanisms. If you run a single, non-clustered Scheduler, connect using the option Connect directly to the Active Scheduler. If the active scheduler is running on this machine (and is not configured to listen only on a particular network interface), then you might use localhost for the machine name and specify the port number (usually 9704). If the Scheduler is clustered, then use the option Connect Through Cluster Controllers. This option ensures that you can successfully connect irrespective of which Scheduler is currently the active Scheduler.  If the system has been secured with SSL, then you must select the <b>SSL</b> check box. If the default SSL configuration is used, then you can leave all other SSL fields empty.
Close Scheduler Connection	Closes the Job Manager connection to Oracle BI Scheduler.
Exit	Shuts down the Job Manager and returns you to the Oracle BI Administration Tool. If you exit the Job Manager while a connection to Oracle BI Scheduler is still open, then the connection closes.

## 6.2.2 Service Management Menu in Job Manager

Table 6–2 describes the Service Management menu options.

**Table 6–2 Job Manager Service Management Menu**

Command	Description
Pause Scheduling	Stops all jobs from executing until scheduling is continued. Pause Scheduling is sometimes required for maintenance purposes. It allows an administrator to intervene and resolve out of control jobs. A custom job that uses excessive resources and that is being scheduled very frequently, might make any other processes on that computer ineffective, including Job Manager. Pausing scheduling offers a chance to remove or modify the job.  Sometimes you might need to Pause Scheduling while Oracle BI Scheduler is stopped. In this case, scheduling continues when Oracle BI Scheduler is restarted, unless you also set the option Pause When Service Starts as described in For more information about the JavaHost service, see, "General Scheduler Configuration Settings That Affect Agents" in <i>Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition</i> .
Continue Scheduling	Resumes Oracle BI Scheduler's regular execution.
Stop Service	Stops the Oracle BI Scheduler service.

## 6.2.3 Jobs Menu in Job Manager

Table 6–3 describes the Jobs menu options.

**Table 6–3 Job Manager Jobs Menu**

Command	Description
Add New Job	Opens the Add New Job window, where you specify the properties for a new job.
Remove Job(s)	Removes the selected job or jobs from Oracle BI Scheduler. When a job is removed, all instances for that job are also removed.

**Table 6–3 (Cont.) Job Manager Jobs Menu**

Command	Description
Modify Job	Opens the Modify Job window where you can modify the properties for an existing job.
View Job	Opens the View Job window where you can view the properties for a job.
Run Job(s) Now	Immediately runs the scheduled job that you selected.
Refresh Job List	Refreshes the job information that is displayed in the Job List in the right pane.  To view the instances that are associated with one or more jobs, first highlight the jobs in the job view. Then press the refresh icon in the instances view below.

## 6.2.4 Instances Menu in Job Manager

An Oracle BI Scheduler instance records information regarding a specific execution of a job in the Oracle BI Scheduler. [Table 6–4](#) describes the Instances menu options.

**Table 6–4 Job Manager Instances Menu**

Command	Description
Cancel Instance(s)	Cancels the running job. When a job has been successfully canceled, the job's status is displayed as "Canceled."
Purge Instance	Opens the Purge Instances dialog box where you can specify the delete instance method to use. You can delete the instance by Job ID, User ID, and Before a particular time.
View Instance	Displays information about the selected instance.
Re-Run Job Instance(s) Now	Re-run a job instance. When you re-run a failed agent job, only the failed items are delivered.
Refresh Instance List	Refreshes the instance information that is displayed in the Instance List in the Job Instance View pane.

## 6.2.5 Instance Properties in Job Manager

[Table 6–5](#) describes the properties of Job Manager instances.

**Table 6–5 Job Manager Instance Properties**

Field	Description
JobID	ID of the job that is associated with this instance.
Status: Running	This is the same for agent jobs and Script jobs. If the instance is running, then the status is running.
Status: Completed	For agent: The agent instance is set to complete if deliveries are successful to all delivery devices of the agent.  For Script: This is set according to the Severity property in the script. See <a href="#">Table 4–2, "Oracle BI Scheduler Read/Write Script Object Properties"</a> .
Status: Failed	For agent: The agent instance is set to failed if deliveries are unsuccessful to any of the delivery devices of the agent.  For Script: This is set according to the Severity property in the script. See <a href="#">Table 4–2, "Oracle BI Scheduler Read/Write Script Object Properties"</a> .

**Table 6–5 (Cont.) Job Manager Instance Properties**

Field	Description
Status: Canceled	Canceling any instance from the Job Manager sets the status to canceled.
Status: Timed Out	If the job has a maximum run time and the running time of the instance exceeds this time, then the status of the instance is set to timed out.
Status: Warning	For agent: The agent instance is set to Warning if deliveries are successful to some delivery devices but not all.  For Script: This is set according to the Severity property in the script. See <a href="#">Table 4–2, "Oracle BI Scheduler Read/Write Script Object Properties"</a> .
InstanceID	Unique ID of this specific instance of the job.
Begin Time	The day and time that the Scheduler initiated the job instance.
End Time	The day and time that the job scheduler completed the job instance.
ExitCode: Agent	The ExitCode of an instance is set to the number of successful deliveries. The count corresponds to the number of successful deliveries to devices, and multiple devices might exist for each recipient of an agent.
ExitCode: Script	The ExitCode of an instance is set according to the ExitCode property in the script.  The default is 0 (zero). See <a href="#">Table 4–2, "Oracle BI Scheduler Read/Write Script Object Properties"</a>
Message	Text message that contain any error information of the instance, warnings, or general messages about the instance execution.

## 6.3 General Oracle BI Scheduler Job Properties

In the Add Job or Modify Job dialog box, use the fields to configure or modify the general properties for a job. [Table 6–6](#) describes the general job properties.

In addition, [Table 6–7](#) describes the job action properties, and [Table 6–8](#) describes the recurrent job triggers.

**Table 6–6 General Oracle BI Scheduler Job Properties**

Field	Description
Name	Short, descriptive name for the job. This field is also displayed in the Job List display in the right pane of the Job Manager window.
Description	Brief description of the job that describes its actions to end users. This field is also displayed in the Job List display in the right pane of the Job Manager window.
UserID	Required for all jobs. For jobs that communicate with the Oracle BI Server or with Oracle BI Presentation Services, the UserID must be a valid Oracle Business Intelligence user ID. This field is also displayed in the Job List display in the right pane of the Job Manager window.  When this job runs, Oracle BI Scheduler executes it on behalf of the user ID that is specified in this field.

**Table 6–6 (Cont.) General Oracle BI Scheduler Job Properties**

Field	Description
Maximum Run Time MS	<p>Specifies the maximum number of milliseconds that this job should run before it is canceled forcibly. If a job exceeds its run time, then it fails with a time-out reason code.</p> <p>To prevent the job from timing out, set this field to 0 (zero).</p> <p><b>Note:</b> One second equals 1,000 milliseconds.</p>
Last Run Time	<p>Display-only field that shows the last time that this job began execution. This field is also displayed in the Job List display in the right pane of the Job Manager window.</p> <p>If the date and time displayed here are for time zones where daylight savings applies, the time zone reflects the daylight savings time. For example, if (GMT) Greenwich Mean Time: Dublin, Edinburgh, Lisbon, London, is set during the summer months, then this means BST (British Summer Time).</p>
Next Run Time	<p>Display-only field that shows recurrent jobs and the next time this job executes. The trigger is used to determine this value.</p> <p>If the date and time displayed here is for time zones where daylight savings applies, the time zone reflects the daylight savings time. For example, if (GMT) Greenwich Mean Time: Dublin, Edinburgh, Lisbon, London, is set during the summer months, then this means BST (British Summer Time).</p>
Running Instance Count	Display-only field that shows the number of currently running instances of this job.
Delete Job When Done	When you select this option, Oracle BI Scheduler deletes the job after its last scheduled execution as defined by its trigger. When there is no next run time, the job is done. When a job is deleted, all instances are deleted as well. For most jobs, you should not select this option, because you can delete a job manually through the Job Manager.
Disabled	When you select this option, the job script does not execute when its trigger expires. However, the next run time is still updated according to the trigger settings. The Disabled field is useful when testing or debugging a new job because an administrator can quickly disable a job without losing all information.
Execute When Missed	If you select this option while Oracle BI Scheduler is stopped (either all scheduling pauses or the Oracle BI Scheduler application stops), and if the job's next run time was missed, then the job runs after Oracle BI Scheduler restarts. If you do not select this option, then the job executes at the its next run time, as defined by its trigger.
Delete Script When Job is Removed	If you select this option, then when a job is removed, its associated job script is also removed. If many jobs reference the same job script, then this option should not be set.

## 6.4 Job Action Properties Available in Job Manager

Table 6–7 describes the job action properties available in the Add Job and Modify Job dialog boxes in Job Manager. Use the fields in the Script area of the Add Job or Modify Job dialog box to define the actions a job performs when it executes.

**Table 6–7 Job Manager Job Action Fields**

Field	Description
Script Type	Oracle BI Scheduler supports VBScript, JScript, Java, and NQCmd. Set this field according to the type of script that is referenced by the Script field. The fields that are displayed depend upon the type of script that you specify.
Load Script from File	(VBScript and JScript only) In the Job Manager, you can enter either a file name or the actual contents of a script in the Script Path field.
Script	(VBScript and JScript only) This value is either a reference to a job script file or the contents of a job script itself. If it is a reference, then enter a file name in this field, such as TestConnect.js. If no path is given, then Oracle BI Scheduler examines the directory that is referred to in the Default Script Path configuration value in For more information about the JavaHost service, see, "General Scheduler Configuration Settings That Affect Agents" in <i>Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition</i> . If a path is given, then the path must be accessible by the Oracle BI Scheduler application.
Parameters	(VBScript, JScript, and Java only) Field values are passed to the job script through the Parameters array. Enter one parameter per line. For example:  c:\oracleBI\data\scheduler cli_snowflake SELECT Lastname FROM Employee
Class Name	(Script Type = Java) The fully qualified implementation class for the Java program action.
Class Path (jar file)	(Script Type = Java) The name of the JAR file that contains the Java program.
Additional Class Path(s)	(Script Type = Java) Other JAR files that contain the utility classes and libraries that the Java program must run properly. Include a comma-delimited list of JAR files. After this parameter, you must append a command line parameter as it is in Job Manager.
DSN (Data Source Name)	(Script Type = NQCmd) The data source name that Oracle BI Scheduler uses to connect to the BI Server.  <b>Note:</b> Impersonation is used with NQCmd. The value in the User ID is the user that Oracle BI Scheduler tries to impersonate when connecting to the BI Server.
SQL Input File	(Script Type = NQCmd) The fully qualified path to the SQL file that NQCmd executes. Type the full path or click the ... button to browse to the file's location. This field is typically used for files that are generated by the aggregate persistence feature.
Additional Command Line Parameters	(Script Type = NQCmd) Parameters that are passed to NQCmd. Enter one parameter per line. For example:  -o D:\foo\bar.txt

## 6.5 Job Triggers in Job Manager

A job trigger determines when and how often a job executes. Use the fields in the Trigger area of the Add Job or Modify Job dialog box to define the actions that a job performs when it executes.

There are two types of Oracle BI Scheduler Job triggers—*single-run* triggers and *recurrent* triggers.

## 6.5.1 Single-Run Triggers

Use the Trigger Type list to select the trigger type. Single-run triggers perform the action once. There are two single-run triggers:

- **Run Now.** This trigger specifies that the job runs immediately. It executes only one time.
- **Run Once.** Jobs of this trigger type execute at the date and time that is specified in the Begin Date and Start Time fields, which become active when you selected **Run Once**. An error occurs if the given time is in the past. If you select the **Set Start Time To Now** option, then this trigger is equivalent to the **Run Now** trigger.

## 6.5.2 Recurrent Triggers

All recurrent triggers specify that the job execute over a period of time at given intervals. Fields used by recurrent triggers are described in [Table 6–8](#). Recurrent Trigger Types are described in [Table 6–9](#).

**Table 6–8 Job Manager Recurrent Trigger Fields**

Field	Description
Timezone	Specifies the time zone that is used to execute the job. Displays the default time zone as specified in the instanceconfig.xml file. For information, see For more information about the JavaHost service, see, "Description of Time Zone Settings" in <i>Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition</i> .
Begin Date	Specifies the date when the first recurrent interval runs. The recurrent interval is defined as the time between Start Time and End Time. This field is hidden if you select the Set Start Time to Now option.
End Date	Specifies the date when the last recurrent interval is run. Becomes active when the Has End Date option is set. If no end date is set, then the job runs forever.
Start Time	Specifies the lower bounds of the recurrent interval. The job's first execution for a day occurs at the time that is specified in this value.
End Time	Specifies the upper bounds of the recurrent interval. The job's last execution for a given day occurs at or before the time that is specified in this value. If this value is less than the Start Time value, then the interval spans midnight of the given day. For example, a trigger with a start time of 11:00 P.M. and an End Time of 2:00 A.M. starts its execution on the date that is specified in Begin Date at 11:00 P.M. and continues until 2:00 A.M. on the following day.
Has End Date	If you select this option, then specify an End Date. If you do not select this option, then the job stays scheduled. <b>Note:</b> The schedule is perpetual; the job instance is not. If you restart Oracle BI Scheduler, then the next run time is set as dictated by the job schedule. If an instance is running while you stop Oracle BI Scheduler, then it is canceled.
Set Start Time To Now	If you select this option, then the Begin Date and Start Time fields are ignored and their values are populated with Oracle BI Scheduler's current date and time.

**Table 6–8 (Cont.) Job Manager Recurrent Trigger Fields**

Field	Description
Interval in Minutes	Specifies the number of minutes between subsequent executions of a job during the recurrent interval. A job starts execution promptly at its Start Time and executes again every $n$ minutes, where $n$ is the value of this field.
Maximum Concurrent Instances	If a job executes every $n$ minutes (from the Interval in Minutes field), then a long-running job might have overlapping executions. Use this field to set the number of concurrent running instances. For an unlimited number of concurrent instances, set this value to zero.

### 6.5.2.1 Recurrent Trigger Types

The recurrent trigger types that are available from the Trigger Type list are described in [Table 6–9](#). The fields pertain to all recurrent triggers. Depending on the trigger type that you select, additional options become active. The examples in the table illustrate how these additional options can be used.

**Table 6–9 Job Manager Recurrent Trigger Types**

Trigger Type	Description and Example
Daily	<p>Runs a job every day or every few days. The Days Interval field specifies the number of days between each subsequent recurrent interval.</p> <p>For example:</p> <p>To run a job every hour between 8:00 A.M. and 5:00 P.M. starting on January 1, 2010 and ending on January 15, 2010, set the Begin Date to 1/1/10, the Start Time to 8:00 A.M., and the End Time to 5:00 P.M. Set the Has End Date flag, the End Date to 1/15/10, the Interval in Minutes to 60, and the Days Interval to 1.</p> <p>To run a job every five minutes forever, set the Begin Date to the desired date, the Start Time to 12:00 P.M., the end time to 11:59 A.M., the Interval In Minutes to 5, and the Days Interval to 1.</p>
Weekly	<p>Runs a job on specified days of the week. The Weeks Interval specifies the number of weeks between each execution. The Days of the Week field specifies on which days the execution occurs.</p> <p>For example:</p> <p>To run a job at noon every other week on Mondays, Wednesdays, and Fridays, set the Begin Date to the desired date, the Start Time and End Time to 12:00 P.M., the Interval in Minutes to 1, the Weeks Interval to 2, and the Days of the Week to Monday, Wednesday, and Friday.</p>
Monthly by Date	<p>Runs a job on specific days of the month. The Months field specifies in which months this job executes. The Days field specifies which days of those months. If the given day does not exist for a given month, then that day is ignored.</p> <p>For example:</p> <p>To run a job at 5:00 P.M. on the 1st and 15th of January, February, and March, set the Begin Date to January 1, the Start Time and End Time to 5:00 P.M., the Interval in Minutes to 1, the Months to January, February, and March, and the Days to 1 and 15.</p> <p>To run a job at 2:00 A.M. on every leap day (February 29th), set the Begin Date to January 1, the Start Time and End Time to 2:00 A.M., the Interval in Minutes to 1, the Months to February, and the Days to 29.</p>

**Table 6–9 (Cont.) Job Manager Recurrent Trigger Types**

Trigger Type	Description and Example
Monthly by DOW (Day of Week)	<p>Runs a job on specific occurrences of specified days of the week during given months. The Months field specifies which months this job executes. The Days of the Week field specifies which days of the week the job executes during those months. The Occurrence field specifies which of those days to execute. The occurrence can be any or all of First, Second, Third, Fourth, and Last. The Last value specifies that either the fourth or fifth occurrence of a given day is used, depending on whether there are four or five occurrences during that month.</p> <p>For example:</p> <p>To run a job on the first and third Fridays of December every hour between the hours of 4:00 A.M. and 8:00 P.M., set the Begin Date to the desired date, the Start Time to 4:00 A.M., the End Time to 8:00 P.M., the Interval in Minutes to 60, the Months to December, the Days of the Week to Friday, and the Occurrence to the First and the Third.</p> <p>To run a job at 3:00 A.M. every time that Daylight Saving Time switches over to Standard Time, set the Begin Date to the desired date, the Start Time and End Time to 3:00 A.M., the Months to October, the Days of the Week to Sunday, and the Occurrence to Last.</p>



---

---

## Setting Up the SA System Subject Area

Oracle BI Presentation Services delivers alerts from Oracle BI Delivers to specified e-mail addresses, phone numbers, and so on. These delivery destinations are stored in the Oracle Business Intelligence profile for each user. In some cases, you might want to automatically populate the phone numbers or e-mail addresses in user profiles.

If you must automate only the population of e-mail addresses in user profiles, then you should populate the e-mail address field for users in your LDAP server or other authentication provider, if possible. These values are used to populate the e-mail address in Oracle Business Intelligence user profiles, enabling users to receive content from Delivers, even if they have not signed into Oracle Business Intelligence. This feature works for any LDAP server that has a mail attribute for its users.

In some cases, however, you might want to automatically populate additional user profile options, in addition to e-mail addresses. For example, you might want to automatically populate a cell phone number as part of the user profile information, if you want Delivers to deliver a format suitable for a cell phone (like text) using an e-mail gateway.

In this situation, you can configure a special subject area in the repository called SA System that retrieves user information from a database and populates the user profile information. This appendix explains how to configure and use the SA System subject area to accomplish this task.

If you choose to use the SA System subject area, then you should discourage users from configuring delivery profiles on their own. By default, values that are specified in delivery profiles take precedence over values that are shown in the SA System subject area.

This appendix contains the following topics:

- [Section A.1, "About the SA System Subject Area"](#)
- [Section A.2, "Setting Up the Data Source for the SA System Subject Area"](#)
- [Section A.3, "Importing SA System Data Into the Repository"](#)
- [Section A.4, "Setting Configuration Options for the SA System Subject Area"](#)

### A.1 About the SA System Subject Area

In previous releases of Oracle Business Intelligence, **SA System** was a subject area that exposed group membership to Delivers and enabled contact information, such as e-mail addresses to be retrieved from a database and used as delivery devices in Delivers. The SA System subject area feature automatically populated delivery devices and profiles for users instead of requiring users to update their My Account screen in

Delivers. The SA System subject area provided the users associated with each group and external e-mail addresses to Delivers.

In this release of Oracle Business Intelligence, Delivers still must determine group and role membership so that it can appropriately deliver alerts. Typically, however, your LDAP identity store is now the source of group and role membership. If SA System is defined and enabled, then membership of application roles and catalog groups is derived from the SA System subject area in Delivers. The names of the applications roles and catalog groups that are selected in an agent are used to determine group membership in the SA System subject area.

Note that you do not need SA System if you are using an LDAP server and you must populate only user profile e-mail addresses. The recommended best practice for populating e-mail addresses in user profiles is to use the mail attribute in your LDAP server. Because most portable devices can read e-mail directly, specific text or SMS formats are often not required for agent delivery, and populating e-mail addresses from LDAP is usually sufficient.

Also note that you do not need to use SA system to use the feature **Get Recipients from the Analysis Used in the Agent Condition**. Instead, this feature is used when the recipients can be determined from the query results and the data to be delivered is specific to those users.

Note that it is possible to configure initialization block-based user authentication using the tables in SA System as a source for the user population. Using the SA System data in this way is separate from using SA System to populate delivery profiles. Rather, these are independent functions that happen to be based on the same user source data.

### A.1.1 About Group and Application Role Resolution

In this release of Oracle Business Intelligence, application roles are used to define security policies rather than groups. When you create an agent, you can choose whether it should be delivered to a user, an application role, or a Catalog group.

However, to maintain backward compatibility with previous releases, SA System still uses the Group Name column in the SA System source table to determine the e-mail addresses for the application roles and catalog groups that are specified for agents. Because of this, the SA System subject area functions the same as it did in previous releases, even though the Oracle Business Intelligence security model has changed significantly in the current release.

Because the group membership in SA System is used to determine the list of recipients rather than the membership of either application roles or Catalog groups, users should not add members to Catalog groups. Alternatively, administrators can synchronize the application role and Catalog group memberships with SA System whenever the memberships are updated.

## A.2 Setting Up the Data Source for the SA System Subject Area

In your external data source, create a table called User that contains columns that correspond to the various delivery options. In addition, you must ensure that every user and group is present in the data.

[Table A-1](#) shows the columns that are required for the SA System subject area table. You must create the columns that are listed in the order shown. Any external schema that has the information in this table can be mapped to the SA System subject area.

**Table A-1 Columns in the SA System User Table**

Column	Data Type	Description
Logon	VARCHAR	The unique user ID of the user that logs on to the system. This cannot be null.
Display Name	VARCHAR	The full name of the user. This can be null.
Group Name	VARCHAR	The name of the group to which this user belongs. If a user belongs to multiple groups, then there should be one row for each group in the SA System table. This should not be null if any data access security is based on group membership.
Time Zone	VARCHAR	This column is currently not used and exists for future use. This should be null.
Language	VARCHAR	This column is currently not used and exists for future use. This should be null.
Locale	VARCHAR	This column is currently not used and exists for future use. This should be null.
Email	VARCHAR	The primary e-mail address for the user. This is a complete SMTP address such as joe.perez@example.com. This can be null.
Email Priority	VARCHAR	This determines when an alert is delivered to this device. The value can be any combination of the three priorities of an agent: H for high priority, N for normal priority, or L for low priority. For example, if high, normal, and low priority alerts are to be delivered to this device, then the field should be HNL. If only high and normal priority alerts are to be delivered, then the field should be HN. This field should not be null if the Email column is specified. This can be null if Email is null.
Email Type	VARCHAR	This field can be one of two text strings, HTML or text. Because most primary e-mail clients can read rich MIME content (HTML with embedded images), HTML is usually the best choice. Choose text to support legacy e-mail clients that can read only plain text e-mail. This field should not be null if the Email column is specified. This can be null if Email is null.
Cell Phone	VARCHAR	This field is the complete SMTP address for the cell phone device that receives text message alerts. For example, 1015551234@cellphoneprovider.com. Only text messages are sent to this device. This can be null.

**Table A-1 (Cont.) Columns in the SA System User Table**

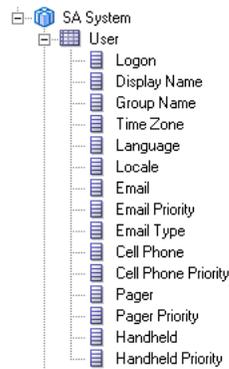
Column	Data Type	Description
Cell Phone Priority	VARCHAR	This determines when an alert is delivered to this device. The value can be any combination of the three priorities of an agent: H for high priority, N for normal priority, and L for low priority. This field should not be null if the Cell Phone column is specified.  This can be null if Cell Phone is null.
Pager	VARCHAR	This field is the complete SMTP address for the pager device that receives text message alerts. For example, 1015555678@pagerprovider.com. Only text messages are sent to this device.  This can be null.
Pager Priority	VARCHAR	This determines when an alert is delivered to this device. The value can be any combination of the three priorities of an agent: H for high priority, N for normal priority, and L for low priority.  This field should not be null if the Pager column is specified. This can be null if Pager is null.
Handheld	VARCHAR	This field is the complete SMTP address for the handheld device that receives text message alerts. For example, joe.perez@handheldprovider.com. Only text messages are sent to this device.  This can be null.
Handheld Priority	VARCHAR	This determines when an alert is delivered to this device. The value can be any combination of the three priorities of an agent: H for high priority, N for normal priority, and L for low priority.  This field should not be null if the Handheld column is specified. This can be null if Handheld is null.

### A.3 Importing SA System Data Into the Repository

After you configure the external data source, you must create and build the subject area in the Oracle BI repository. To do this, you first import the User table from the data source into the Physical layer. Then, map the User table and columns from the Physical layer to the Business Model and Mapping layer. Finally, map the User table and columns from the Business Model and Mapping layer to the Presentation layer. The name for the subject area must always be SA System.

See *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* for information about importing data into the repository and modeling information in the Business Model and Mapping layer and Presentation layer.

Figure A-1 shows the table and columns in the Presentation layer of the repository.

**Figure A-1 SA System Subject Area in the Presentation Layer**

## A.4 Setting Configuration Options for the SA System Subject Area

You can control the availability of the delivery options that are configured in the SA System subject area and the user-defined delivery options by including certain elements in the instanceconfig.xml file. These elements take effect only if the SA System subject area is being used.

Before you begin this procedure, ensure that you are familiar with the information in "Using a Text Editor to Update Oracle Business Intelligence Configuration Settings" in *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

### To change configuration options for the SA System subject area:

1. Open the instanceconfig.xml file for editing, as described in "Where are Configuration Files Located?" in *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*
2. Locate the sections in which you must add the following elements:
  - SystemSubjectArea: Specifies whether to recognize the delivery devices and deliver profiles that are configured in the SA System subject area:
    - **True.** Recognizes the delivery devices and delivery profiles that are configured in the SA System subject area and displays them on the My Account page. (Default)
    - **False.** Ignores the delivery devices and delivery profiles that are configured in the SA System subject area and does not display them on the My Account page.

Include this element within the ServerInstance element.

- IgnoreWebcatDeliveryProfiles: Specifies whether to ignore user-defined delivery devices and deliver profiles:
  - **True.** Ignores the user-defined delivery devices and delivery profiles and does not display them on the My Account page. (This means that users cannot create new delivery devices and delivery profiles.)
  - **False.** Recognizes the user-defined delivery devices and delivery profiles and displays them on the My Account page. (Default)

Include this element within the Alerts element, which is itself included in the ServerInstance element.

- `UpperCaseRecipientNames`: Specifies that only users whose user names are uppercase can have agents delivered to them.

For example, suppose that you have users with names of `user_lowercase` and `USER_UPPERCASE`. If you set the `UpperCaseRecipientNames` element to `true`, then agents are sent only to `USER_UPPERCASE`.

Include this element within the `Alerts` element, which is itself included in the `ServerInstance` element.

3. Include the elements and their ancestor elements as appropriate, as shown in the following example.

```
<ServerInstance>
  <SubjectAreaMetadata>
    <SystemSubjectArea>true</SystemSubjectArea>
  </SubjectAreaMetadata>
  <Alerts>
    <IgnoreWebcatDeliveryProfiles>false</IgnoreWebcatDeliveryProfiles>
    <UpperCaseRecipientNames>true</UpperCaseRecipientNames>
  </Alerts>
</ServerInstance>
```

4. Save your changes and close the file.
5. Restart Oracle Business Intelligence.

#### A.4.1 Managing the Case of Login Names for the SA System Subject Area

When the SA System subject area is used, login names are compared to the `Logon` column in the SA System subject area. By default, this comparison is case-sensitive. This means, for example, that a login of "Fred" does not match an SA System subject area entry of "fred." If the authentication method is case-sensitive, then this works fine because the login "fred," accepted at login, matches "fred" in the `Logon` column in the SA System subject area.

If the case of login names does not match, then invalid user name errors might result. To avoid this situation, ensure that the `SA_SYSTEM Logon` value and the LDAP user name value have the same case. For example, set the `LOGON` value in the `SA_USER` table to lowercase by including it within the `Lower()` function.

You can also use the `UpperCaseRecipientNames` element in the `instanceconfig.xml` file to ensure delivery only to users whose names are uppercase.

---

---

# Glossary

This glossary defines terms for Oracle Business Intelligence Enterprise Edition. See also the Oracle Fusion Middleware Master Glossary for additional terms and definitions.

## **action**

Provides functionality to navigate to related content or to invoke operations, functions or processes in external systems. You can include actions in analyses, dashboard pages, agents, scorecard objectives, scorecard initiatives, and KPIs.

See also [action link](#).

## **Action Framework**

The Action Framework is a component of the Oracle BI EE architecture and includes a J2EE application called the Action Execution Service (AES) and actions-specific JavaScript functionality deployed as part of Oracle BI EE. The action framework also includes client-side functionality for creating actions and invoking certain action types directly from the browser.

## **action link**

A link to an action that you have embedded in an analysis, dashboard page, scorecard objective, scorecard initiative, or KPI that, when clicked, runs an associated action.

See also [action](#).

## **ADF Business Intelligence Component**

Provides the developer the ability to include Oracle BI Presentation Catalog objects in ADF Applications. This component uses a SOAP connection to access the catalog.

## **Administration Server**

Part of the WebLogic server domain and runs the processes that manage Oracle Business Intelligence components. The Administration Server includes the Oracle WebLogic Server Administration Console, Oracle Fusion Middleware Control, and JMX MBeans. For a Simple Install type, the Administration Server also includes Java components for Oracle Business Intelligence such as Oracle BI Publisher and Oracle Real-Time Decisions.

See also [Fusion Middleware Control](#), [Java components](#) and [Managed Server](#).

## **Administration Tool**

See [Oracle BI Administration Tool](#).

**agent**

Enables you to automate your business processes. You can use agents to provide event-driven alerting, scheduled content publishing, and conditional event-driven action execution.

Agents can dynamically detect information-based problems and opportunities, determine the appropriate individuals to notify, and deliver information to them through a wide range of devices (e-mail, phones, and so on).

**aggregate persistence**

A feature that automates the creation and loading of aggregate tables and their corresponding Oracle Business Intelligence metadata mappings to enable aggregate navigation.

**aggregate table**

A table that stores precomputed results from measures that have been aggregated over a set of dimensional attributes. Each aggregate table column contains data at a given set of levels. For example, a monthly sales table might contain a precomputed sum of the revenue for each product in each store during each month. Using aggregate tables optimizes performance.

**aggregation rule**

In an Oracle BI repository, a rule applied to a logical column or physical cube column that specifies a particular aggregation function to be applied to the column data, such as SUM.

In Presentation Services, users can see the rules that have been applied in the repository. Users can also change the default aggregation rules for measure columns.

**alias table**

A physical table that references a different physical table as its source. You can use alias tables to set up multiple tables, each with different keys, names, or joins, when a single physical table must serve in different roles. Because alias table names are included in physical SQL queries, you can also use alias tables to provide meaningful table names, making the SQL statements easier to read.

**analysis**

A query that a user creates on the Criteria tab in Presentation Services. An analysis can optionally contain one or more filters or selection steps to restrict the results.

See also [filter](#) and [selection step](#).

**analysis criteria**

Consists of the columns, filters, and selection steps that you specify for an analysis.

See also [analysis](#).

**analysis prompt**

A prompt that is added to an analysis. When the user selects a prompt value, that value then determines the content that displays in the analysis that contains the prompt, only.

See [dashboard prompt](#) and [prompt](#).

**attribute**

The details of a dimension in an Oracle BI repository. Attributes usually appear as columns of a dimension table.

**attribute column**

In Presentation Services, a column that holds a flat list of values that are also known as members. No hierarchical relationship exists between these members, as is the case for members of a hierarchical column. Examples include ProductID or City.

See [hierarchical column](#).

**BI Composer**

BI Composer is a simple-to-use wizard that enables you to quickly and easily create, edit, or view analyses without the complexities of the Analysis editor.

**BI domain**

Contains configurable system components (the coreapplication) and Java components (the WebLogic server domain), and includes the Web-based management tools and applications that use resources.

A BI domain can be a set of middleware homes spread across one or more physical servers.

See also [BI instance](#).

**BI instance**

Refers to the system components (coreapplication) of a BI domain

See also [BI domain](#).

**BI object**

A piece of business intelligence content that is created with Presentation Services and saved to the Oracle BI Presentation Catalog. Examples of BI objects include analyses, dashboards, dashboard pages, scorecards, and KPIs.

**BI Search**

A search tool that resides outside of Presentation Services. BI Search is available from the Home Page after the administrator adds a link to the BI Search URL. BI Search provides a mechanism for searching for objects in the Oracle BI Presentation Catalog that is similar to a full-text search engine.

**bookmark link**

Captures the path to a dashboard page and all aspects of the page state.

See [prompted link](#).

**bridge table**

A table that enables you to resolve many-to-many relationships between two other tables.

**briefing book**

See [Oracle BI Briefing Books](#).

**business model**

An object in the Oracle BI repository that contains the business model definitions and the mappings from logical to physical tables. Business models are always dimensional,

unlike objects in the Physical layer, which reflect the organization of the data sources. Each business model contains logical tables, columns, and joins.

### **Business Model and Mapping layer**

A layer of the Oracle BI repository that defines the business, or logical, model of the data and specifies the mapping between the business model and the Physical layer schemas. This layer can contain one or more business models.

The Business Model and Mapping layer determines the analytic behavior that is seen by users, and defines the superset of objects available to users. It also hides the complexity of the source data models.

### **business owner**

The person responsible for managing and improving the business value and performance of a KPI or scorecard object, such as an objective, cause & effect map, and so on.

### **catalog**

See [Oracle BI Presentation Catalog](#).

### **cause & effect map**

A component of a scorecard that lets you illustrate the cause and effect relationships of an objective or KPI.

See also [Oracle Scorecard and Strategy Management](#).

### **chronological key**

A column in a time dimension that identifies the chronological order of the members within a dimension level. The key must be unique at its level.

### **Cluster Controller**

A process that serves as the first point of contact for new requests from Presentation Services and other clients. The Cluster Controller determines which Oracle BI Server in the cluster to direct the request to based on Oracle BI Server availability and load. It monitors the operation of servers in the cluster, including the Oracle BI Scheduler instances. The Cluster Controller is deployed in active-passive configuration.

### **column**

In an Oracle BI repository, columns can be physical columns, logical columns, or presentation columns.

In Presentation Services, indicates the pieces of data that an analysis returns. Together with filters and selection steps, columns determine what analyses contain. Columns also have names that indicate the types of information that they contain, such as Account and Contact.

See also [analysis](#), [attribute column](#), [hierarchical column](#), and [measure column](#).

### **column filter**

See [filter](#).

### **column prompt**

A type of filter that enables you to build specific value prompts on a data column to either exist alone on the dashboard or analysis or to expand or refine existing dashboard and analysis filters.

See also [prompt](#).

**complex join**

A join in the Physical layer of an Oracle BI repository that uses an expression other than equals.

**condition**

Objects that return a single Boolean value based on the evaluation of an analysis or of a key performance indicator (KPI). You use conditions to determine whether agents deliver their content and execute their actions, whether actions links are displayed in dashboard pages, or whether sections and their content are displayed in dashboard pages.

See also [action](#), [action link](#), [agent](#) and [key performance indicator \(KPI\)](#).

**connection pool**

An object in the Physical layer of an Oracle BI repository that contains the connection information for a data source.

See also [Physical layer](#).

**content designer**

The user who creates business intelligence objects such as analyses, dashboards, and scorecards.

**contextual event action**

A predelivered action that uses the Action Framework to pass content from the business intelligence object to another region on an ADF page.

See also [action](#), [Action Framework](#), and [action link](#).

**criteria**

See [analysis criteria](#).

**cube**

An OLAP (online analytical processing) data structure that lets data be analyzed more quickly and with greater flexibility than structures in relational databases. Cubes are made up of measures and organized by dimensions. Cubes in multidimensional data sources roughly correspond to star schemas in relational database models.

**currency prompt**

A prompt that enables the user to change the currency type that displays in the currency columns on an analysis or dashboard.

See also [prompt](#).

**custom view**

A component of a scorecard that lets you show a customized view of your business and strategy data.

See also [Oracle Scorecard and Strategy Management](#).

**dashboard**

An object that provides personalized views of corporate and external information. A dashboard consists of one or more pages. Pages can display anything that you can

access or open with a Web browser, such as results of analyses, images, alerts from agents, and so on.

### **dashboard prompt**

A prompt that is added to the dashboard. When the user selects a prompt value, that value then determines the content that displays in all analyses that are included on the dashboard.

See [analysis prompt](#) and [prompt](#).

### **Dashboard URL**

Used for incorporating or referencing the content of a specific dashboard in external portals or applications. It has several forms and optional arguments that you can use to control its behavior.

### **data source name (DSN)**

A data structure that contains the information about a specific database, typically used by an ODBC driver to connect to the database. The DSN contains information such as the name, directory, and driver of the database.

Connection pool objects in the Physical layer of the Oracle BI repository contain DSN information for individual data sources.

### **database hint**

Instructions placed within a SQL statement that tell the database query optimizer the most efficient way to execute the statement. Hints override the optimizer's execution plan, so you can use hints to improve performance by forcing the optimizer to use a more efficient plan. Hints are supported only for Oracle Database data sources.

### **dimension**

A hierarchical organization of logical columns (attributes). One or more logical dimension tables might be associated with at most one dimension.

A dimension might contain one or more (unnamed) hierarchies. There are two types of logical dimensions: dimensions with level-based hierarchies (structure hierarchies), and dimensions with parent-child hierarchies (value hierarchies).

A particular type of level-based dimension, called a time dimension, provides special functionality for modeling time series data.

See also [hierarchy](#).

### **dimension table**

A logical table that contains columns used by a particular dimension. A dimension table cannot be a fact table.

See also [fact table](#).

### **driving table**

A mechanism used to optimize the manner in which the Oracle BI Server processes multi-database joins when one table is very small (the driving table) and the other table is very large.

### **DSN**

See [data source name \(DSN\)](#).

**event polling table**

Event polling tables (also called event tables) provide information to the Oracle BI Server about which physical tables have been updated. They are used to keep the query cache up-to-date. The Oracle BI Server cache system polls the event table, extracts the physical table information from the rows, and purges stale cache entries that reference those physical tables.

**Essbase**

A multidimensional database management system available from Oracle that provides a multidimensional database platform upon which to build business intelligence applications. Also referred to as Oracle's Hyperion Essbase.

**fact table**

In an Oracle BI repository, a logical table in the Business Model and Mapping layer that contains measures and has complex join relationships with dimension tables.

See also [dimension table](#).

**filter**

Criteria that are applied to attribute and measure columns to limit the results that are displayed when an analysis is run. For measure columns, filters are applied before the query is aggregated and affect the query and thus the resulting values.

See also [prompt](#) and [selection step](#).

**foreign key**

A column or a set of columns in one table that references the primary key columns in another table.

**fragmentation content**

The portion, or fragment, of the set of data specified in a logical table source when the logical table source does not contain the entire set of data at a given level.

Fragmentation content is defined by the logical columns that are entered in the **Fragmentation content** box in the Content tab of the Logical Table Source dialog box.

**Fusion Middleware Control**

Provides Web-based management tools that enable you to monitor and configure Fusion Middleware components.

**global header**

An object in the user interface for Oracle BI Presentation Services that contains links and options that enable the user to quickly begin a task or locate a specific object within the Oracle BI Presentation Catalog. The global header always displays in the Presentation Services user interface, thus enabling users to quickly access links and search the catalog without having to navigate to the Home Page or Catalog page.

**Go URL**

Used to incorporate specific business intelligence results into external portals or applications. The Go URL is used when you add a result to your favorites or add a link to a request to a dashboard or external Web site. It has several forms and optional arguments that you can use to control its behavior.

**hierarchical column**

In Presentation Services, a column that holds data values that are organized using both named levels and parent-child relationships. This column is displayed using a tree-like structure. Individual members are shown in an outline manner, with lower-level members rolling into higher-level members. For example, a specific day belongs to a particular month, which in turn is within a particular year. Examples include Time or Geography.

See also [attribute column](#).

**hierarchy**

In an Oracle BI repository, a system of levels in a logical dimension that are related to each other by one-to-many relationships. All hierarchies must have a common leaf level and a common root (all) level.

Hierarchies are not modeled as separate objects in the metadata. Instead, they are an implicit part of dimension objects.

See also [dimension](#), [logical level](#), and [presentation hierarchy](#).

**hierarchy level**

In Presentation Services, an object within a hierarchical column that either rolls up or is rolled up from other levels. Corresponds to a presentation level in an Oracle BI repository.

See also [presentation level](#).

**home page**

Provides an intuitive, task-based entry way into the functionality of Presentation Services. The Home page is divided into sections that enable you to quickly begin specific tasks, locate an object, or access technical documentation.

**image prompt**

A prompt that provides an image with different areas mapped to specific values. The user clicks an image area to select the prompt value that populates the analysis or dashboard.

See also [prompt](#).

**initialization block**

Used to initialize dynamic repository variables, system session variables, and nonsystem session variables. An initialization block contains the SQL statements that are executed to initialize or refresh the variables that are associated with that block.

**initiative**

Used in a scorecard, an initiative is a time-specific task or project that is necessary to achieve objectives. As such, you can use initiatives that support objectives as milestones as they reflect progress toward strategy targets.

See also [objective](#) and [Oracle Scorecard and Strategy Management](#).

**Java components**

Fusion Middleware Control components that are deployed as one or more Java EE applications (and a set of resources) and are managed by Node Manager.

See also [Node Manager](#).

**key performance indicator (KPI)**

A measurement that defines and tracks specific business goals and strategic objectives. KPIs often times roll up into larger organizational strategies that require monitoring, improvement, and evaluation. KPIs have measurable values that usually vary with time, have targets to determine a score and performance status, include dimensions to allow for more precise analysis, and can be compared over time for trending purposes and to identify performance patterns.

See also [Oracle Scorecard and Strategy Management](#).

**KPI watchlist**

A method of distributing KPIs to end users. A watchlist is a collection of KPIs that are built by adding the KPIs that are stored in the Oracle BI Presentation Catalog. After a KPI watchlist is built and saved, it is stored as a catalog object and can be added to dashboards and scorecards.

See also [key performance indicator \(KPI\)](#).

**level**

See [hierarchy level](#).

**logical display folder**

Folders used to organize objects in the Business Model and Mapping layer of an Oracle BI repository. They have no metadata meaning.

**logical join**

Joins that express relationships between logical tables. Logical joins are *conceptual*, rather than physical, joins. In other words, they do not join to particular keys or columns. A single logical join can correspond to many possible physical joins.

**logical layer**

See [Business Model and Mapping layer](#).

**logical level**

In an Oracle BI repository, a component of a level-based hierarchy that either rolls up or is rolled up from other levels.

Parent-child hierarchies have implicit, inter-member levels between ancestors and descendants that are not exposed as logical level objects in the metadata. Although parent-child hierarchies also contain logical level objects, these levels are system generated and exist to enable aggregation across all members only.

See also [dimension](#) and [hierarchy](#).

**Logical SQL**

The SQL statements that are understood by the Oracle BI Server. The Oracle BI Server Logical SQL includes standard SQL, plus special functions (SQL extensions) like `AGO`, `TODATE`, `EVALUATE`, and others.

Clients like Presentation Services send Logical SQL to the Oracle BI Server when a user makes a request. In addition, Logical SQL is used in the Business Model and Mapping layer to enable heterogeneous database access and portability. The Oracle BI Server transforms Logical SQL into physical SQL that can be understood by source databases.

**logical table**

A table object in the Business Model and Mapping layer of an Oracle BI repository. A single logical table can map to one or more physical tables. Logical tables can be either fact tables or dimension tables.

See also [dimension table](#) and [fact table](#).

**logical table source**

Objects in the Business Model and Mapping layer of an Oracle BI repository that define the mappings from a single logical table to one or more physical tables. The physical to logical mapping can also be used to specify transformations that occur between the Physical layer and the Business Model and Mapping layer, and to enable aggregate navigation and fragmentation.

**Managed Server**

An individual J2EE application container (JMX MBean container). It provides local management functions on individual hosts for Java components and system components contained within the local middleware home, and refers to the Administration Server for all of its configuration and deployment information.

See also [Administration Server](#) and [Fusion Middleware Control](#).

**MDS**

Oracle Metadata Services. A core technology of the Application Development Framework. MDS provides a unified architecture for defining and using metadata in an extensible and customizable manner.

See also [MDS XML](#).

**MDS XML**

An XML format that is compatible with Oracle Metadata Services. MDS XML is a supported format for the Oracle BI repository. It enables integration with third-party source control management systems for offline repository development.

MDS XML format is different from the XML format generated by the Oracle BI Server XML API.

See also [MDS](#), [Oracle BI repository](#), and [Oracle BI Server XML API](#).

**measure column**

A column that can change for each record and can be added up or aggregated. Typical measures are sales dollars and quantity ordered. Measures are calculated from data sources at query time.

Measure columns are displayed in the Oracle BI repository, usually in fact tables, or in Presentation Services.

**metadata**

Data about data. Metadata objects include the descriptions of schemas (such as tables, columns, data types, primary keys, foreign keys, and so on) and logical constructs (like fact tables, dimensions, and logical table source mappings).

The Oracle BI repository is made up of the metadata used by the Oracle BI Server to process queries.

**metadata dictionary**

A static set of XML documents that describe metadata objects, such as a column, including its properties and relationships with other metadata objects. A metadata dictionary can help users obtain more information about metrics or attributes for repository objects.

**mission statement**

A statement in a scorecard that specifies the key business goals and priorities that are required to achieve your vision.

See also [Oracle Scorecard and Strategy Management](#) and [vision statement](#).

**multi-database join**

A join between two tables in an Oracle BI repository, where each table resides in a different database.

**Node Manager**

A daemon process that provides remote server start, stop, and restart capabilities when Java processes become unresponsive or terminate unexpectedly.

See also [Java components](#).

**OCI**

See [Oracle Call Interface \(OCI\)](#).

**ODBC**

See [Open Database Connectivity \(ODBC\)](#).

**object properties**

Information about an object and attributes that the owner can assign to an object. Examples of properties include name, description, date stamps, read-only access, and do not index flag.

See also [permissions](#).

**objective**

A required or desired outcome in a scorecard that forms your corporate strategy.

See also [initiative](#) and [Oracle Scorecard and Strategy Management](#).

**offline mode**

In the Oracle BI Administration Tool, a mode where a repository builder can edit a repository that is not loaded into the Oracle BI Server.

**online mode**

In the Oracle BI Administration Tool, a mode where a repository builder can edit a repository while it is available for query operations. Online mode also allows user session monitoring for users connected to the subject areas in the repository.

**opaque view**

A Physical layer table that consists of a `SELECT` statement. In the Oracle BI repository, opaque views appear as view tables in the physical databases, but the view does not actually exist.

### **Open Database Connectivity (ODBC)**

A standard interface used to access data in both relational and nonrelational databases. Database applications can use ODBC to access data stored in different types of database management systems, even if each database uses a different data storage format and programming interface.

### **OPMN**

See [Oracle Process Manager and Notification Server \(OPMN\)](#).

### **Oracle BI Administration Tool**

A Windows application that is used to create and edit Oracle BI repositories. The Administration Tool provides a graphical representation of the three parts of a repository: the Physical layer, the Business Model and Mapping layer, and the Presentation layer.

### **Oracle BI Briefing Books**

A collection of static or updatable snapshots of dashboard pages, individual analyses, and BI Publisher reports. You can download briefing books in PDF or MHTML format for printing and viewing. You also can update, schedule, and deliver briefing books using agents.

### **Oracle BI JavaHost**

A service that gives Presentation Services the ability to use functionality that is provided in Java libraries to support components such as graphs. The services are provided based on a request-response model.

### **Oracle BI Logical SQL View Object**

Provides the developer the ability to create a Logical SQL statement to access the Oracle BI Server and fetch business intelligence data and bind it to native ADF components for inclusion on an ADF page. This view object uses a BI JDBC connection to the Oracle BI Server.

### **Oracle BI Presentation Catalog**

Stores business intelligence objects, such as analyses and dashboards, and provides an interface where users create, access, and manage objects, and perform specific object-based tasks (for example, export, print, and edit). The catalog is organized into folders that are either shared or personal.

### **Oracle BI Presentation Services**

Provides the framework and interface for the presentation of business intelligence data to Web clients. It maintains a Presentation Catalog service on the file system for the customization of this presentation framework. It is a standalone process and communicates with the Oracle BI Server using ODBC over TCP/IP. It consists of components that are known as Answers, Delivers, and Interactive Dashboards.

See also [ODBC](#); [Oracle BI Server](#); [Oracle BI Presentation Catalog](#); [Oracle BI Presentation Services server](#).

### **Oracle BI Presentation Services server**

The Oracle BI Web server that exchanges information and data with the Oracle BI Server.

**Oracle BI Publisher**

A J2EE application that provides enterprise-wide publishing services in Oracle Business Intelligence. It generates highly formatted, pixel-perfect reports.

See also [report](#).

**Oracle BI Publisher report**

See [report](#).

**Oracle BI repository**

The set of Oracle Business Intelligence metadata that defines logical schemas, physical schemas, physical-to-logical mappings, aggregate table navigation, and other constructs. Oracle BI repositories can be in binary (RPD) format, in which repository metadata is contained in a single file with an extension of .rpd, or in a set of MDS XML documents. MDS XML format repositories are used for offline development only and cannot be loaded into the Oracle BI Server. Oracle BI repositories in both formats can be edited using the Oracle BI Administration Tool.

See also [metadata](#) and [Oracle BI Administration Tool](#).

**Oracle BI Scheduler**

An extensible scheduling application for scheduling results to be delivered to users at specified times. It is the engine behind the Oracle BI Delivers feature.

See also [results](#).

**Oracle BI Server**

A standalone process that maintains the logical data model that it provides to Presentation Services and other clients through ODBC. Metadata is maintained for the data model in a local proprietary file called the repository file. The Oracle BI Server processes user requests and queries underlying data sources.

**Oracle BI Server XML API**

Provides utilities to create a generic, XML-based representation of the Oracle BI repository metadata. You can use this XML file version of the repository to programmatically modify the metadata. The Oracle BI Server XML API objects correspond to metadata repository objects in an RPD file. These objects differ from XML objects in the Oracle BI Presentation Catalog.

The XML generated by the Oracle BI Server XML API is different from the MDS XML format used for Oracle BI repositories integrated with third-party source control management systems.

See also [MDS XML](#).

**Oracle Business Intelligence Mobile**

Oracle Business Intelligence Mobile allows you to view Oracle BI EE content on supported mobile devices such as the Apple iPhone and Apple iPad.

Using Oracle Business Intelligence Mobile, you can view and analyze BI content such as analyses and dashboards, BI Publisher content, scorecard content, and content delivered by agents.

**Oracle Business Intelligence Web Services**

See [Oracle Business Intelligence Session-Based Web Services](#) and [Oracle Business Intelligence Web Services for SOA](#).

### **Oracle Business Intelligence Session-Based Web Services**

An API that implements SOAP. These Web services are designed for programmatic use, where a developer uses one Web service to invoke many different business intelligence objects. These Web services provide functionality on a wide range of Presentation Services operations. These Web services enable the developer to extract results from Oracle BI Presentation Services and deliver them to external applications, perform Presentation Services management functions, and execute Oracle Business Intelligence alerts (known as Intelligent Agents).

See also [Oracle Business Intelligence Web Services for SOA](#).

### **Oracle Business Intelligence Web Services for SOA**

Contains three Web services, ExecuteAgent, ExecuteAnalysis, and ExecuteCondition, which are hosted by the bimiddleware J2EE application. These web services are designed to enable developers to use third-party Web services clients (for example, Oracle SOA Suite) to browse for and include business intelligence objects in service oriented architecture components.

See also [Oracle Business Intelligence Session-Based Web Services](#).

### **Oracle Call Interface (OCI)**

A connection interface that the Oracle BI Server can use to connect to Oracle Database data sources. You should always use OCI when importing metadata from or connecting to an Oracle Database.

### **Oracle OLAP**

Oracle Database has an OLAP Option that provides an embedded, full-featured online analytical processing server.

Oracle Business Intelligence supports Oracle OLAP as a data source. When you import metadata from an Oracle OLAP source, the Oracle OLAP objects appear in the Physical layer of the Administration Tool. Oracle OLAP objects include Analytic Workspaces, which are containers for storing related cubes.

### **Oracle Process Manager and Notification Server (OPMN)**

A process management tool that manages all system components (server processes), and supports both local and distributed process management, automatic process recycling and the communication of process state (up, down, starting, stopping). OPMN detects process unavailability and automatically restarts processes).

See also [system components](#).

### **Oracle Scorecard and Strategy Management**

A performance management tool that lets you describe and communicate your business strategy. You can drive and assess your corporate strategy and performance from the top of your organization down, or from the bottom up.

### **Oracle Technology Network (OTN)**

A repository of technical information about Oracle's products where you can search for articles, participate in discussions, ask the user community technical questions, and search for and download Oracle products and documentation.

### **parent-child hierarchy**

A hierarchy of members that all have the same type. All the dimension members of a parent-child hierarchy occur in a single data source. In a parent-child hierarchy, the

inter-member relationships are parent-child relationships between dimension members.

See also [dimension](#).

### **parent-child relationship table**

A table with values that explicitly define the inter-member relationships in a parent-child hierarchy. Also called a closure table.

### **pass-through calculation**

A calculation that is not computed by the Oracle BI Server but instead is passed to another data source. Enables advanced users to leverage data source features and functions without the need to modify the Oracle BI repository.

### **permissions**

Specify which users can access an object, and limit how users can interact with an object. Examples of permissions include write, delete, and change permissions.

See [object properties](#).

### **perspective**

A category in your organization with which to associate initiatives, objectives, and KPIs in a scorecard. A perspective can represent a key stakeholder (such as a customer, employee, or shareholder/financial) or a key competency area (such as time, cost, or quality).

See also [initiative](#), [key performance indicator \(KPI\)](#), [objective](#), and [Oracle Scorecard and Strategy Management](#).

### **physical catalog**

An object in the Physical layer of an Oracle BI repository that groups different schemas. A catalog contains all the schemas (metadata) for a database object.

### **physical display folder**

Folders that organize objects in the Physical layer of an Oracle BI repository. They have no metadata meaning.

### **physical join**

Joins between tables in the Physical layer of an Oracle BI repository.

### **Physical layer**

A layer of the Oracle BI repository that contains objects that represent physical data constructs from back-end data sources. The Physical layer defines the objects and relationships available for writing physical queries. This layer encapsulates source dependencies to enable portability and federation.

### **physical schema**

An object in the Physical layer of an Oracle BI repository that represents a schema from a back-end database.

### **physical table**

An object in the Physical layer of an Oracle BI repository, usually corresponding to a table that exists in a physical database.

See also [Physical layer](#).

### **presentation hierarchy**

An object in the Presentation layer of an Oracle BI repository that provides an explicit way to expose the multidimensional model in Presentation Services and other clients. Presentation hierarchies expose analytic functionality such as member selection, custom member groups, and asymmetric queries. Users can create hierarchy-based queries using presentation hierarchies.

In Presentation Services, presentation hierarchies are displayed as hierarchical columns.

See also [hierarchical column](#) and [presentation level](#).

### **Presentation layer**

Provides a way to present customized, secure, role-based views of a business model to users. It adds a level of abstraction over the Business Model and Mapping layer in the Oracle BI repository. The Presentation layer provides the view of the data seen by users who build analyses in Presentation Services and other client tools and applications.

See also [Business Model and Mapping layer](#).

### **presentation level**

In the Oracle BI repository, a component of a presentation hierarchy that either rolls up or is rolled up from other levels. Presentation levels are displayed as levels within hierarchical columns in Presentation Services.

See also [hierarchy level](#) and [presentation hierarchy](#).

### **Presentation Services**

See [Oracle BI Presentation Services](#).

### **Presentation Services server**

See [Oracle BI Presentation Services server](#).

### **presentation table**

An object in the Presentation layer of an Oracle BI repository that is used to organize columns into categories that make sense to the user community. A presentation table can contain columns from one or more logical tables. The names and object properties of the presentation tables are independent of the logical table properties.

### **primary key**

A column (or set of columns) where each value is unique and identifies a single row of a table.

### **process instance**

A unique process on an individual workstation that is associated with a BI instance.

See also [BI instance](#).

### **prompt**

A type of filter that enables the content designer to build and specify data values or the end user to choose specific data values to provide a result sets for an individual analysis or multiple analyses included on a dashboard or dashboard page. A prompt expands or refines existing dashboard and analysis filters.

The types of prompts are column prompts, currency prompts, image prompts, and variable prompts.

---

See also [column prompt](#), [currency prompt](#), [filter](#), [image prompt](#), and [variable prompt](#).

**prompted link**

Captures the path to a dashboard page and a simplified presentation of the dashboard prompt.

See [bookmark link](#).

**query**

Contains the underlying SQL statements that are issued to the Oracle BI Server. You do not have to know a query language to use Oracle Business Intelligence.

**query cache**

A facility to store query results for use by other queries.

**ragged hierarchy**

See [unbalanced hierarchy](#).

**report**

The response returned to the user from the execution of a query created using Oracle BI Publisher. Reports can be formatted, presented on a dashboard page, saved in the Oracle BI Presentation Catalog, and shared with other users.

See also [analysis](#).

**repository**

See [Oracle BI repository](#).

**repository variable**

See [variable](#).

**results**

The output returned from the Oracle BI Server for an analysis.

See also [analysis](#).

**scorecard**

See [Oracle Scorecard and Strategy Management](#).

**selection step**

A choice of values that is applied after the query is aggregated that affects only the members displayed, not the resulting aggregate values. Along with filters, selection steps restrict the results for an analysis.

See also [analysis](#) and [filter](#).

**session variable**

See [variable](#).

**skip-level hierarchy**

A hierarchy where some members do not have a value for a particular ancestor level. For example, in the United States, the city of Washington in the District of Columbia does not belong to a state. The expectation is that users can still navigate from the country level (United States) to Washington and below without the need for a state.

See also [hierarchy](#).

**snowflake schema**

A dimensional schema where one or more of the dimensions are partially or completely normalized.

**SQL**

See [structured query language \(SQL\)](#).

**star schema**

A relational schema that allows dimensional analysis of historical information. Star schemas have one-to-many relationships between the logical dimension tables and the logical fact table. Each star consists of a single fact table joined to a set of denormalized dimension tables.

**strategy map**

A component of a scorecard that shows how the objectives that have been defined for a scorecard and the KPIs that measure their progress are aligned by perspectives. It also shows cause and effect relationships.

See also [Oracle Scorecard and Strategy Management](#).

**strategy tree**

A component of a scorecard that shows an objective and its supporting child objectives and KPIs hierarchically in a tree diagram.

See also [Oracle Scorecard and Strategy Management](#).

**structured query language (SQL)**

A standard programming language for querying and modifying data. Oracle Business Intelligence supports standard SQL-92 with several value-added proprietary extensions.

See also [Logical SQL](#).

**subject area**

In an Oracle BI repository, an object in the Presentation layer that organizes and presents data about a business model. It is the highest-level object in the Presentation layer and represents the view of the data that users see in Presentation Services. Oracle BI repository subject areas contain presentation tables, presentation columns, and presentation hierarchies.

In Presentation Services, subject areas contain folders, measure columns, attribute columns, hierarchical columns, and levels.

**system components**

Server processes (not Java applications) that are managed by the Oracle Process Manager and Notification server (OPMN).

See also [Oracle Process Manager and Notification Server \(OPMN\)](#).

**transformation**

Work that is performed on data when moving from a database to another location (sometimes another database). Some transformations are typically performed on data when it is moved from a transaction system to a data warehouse system.

**unbalanced hierarchy**

A hierarchy where the leaves do not have the same depth. For example, an organization might choose to have data for the current month at the day level, data for the previous at the month level, and data for the previous five years at the quarter level.

See also [hierarchy](#).

**value hierarchy**

See [parent-child hierarchy](#).

**variable**

Objects in an Oracle BI repository that are used to streamline administrative tasks and dynamically modify metadata content to adjust to a changing data environment.

Variables are of the following types:

- Repository variables have a single value at any point in time. There are two types of repository variables: static and dynamic.
- Session variables are created and assigned a value when each user logs on. There are two types of session variables: system and nonsystem.

**variable prompt**

Enables the user to select a value specified in the variable prompt to display on the dashboard. A variable prompt is not dependent upon column data, but enables you to manipulate, for example add or multiply, the column data on an analysis.

See also [prompt](#).

**virtual physical table**

A physical table that is made from a stored procedure or a `SELECT` statement. Creating virtual tables can provide the Oracle BI Server and the underlying databases with the proper metadata to perform some advanced query requests.

**vision statement**

A short statement in a scorecard that describes what your organization wants to become sometime in the future. For example, it might be to become the most successful business in the South America Polypropylene Market.

See also [mission statement](#) and [Oracle Scorecard and Strategy Management](#).

**WebLogic server domain**

Contains Java components that are configured to participate in the servicing of SOAP, HTTP, and other forms of requests.

**WebLogic Scripting Tool (WLST)**

A command-line scripting interface that enables you to configure, manage, and persist changes to WebLogic Server instances and domains and to monitor and manage server run-time events.

**XML API**

See [Oracle BI Server XML API](#).



---

---

# Index

## A

---

### agents

- administering with SA System subject area, A-1
- configuration settings affecting, 1-4
- definition of, 1-1
- jobs, adding and modifying, 3-2
- modifying in Job Manager, 3-3
- property fields, 3-3
- rerunning, 3-3
- running in parallel, 1-2
- Scheduler, 1-1
- support in Scheduler, 1-2

## C

---

- canceling instances, 6-3
- case of logon names, A-6
- certification information, 0-viii
- class, Java SchedulerJobException, 5-6
- cluster
  - configuring Scheduler instances, 2-3
- CreateArray object method, 4-10
- custom script
  - configuring properties, 4-2, 4-4
  - copy results to file system example, 4-5
  - creating, 4-2

## D

---

- daily trigger type, 6-8
- data sources, setting up, 1-4
- database, back-end
  - Scheduler requirement, 1-2
- databases, supported, 0-viii
- DayOfWeek Enum values, 4-7
- DeregisterCancelCommand object method, 4-11
- DSN
  - configuring entry for SQL Server databases, 2-3

## E

---

- e-mail addresses, populating for agents, A-1
- ExitCode instance property, 6-4
- ExitCode script object property, 4-6
- extensions, Java
  - example, 5-7

- overview, 5-4

## G

---

- GetConfigurationValue object method, 4-11
- GetTempFileName object method, 4-11
- groups
  - in SA System subject area, A-2

## I

---

- IgnoreWebcatDeliveryProfiles configuration element, A-5
- installing and upgrading, 1-4
- InstanceID instance property, 6-4
- InstanceID script object property, 4-6
- instances
  - canceling, 6-3
  - ExitCode property, 6-4
  - InstanceID property, 6-4
  - job instances, managing, 3-4
  - JobID property, 6-3
  - Message property, 6-4
  - purging, 6-3
  - Scheduler job, 3-4
  - Status property, 6-3
  - viewing, 6-3
- interface, Java
  - SchedulerJavaExtension, 5-5
  - SchedulerJobInfo, 5-5

## J

---

- Java jobs
  - configuring config.xml to enable use by Scheduler, 5-3
  - example, 5-7
  - overview, 5-4
  - using, 5-1
- Java program
  - configuring to use with agents, 5-3
  - example of creating, 5-2
- JavaHost
  - configuration file, 5-3
- job action
  - definition of, 6-5

- Load Script from File field, 6-6
- Parameters (script) field, 6-6
- Script field, 6-6
- Script Type field, 6-6
- job failures, troubleshooting, 4-18
- job instances, managing, 3-4
- Job Manager
  - about, 6-1
  - agent jobs, adding and modifying, 3-2
  - File menu options, table of, 6-2
  - Instances menu options, table of, 6-3
  - job instances, managing, 3-4
  - Jobs menu options, table of, 6-2
  - menus, 6-1
  - overview, 3-1
  - Service Management menu options, table of, 6-2
  - shutting down, 6-2
- job scripts
  - day-of-month flags, 4-7
  - day-of-week flags, 4-7
  - definition of, 4-1
  - Delete Script When Job is Removed, 6-5
  - deleting at end of job, 6-5
  - Disabled field, 6-5
  - disabling, 6-5
  - locating, 1-3
  - month-of-year constants, 4-8
  - occurrence constants, 4-9
  - Script object, 4-1
  - Script object read-only properties, 4-5
  - Script object read/write properties, 4-6
  - Script-defined constants, 4-7
- job triggers
  - definition of, 6-6
  - recurrent trigger fields, 6-7
  - recurrent trigger types, 6-8
  - setting, 6-6
  - single-run triggers, 6-7
- JobID instance property, 6-3
- JobID script object property, 4-6
- jobs
  - about setting up, 6-1
  - adding or modifying, 3-2
  - automatic deletion after execution, 6-5
  - Delete Job When Done field, 6-5
  - Execute When Missed field, 6-5
  - job property configuration fields, 6-4
  - last run time and next run time fields, 6-5
  - Last Run Time field, 6-5
  - Maximum Run Time MS field, 6-5
  - number of instances running, 6-5
  - re-running, 3-3
  - Running Instance Count field, 6-5
  - scheduling and canceling, 6-2
  - time out setting, 6-5
- JScript and VBScript job failures, troubleshooting, 4-18

## L

---

- LaunchProcess object method, 4-12
- logon names, case of, A-6

## M

---

- managing job instances, 3-4
- memory requirements, 0-viii
- Message instance property, 6-4
- Message script object property, 4-6
- methods, running and canceling Java, 5-5
- minimum disk space, 0-viii
- Monthly by Date trigger type, 6-8
- Monthly by DOW trigger type, 6-9
- month-of-year constants, 4-8

## N

---

- names, case of logon, A-6
- nqJobDeleteScriptWhenDone script-defined constant, 4-8
- nqJobDeleteWhenDone script-defined constant, 4-8
- nqJobDisabled script-defined constant, 4-8
- nqJobExecuteWhenMissed script-defined constant, 4-8
- nqJobHasEndDate script-defined constant, 4-8
- nqJobNoFlags, 4-8
- nqJobScriptContainsPath script-defined constant, 4-8
- nqJobStartNow script-defined constant, 4-8

## O

---

- occurrence constants, table of values, 4-9
- OnError object event, 4-17
- operating systems, supported, 0-viii
- Oracle BI repository
  - mapping SA System subject area in, A-4

## P

---

- ParameterCount script object property, 4-6
- pausing and continuing the Scheduler, 6-2
- platforms, supported, 0-viii
- purging instances, 6-3

## R

---

- RCU, using to create Scheduler database
  - schemas, 2-1
- recurrent trigger
  - Begin Date, 6-7
  - End Date, 6-7
  - End Time, 6-7
  - fields related to, 6-7
  - Has End Date, 6-7
  - Interval in Minutes, 6-8
  - Maximum Concurrent Instances, 6-8
  - Set Start Time To Now, 6-7
  - Start Time, 6-7

- trigger types, 6-8
- RegisterCancelCommand object method, 4-12
- requirements, system, 0-viii
- running job, definition of, 6-3

## S

---

- S\_NQ\_ERR\_MSG, 1-3
- S\_NQ\_INSTANCE, 1-3
- S\_NQ\_JOB, 1-3
- S\_NQ\_JOB\_PARAM, 1-3
- SA System subject area, A-6
  - about, A-1
  - about group resolution, A-2
  - columns for, A-3
  - importing data from, A-4
  - managing the case of login names, A-6
  - setting configuration options for, A-5
  - setting up, A-2
- ScheduleJobDaily object method, 4-13
- ScheduleJobMonthlyDate object method, 4-13
- ScheduleJobMonthlyDOW object method, 4-14
- ScheduleJobNow object method, 4-15
- ScheduleJobOnce object method, 4-16
- ScheduleJobWeekly object method, 4-16
- Scheduler
  - adding jobs in Job Manager, 3-2
  - configuring database, 2-1
  - connection, opening and closing, 6-2
  - database schemas, 1-3
  - database schemas created in RCU, 2-1
  - databases supported, 1-2
  - Java Job properties, 5-4
  - Job Manager, role of, 3-1
  - modifying jobs in Job Manager, 3-2
  - overview, 1-1
  - pausing and continuing, 6-2
  - role of, 1-1
  - scheduling and removing jobs, 6-2
  - schema tables, 1-3
  - scripted jobs, 1-1
  - stopping, 6-2
  - table names, 2-4
  - tables, 1-3
  - tables and database support, 1-2
- Script object
  - overview, 4-1, 5-4
  - read-only properties, 4-5
  - read/write properties, 4-6
- script-defined constants
  - about, 4-7
  - day-of-month, 4-7
  - day-of-month flags, 4-7
  - day-of-week, 4-7
  - day-of-week constants, 4-7
  - job status flags, 4-8
  - JobFlagsEnum, 4-8
  - month-of-year, 4-8
  - occurrence flags, 4-9
  - severity, 4-7

- scripted jobs
  - definition of, 1-1
  - Scheduler, 1-1
- scripts, object methods, 4-9
- Severity script object property, 4-6
- Single-Run Triggers, 6-7
- SSL information, 1-4
- standalone script jobs, 4-1
- Status instance property, 6-3
- stopping the Scheduler, 6-2
- subject area, SA System, A-6
- supported installation types, 0-viii
- system requirements, 0-viii
- SystemSubjectArea configuration element, A-5

## T

---

- topics of interest in other guides, 1-4
- troubleshooting
  - JScript and VBScript job failures, 4-18

## U

---

- UpperCaseRecipientNames configuration element, A-6
- user profile options, populating with SA System subject area, A-1
- UserID script object property, 4-6

## V

---

- viewing instances, 6-3

## W

---

- weekly trigger type, 6-8
- Windows
  - Scheduler tables, 1-3
  - Scheduler and database support, 1-2

