



Oracle® Business Intelligence Scheduler Guide

Version 10.1.3.2

December 2006

Part Number: B31768-01

Copyright © 2006, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States 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, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Chapter 1: What's New in This Release

Chapter 2: General Information About Oracle BI Scheduler

What is Oracle BI Scheduler? 9

About iBot Support in Oracle BI Scheduler 10

About Database Support in Oracle BI Scheduler 10

About Oracle BI Scheduler Schemas 11

About the Oracle BI Scheduler Log File 11

Chapter 3: Oracle BI Scheduler Job Manager Menus

About Scheduler Job Parameters 13

About the Oracle BI Scheduler Job Manager 13

Toolbar Menus in Oracle BI Scheduler Job Manager 14

File Menu in Oracle BI Scheduler Job Manager 14

Service Management Menu in Oracle BI Scheduler Job Manager 15

Jobs Menu in Oracle BI Scheduler Job Manager 15

Instances Menu in Oracle BI Scheduler Job Manager 16

Instance Properties in Oracle BI Scheduler Job Manager 16

Configuration Dialog Box in Oracle BI Scheduler Job Manager 18

Oracle BI Scheduler Tabs in Job Manager 19

Mail Tabs in Oracle BI Scheduler Job Manager 24

iBots Tab in Oracle BI Scheduler Job Manager 27

Workflow Tab in Oracle BI Scheduler Job Manager 29

Java Extension Tab in Oracle BI Scheduler Job Manager 29

General Oracle BI Scheduler Job Properties 31

Job Actions in Oracle BI Scheduler Job Manager 33

Job Triggers in Oracle BI Scheduler Job Manager 35

Chapter 4: Configuring Oracle BI Scheduler Job Manager

Opening Oracle BI Scheduler Job Manager under Windows 39

Adding an Oracle BI Scheduler Job in Job Manager	40
Modifying an Oracle BI Scheduler Job in the Job Manager	40
Restoring or Changing Default Job Manager Values	41
Modifying an iBot Job in Oracle BI Scheduler	41
Re-Running a Job Instance	41
Managing Oracle BI Scheduler Job Instances	42
Viewing or Refreshing Oracle BI Scheduler Job Instances	42
Canceling Oracle BI Scheduler Job Instances	43
Purging Oracle BI Scheduler Job Instances	43
Triggering Workflow Tasks in Other Applications	44
Configuring Oracle BI Scheduler to Trigger Workflows	45
Third-Party Triggering of Oracle BI Scheduler Jobs	45

Chapter 5: Using Oracle BI Scheduler Job Scripts

Job Manager Script for Cache Clearance	47
Script for iBot Chaining	50
Configuring Custom Script Properties for Oracle BI Scheduler	51
Configuring Custom Script Properties for Oracle BI Delivers	52
Read-Only Oracle BI Scheduler Script Object Properties	53
Oracle BI Scheduler Read/Write Script Object Properties	54
Oracle BI Scheduler Script-Defined Constants	55
Severity Constants	55
DayEnum Constants	56
DayOfWeekEnum Constants	56
JobFlagsEnum Constants	57
MonthEnum Constants	57
OccurrenceEnum Constants	58
Oracle BI Scheduler Script Object Methods and Events	59
CreateArray Method	60
DeregisterCancelCommand Method	61
GetConfigurationValue Method	61
GetTempFileName Method	62
LaunchProcess Method	62
RegisterCancelCommand Method	62
ScheduleJobDaily Method	63
ScheduleJobMonthlyDate Method	64
ScheduleJobMonthlyDOW Method	65

ScheduleJobNow Method	67
ScheduleJobOnce Method	67
ScheduleJobWeekly Method	68
OnError Event	69

Chapter 6: Configuring Java Job Properties for Oracle BI Scheduler

Oracle BI Scheduler Java Jobs	71
Configuring Java Job Properties for Oracle BI Scheduler	71
Configuring Java Job Properties for Oracle BI Delivers	72
Oracle BI Scheduler Custom Java Program Package	73
SchedulerJavaExtension Interface	73
SchedulerJobInfo Interface	73
SchedulerJobException Class	75
Oracle BI Scheduler Java Extension Example	76

Index

1

What's New in This Release

Oracle Business Intelligence Enterprise Edition consists of components that were formerly available from Siebel Systems as Siebel Business Analytics Platform, with a number of significant enhancements.

The Oracle Business Intelligence Scheduler Guide is part of the documentation set for Oracle Business Intelligence Enterprise Edition. This guide contains information about using Oracle BI Scheduler with scripted jobs and with Oracle Business Intelligence Delivers and iBots. This guide contains new material and material that was previously published under the title *Siebel Business Analytics Scheduler Guide*.

Oracle recommends reading the Oracle Business Intelligence Enterprise Edition Release Notes before installing, using, or upgrading the Oracle BI Infrastructure. The Oracle Business Intelligence Enterprise Edition Release Notes are available:

- On the Oracle Business Intelligence Enterprise Edition CD-ROM.
- On the Oracle Technology Network at http://www.oracle.com/technology/documentation/bi_ee.html (to register for a free account on the Oracle Technology Network, go to <http://www.oracle.com/technology/about/index.html>).

What's New in Oracle Business Intelligence Scheduler Guide, Version 10.1.3.2

Table 1 lists changes described in this version of the documentation to support version 10.1.3.2 of the software.

Table 1. What's New in *Oracle Business Intelligence Scheduler Guide*, Version 10.1.3.2

Topic	Description
All	All topics have been updated with new product names.
Chapter 3, "Oracle BI Scheduler Job Manager Menus"	All Oracle BI Scheduler configuration tasks were moved to this chapter. (For configuration information specific to Oracle's Siebel Workflow application, see the documentation for that application.) Content was added describing all new Job Manager fields.
Chapter 4, "Configuring Oracle BI Scheduler Job Manager"	All Job Manager configuration tasks were moved to this chapter.

Table 1. What's New in *Oracle Business Intelligence Scheduler Guide*, Version 10.1.3.2

Topic	Description
Chapter 5, "Using Oracle BI Scheduler Job Scripts"	All script procedures and information were moved to this chapter.
"Instance Properties in Oracle BI Scheduler Job Manager" on page 16	<ul style="list-style-type: none"> ■ The ErrorMessage field was renamed to Message. ■ A new status, Warning, was added to the Status field. ■ A list of meanings for the status of the job instance was added.
"Re-Running a Job Instance" on page 41	Added a procedure describing how to re-run a job instance.
"Job Manager Script for Cache Clearance" on page 47	The script was updated.
"Oracle BI Scheduler Read/Write Script Object Properties" on page 54	<ul style="list-style-type: none"> ■ The ErrorMessage field was renamed to Message. ■ A new Object Property, Severity, was added.
Chapter 6, "Configuring Java Job Properties for Oracle BI Scheduler"	This chapter contains all content relating to Java Job properties. All content in this chapter is new.

2

General Information About Oracle BI Scheduler

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

- [“What is Oracle BI Scheduler?” on page 9](#)
- [“About iBot Support in Oracle BI Scheduler” on page 10](#)
- [“About Database Support in Oracle BI Scheduler” on page 10](#)
- [“About Oracle BI Scheduler Schemas” on page 11](#)
- [“About the Oracle BI Scheduler Log File” on page 11](#)

What is Oracle BI Scheduler?

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

- Scripted jobs.

Scripted jobs are set up and submitted using the Job Manager feature of the Oracle Business Intelligence Server Administration Tool. For example, a scripted job could periodically load Oracle BI Server usage statistics into a back-end database. For scripted jobs, Oracle BI Scheduler communicates with Oracle BI Server.

Oracle BI Scheduler supports two published Java interfaces and two scripting languages.

- Supported Java interfaces:
 - SchedulerJavaExtension.java
 - SchedulerJobInfo.java
- Supported script languages:
 - VBScript
 - JScript

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

- Unscripted jobs, called *iBots*.

iBots deliver reports and alerts to end users. iBots are configured and submitted for execution using Oracle Business Intelligence Delivers. Oracle BI Scheduler communicates with Oracle Business Intelligence Presentation Services for unscripted jobs. For more information, see *Oracle Business Intelligence Presentation Services Administration Guide*.

About iBot Support in Oracle BI Scheduler

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

Although use of the Job Manager is not required for iBot jobs, you can use the Job Manager to further configure an iBot. This action requires that you first configure Oracle BI Scheduler. For information, see [“Configuration Dialog Box in Oracle BI Scheduler Job Manager” on page 18](#).

For information about iBots, see the Oracle BI Delivers chapter in the *Oracle Business Intelligence Answers, Delivers, and Interactive Dashboards User Guide*, or to the online help for Oracle BI Delivers.

About Database Support in Oracle BI Scheduler

Oracle BI Scheduler uses a single commercial back-end database to store pertinent information about a job, its instances, and its parameters. Oracle BI Scheduler works with all supported databases for Oracle Business Intelligence Enterprise Edition. For information about supported databases, see *Oracle Business Intelligence Enterprise Edition System Requirements and Supported Platforms*.

The Scheduler service starts only if the back-end database satisfies the following conditions:

Back-End Database Requirement	Notes
Configured	There is a one-to-one relationship between the back-end database and Oracle BI Scheduler. Do not configure multiple Scheduler applications to use a single back-end database.
Operational	For information about the specific back-end databases supported by Oracle BI Scheduler, see <i>System Requirements and Supported Platforms for Oracle Business Intelligence Applications</i> .
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, the Scheduler service may not start in some cases.</p>

About Oracle BI Scheduler Schemas

The database schemas associated with Oracle BI Scheduler are located in the directory \$INSTALLDIR\server\Schema\. This directory holds several SQL scripts that can be used by major commercial databases to create the tables in Oracle BI Scheduler's schema.

The schemas include four tables:

- S_NQ_JOB, used to store information about scheduled jobs.
- S_NQ_INSTANCE, used to store information about job instances.
- S_NQ_JOB_PARAM, used to store information about job parameters.
- S_NQ_ERR_MSG, used to store information about job instances that do not complete successfully.

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 the *Oracle Business Intelligence Infrastructure Installation and Configuration Guide*.
- Schemas store path and file names to job scripts, providing easy updates to several jobs if they share a script.
- Oracle BI Scheduler tables are included in transactional (OLTP) databases of Oracle's Siebel Customer Relationship Management applications. If you have version 7.7 or later of a Siebel CRM operational application installed, you do not need to install Oracle BI Scheduler tables.

NOTE: Usage Tracking is where the system collects usage statistics on each logical query submitted to the server. If usage tracking is enabled in the configuration file NQSCONFIG.INI, which is installed in the in \$INSTALLDIR\server\config directory, Oracle BI Scheduler also uses the S_NQ_ACCT table. This table is independent of Oracle BI Scheduler. When usage tracking is enabled, the Oracle BI Server generates Usage Tracking data files. A sample JavaScript is provided with the Oracle Business Intelligence installation, 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 about usage tracking, see the *Oracle Business Intelligence Server Administration Guide*.

About the Oracle BI Scheduler Log File

Oracle BI Scheduler logs information about its startup and shutdown activities in the NQScheduler.log file, located in the directory \$INSTALLDIR\server\Log\. Entries are written in UTF-8 format.

You can use a text editor to view this file. Set your viewer to UTF-8 to display readable text.

3

Oracle BI Scheduler Job Manager Menus

This topic area describes the Job Manager feature of the Business Intelligence Server Administration Tool. It is the first part of the process of configuring the Scheduler from the Job Manager.

This topic area contains the following topics:

- [“About Scheduler Job Parameters” on page 13](#)
- [“About the Oracle BI Scheduler Job Manager” on page 13](#)
- [“Toolbar Menus in Oracle BI Scheduler Job Manager” on page 14](#)
- [“Instance Properties in Oracle BI Scheduler Job Manager” on page 16](#)
- [“Configuration Dialog Box in Oracle BI Scheduler Job Manager” on page 18](#)
- [“General Oracle BI Scheduler Job Properties” on page 31](#)
- [“Job Actions in Oracle BI Scheduler Job Manager” on page 33](#)
- [“Job Triggers in Oracle BI Scheduler Job Manager” on page 35](#)

About Scheduler Job Parameters

A Scheduler job has the following parameters:

- Properties
- Actions to perform
- An execution schedule that determines when it runs

About the Oracle BI Scheduler 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

Toolbar Menus in Oracle BI Scheduler Job Manager

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

- [“File Menu in Oracle BI Scheduler Job Manager” on page 14](#)
- [“Service Management Menu in Oracle BI Scheduler Job Manager” on page 15](#)
- [“Jobs Menu in Oracle BI Scheduler Job Manager” on page 15](#)
- [“Instances Menu in Oracle BI Scheduler Job Manager” on page 16](#)

File Menu in Oracle BI Scheduler Job Manager

[Table 2 on page 14](#) describes the File menu options.

Table 2. Job Manager File Menu Options

Command	Description
Open Scheduler Connection	Opens the Machine Name dialog box, where you specify the name of the machine running Oracle BI Scheduler. If Oracle BI Scheduler resides on the same machine as the Oracle BI Server Administration Tool, you can enter <code>Local</code> .
Close Scheduler Connection	Closes the Job Manager connection to Oracle BI Scheduler.
Configuration Options	Opens the Scheduler Configuration dialog box, where you specify the name of the machine running Oracle BI Scheduler. The Job Manager Configuration dialog box then appears, where you can set configuration options.
Exit	Shuts down the Job Manager and returns you to the Server Administration Tool. If you exit the Job Manager while a connection to Oracle BI Scheduler is still open, the connection closes.

Service Management Menu in Oracle BI Scheduler Job Manager

Table 3 on page 15 describes the Service Management menu options.

Table 3. 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. Sometimes you may 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 in the Scheduler tab of the Job Manager Configuration dialog box.
Continue Scheduling	Resumes Oracle BI Scheduler's regular execution.
Stop Service	Stops the Oracle BI Scheduler service.

Jobs Menu in Oracle BI Scheduler Job Manager

Table 4 on page 15 describes the Jobs menu options.

Table 4. 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.
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 displayed in the Job List in the right pane.

Instances Menu in Oracle BI Scheduler Job Manager

An Oracle BI Scheduler instance is a record that records information regarding a specific execution of an Oracle BI Scheduler job. [Table 5 on page 16](#) describes the Instances menu options.

Table 5. Job Manager Instances Menu

Command	Description
Cancel Instance(s)	Cancels the running job. When a job has been successfully cancelled, the job's status appears as "Cancelled."
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 iBot job, only the failed items will be delivered.
Refresh Instance List	Refreshes the instance information displayed in the Instance List in the Job Instance View pane.

Instance Properties in Oracle BI Scheduler Job Manager

[Table 6 on page 16](#) describes the properties of Job Manager instances.

Table 6. Job Manager Instance Properties

Field	Description
JobID	ID of the job associated with this instance.
InstanceID	Unique ID of this specific instance of the Job.

Table 6. Job Manager Instance Properties

Field	Description	
Status	The Status of the job instance can be one of the following:	
	■ Running	This is the same for iBot jobs and Script Jobs. If the instance is running, the status is running.
	■ Completed	iBot: The iBot instance is set to completed if we successfully deliver to all delivery devices of the iBot.
		Script: This is set according to the Severity property in the script. Refer to Table 18 on page 54 .
	■ Failed	iBot: The iBot instance is set to failed if we were not able to successfully deliver to any of the delivery devices of the iBot.
		Script: This is set according to the Severity property in the script. Refer to Table 18 on page 54 .
	■ Warning	iBot: The iBot instance is set to Warning if we were able to successfully deliver to some of the delivery devices but not all.
		Script: This is set according to the Severity property in the script. Refer to Table 18 on page 54 .
■ Cancelled	Canceling any instance from the job manager will set the status to cancelled.	
	■ Timed Out	If the job has a maximum run time, and the running time of the instance exceeds this, then the status of the instance will be set to timed out.
Begin Time	The day and time the scheduler initiated the job instance.	
End Time	The day and time the job scheduler completed the job instance.	
ExitCode	Exit code of the instance. Depending on the job type, this code can be interpreted differently.	
	■ iBot	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 there may be more than one device for each recipient of an iBot.
	■ Script	The ExitCode of an instance is set according to the ExitCode property in the script. The default is 0 (zero). Refer to Table 18 on page 54 .
Message	Text message containing any error information of the instance, warnings, or general messages about the instance execution.	

Configuration Dialog Box in Oracle BI Scheduler Job Manager

Under Windows platforms, Oracle BI Scheduler configuration options are set in the Job Manager Configuration dialog box, a feature of the Business Intelligence Server Administration Tool. To access the Job Manager Configuration dialog box, refer to [“Opening Oracle BI Scheduler Job Manager under Windows” on page 39](#).

NOTE: In UNIX, Oracle BI Scheduler configuration options are set using `schconfig.exe`, a console-based application. The configuration options in `schconfig.exe` are identical to those available in the Job Manager Configuration dialog box.

The Job Manager Configuration dialog box has a set of tabs, as shown in [Table 7 on page 18](#).

Table 7. Job Manager Scheduler Configuration Dialog Box Tabs

Tab Name	Described in Section. . .
Scheduler	“Oracle BI Scheduler Tabs in Job Manager” on page 19
Mail	“Mail Tabs in Oracle BI Scheduler Job Manager” on page 24
iBots	“iBots Tab in Oracle BI Scheduler Job Manager” on page 27
Workflow	“Workflow Tab in Oracle BI Scheduler Job Manager” on page 29
Java Extension	“Java Extension Tab in Oracle BI Scheduler Job Manager” on page 29

Oracle BI Scheduler Tabs in Job Manager

The Job Manager Scheduler tab is divided into secondary tabs for Oracle BI Scheduler access to, and configuration of, its back-end database; for general Oracle BI Scheduler behavior; and for secure sockets and clustering configuration. [Table 8 on page 20](#) describes fields in the Oracle BI Scheduler tabs.

Table 8. Job Manager Scheduler Tabs and Fields

Tab	Field	Description
Database —Connection Pool	Database Type	Oracle Scheduler back-end database type. Choose a type from the drop-down list.
	Call Interface	The type of call the connection to the database makes. The call you choose from the drop-down list determines the application programmer interface (API) used to access the data source. The types of call interfaces in the list varies depending on the database type selected.
	Data Source Name	Identifies the data source name (DSN) of the data source to use for the connection. The DSN must contain valid logon information for a data source. If the information is invalid, the database logon fails. For example, if the DSN of the Oracle BI Scheduler back-end database is ORCL8_Scheduler, you enter ORCL8_Scheduler in this field.
	User name	The user name that Oracle BI Scheduler uses to log on to the data source. The user name must have read and write permissions to the back-end database and must be the same as the table owner for databases that require fully qualified table names (for example, ORACLE.S_NQ_JOB).
	Password/Confirm Password	Password used to log on to the data source. For security, the password is encrypted in the registry.
	Timeout (Minutes)	Specifies the amount of time in minutes that a connection to the data source remains open after an operation completes. During this time, new operations use this connection rather than open a new one, up to the number specified for Maximum Connections. The time is reset after each completed connection request. Specify a value of 1 or greater. The default value is 60.
	Maximum Connections	Maximum number of database connections Oracle BI Scheduler can open concurrently. Specify a value of 1 or greater. When this limit is reached, the connection request waits until a connection becomes available. The default value is 5.

Table 8. Job Manager Scheduler Tabs and Fields

Tab	Field	Description
Database —Database Tables	Jobs	S_NQ_JOB is a database table used to store information about scheduled jobs. NOTE: For information about modifying the database table names, see the <i>Oracle Business Intelligence Applications Installation and Configuration Guide</i> .
	Instances	S_NQ_INSTANCE is a database table used to store information about job instances.
	Parameters	S_NQ_JOB_PARAM is a database table used to store information about job parameters.
	Messages	S_NQ_ERR_MSG is a database table used to store information about job instances that do not complete successfully.
General	Scheduler Script Path	Refers to the path where Oracle BI Scheduler-created job scripts are stored. In general, you should not add or remove scripts from this directory. By default, this field is set to \$OracleBI\Scripts\Scheduler where \$OracleBI is the location in which the Oracle BI software is installed.
	Default Script Path	Path where user-created job scripts (not iBots) are stored. If a filename is entered in the Script field when adding or modifying a job, Oracle BI Scheduler examines the contents of this directory for the specified file. However, if a full path is given in the Script field, this directory is not examined. By default, this field is set to \$OracleBI\Scripts\Common where \$OracleBI is the location in which the Oracle BI software is installed.
	Temporary File Path	Path that specifies where temporary files are stored during Oracle BI Scheduler execution.
	Bulk Fetch Buffer Size (bytes)	Used in the database gateways. Specifies the maximum size in bytes of a bulk fetch page for retrieving data from a data source. The default value is 33,792 bytes.
	Server Port Number	Defaults the Scheduler port number.
	Purge Old instances after X days	Specifies the number of days after which old job instances are deleted from the back-end database automatically. To prevent old job instances from being deleted automatically, set the value to 0 (zero). The default value is to delete after seven days.

Table 8. Job Manager Scheduler Tabs and Fields

Tab	Field	Description
	Purge DB every X minutes	<p>Oracle BI Scheduler updates the tables and flags the affected rows as deleted.</p> <p>NOTE: Oracle BI Scheduler does not actually issue SQL DELETE statements when jobs or instances are removed.</p> <p>After every X minutes (where X is defined as the value of this field), the actual SQL DELETE statements are issued. The default value is every 60 minutes.</p>
	Minimum Execution Threads	Specifies the minimum number of multiple threads in the Oracle BI Scheduler thread pool.
	Maximum Execution Threads	Specifies the maximum number of multiple threads in the Oracle BI Scheduler thread pool.
	Pause When Service Starts	Specifies that no jobs should execute when Oracle BI Scheduler starts. While Oracle BI Scheduler pauses, users can add, modify, and remove jobs. However, no jobs execute. Select Service Management > Continue Scheduling to continue with regular execution.
	Administrator Name	<p>Specify a user with administrator's privileges in this field to allow Oracle BI Scheduler and Oracle BI Presentation Services to log onto the Oracle BI Server as other users. Oracle BI Scheduler runs jobs on the Oracle BI Server and Oracle BI Presentation Server for users without storing the user's password. The user specified must be a Repository Administrator with the ability to log in as other users, and have basic access to the Presentation catalog (to access the shared folder).</p> <p>NOTE: For information about changing the administrator name, see the <i>Oracle Business Intelligence Infrastructure Installation and Configuration Guide</i> or the <i>Oracle Business Enterprise Edition Deployment Guide</i>.</p>
	Administrator Password/ Confirm Password	<p>Password for the administrator with privileges to log on as other users. The password is encrypted in the registry. You can change it by using the Job Manager (Windows) or schconfi g. exe (UNIX).</p> <p>NOTE: For information about changing the administrator name, see the <i>Oracle Business Intelligence Infrastructure Installation and Configuration Guide</i> or the <i>Oracle Business Enterprise Edition Deployment Guide</i>.</p>

Table 8. Job Manager Scheduler Tabs and Fields

Tab	Field	Description
Advanced	Use Secure Socket Layer	This box is unchecked by default. Select this box to enable the Secure Socket Layer (SSL) for the Scheduler. After selecting this box, you must enter the SSL information. NOTE: The SSL fields allow you to configure Scheduler to communicate via SSL. For more information about setting up and locating the SSL information to enter into these fields, see the <i>Oracle Business Enterprise Edition Deployment Guide</i> .
	SSL Certificate File Path	
	SSL Certificate Private Key File	
	SSL File Containing Passphrase	
	SSL Require Client Certificate	
	SSL Certificate Verification Depth	
	CA Certificate Directory	
	CA Certificate File	
	SSL Trusted Peer DNs	
	SSL Cipher List	
	Script Server Port	The port used for interprocess communication between the script processes and the Scheduler process. This port will only accept connections from the same machine on which Scheduler is running. The default port is 9707.
	Participant in Cluster	The box is unchecked by default. If you want the Business Intelligence Server to be part of a cluster and wait for a connection from the Cluster Controller Server to notify it, select this box. If you select this box, you must also specify a value in the Cluster Monitor Port field. NOTE: For information about clustering, see the <i>Oracle Business Enterprise Edition Deployment Guide</i> .
	Cluster Monitor Port	The port to which the Business Intelligence Server listens on the Cluster Controller Server. The default port is 9708.

Mail Tabs in Oracle BI Scheduler Job Manager

The Job Manager Mail tab is divided into secondary tabs for Oracle BI Scheduler for SMTP mail delivery configuration, and for secure sockets and clustering configuration. [Table 9 on page 25](#) describes fields in the Mail tabs.

Table 9. Job Manager Mail Tabs and Fields

Tab	Field	Description
General	From Display Name	Used in the SMTP From field as a meaningful substitution for the sender's address. The default is Oracle BI Delivers <iBots@defaultmailserver.com>.
	Sender Address	Email address on the SMTP Server used as the sender's reply-to address for all mail sent from Oracle BI Scheduler. The initial value is default tuser@default tmail server.com, which must be changed to reflect a valid email address. NOTE: If you want to indicate that email recipients should not reply, consider adding "no_reply@mycompany.com" or "do_not_reply@mycompany.com" to this field.
	SMTP Server	Name of the SMTP server that delivers the mail. The initial value is default tmail server.com, which must be changed to a valid server name to deliver mail. You can find this information in the email client settings
	SMTP Port	Port number for SMTP on the given server. Most mail servers use port 25 as the mail port. The default is port 25.
	Number of Retries Upon Failure	If Oracle Delivers fails to deliver an email, it tries to resend it. Number of Retries Upon Failure sets the number of retry attempts. The default is 1.
	Maximum Recipients	Prevents sending to more recipients than your mail server's limit. If the mailing list has more subscribers than the Maximum Recipient value, Oracle Delivers splits the list into smaller lists and sends an email for each list. The default is 0, which means unlimited.
	Use Bcc Instead of To	Puts the subscribers list on the BCC line instead of the To line. Prevents subscribers from seeing the names of the others on the list and accidentally (or maliciously) replying to the entire list. This box is checked (Yes) by default.

Table 9. Job Manager Mail Tabs and Fields

Tab	Field	Description
General (continued)	Authenticate against SMTP Server	Specifies that the SMTP Server requires authentication to send email to an address outside of its domain. When you select this field, fill in the Username and Password fields for an email user on the SMTP server. When this field is not selected, the Username and Password fields are not used.
	Username	When you select the option Authenticate against SMTP Server, Username specifies the user name for an email user on the specified SMTP server. If left blank, the email server does not authenticate any email from Oracle Delivers.
	Email Password/Confirm Password	When you select the option Authenticate against SMTP Server, these fields specify the user's password for the specified user name.
Advanced	Use Secure Socket Layer	This box is unchecked by default. Select this box to enable the Secure Socket Layer (SSL) for mail delivery. After selecting this box, you must enter the SSL information. NOTE: The SSL fields allow you to configure the SMTP mail server to communicate via SSL. For more information about setting up and locating the SSL information to enter into these fields, see the <i>Oracle Business Enterprise Edition Deployment Guide</i> .
	CA Certificate Directory	
	CA Certificate File	
	SSL Certificate Verification Depth	
	SSL Cipher List	

iBots Tab in Oracle BI Scheduler Job Manager

Oracle iBots are functionally a combination of data stored in Oracle BI Presentation Services and Oracle BI Scheduler. The fields in the Job Manager iBots tab describe the behavior of all iBots that run on a specific Oracle BI Scheduler. [Table 10 on page 27](#) describes each field in the iBots tab.

If you change any of the values in these fields, you can restore the default values as shown in the procedure shown in the topic [“Restoring or Changing Default Job Manager Values” on page 41](#).

Table 10. Oracle BI Scheduler Job Manager iBots Tab, Configuration Fields

Field	Description
OBI Presentation Server	<p>The machine name and port where the Oracle BI Presentation Services is running.</p> <p>The server name must be specified and optionally can be followed by a colon and port number. If you do not specify a port number, Oracle BI Scheduler uses the default Web server port number 9710. For example, if the Web Server runs on machine1 and port 8900, the value for this field is machine1:8900.</p> <p>If you are using clustered SAW servers, specify the machine names as a comma-separated list. For example: localhost:9710, saw1:9710, oracleBI:9710.</p>
Log Directory	<p>iBots can create log files if exceptional error conditions occur. iBot Log Directory specifies the directory where these files are saved. The directory must be accessible to the Oracle BI Scheduler server. In Windows, the default installation runs the service as a system account, which prevents Oracle BI Scheduler from writing to or reading from network directories. If you put script files on network shares, or your scripts access network shares, then Oracle BI Scheduler must be run as a network user.</p>
Purge files older than (days)	<p>Specifies the number of days after which old iBot logs are deleted automatically. To prevent old logs from being deleted automatically, set the value to 0 (zero). The default value is delete after seven days.</p>
Number of Global Tries	<p>A Web or mail server that has too many people logged on may reject new connections, including connections from Oracle BI Scheduler. To cope with such overload, an iBot retries the connection. This field sets the maximum number of tries to get global information about what to deliver and to whom before the iBot gives up.</p>
Minimum Global Sleep Seconds	<p>The minimum number of seconds that the iBot randomly sleeps after its connection is refused before it attempts to reconnect to the server to get global information about what to deliver and to whom.</p>
Maximum Global Sleep Seconds	<p>The maximum number of seconds that the iBot randomly sleeps after its connection is refused before it attempts to reconnect to the server to get global information about what to deliver and to whom.</p>

Table 10. Oracle BI Scheduler Job Manager iBots Tab, Configuration Fields

Field	Description
Number of Request Tries	After an iBot has received the global information, it issues a series of unique requests to the server for each user. Specifies the number of times Oracle BI Scheduler attempts to connect to the server to issue these requests.
Minimum Request Sleep Seconds	The minimum number of seconds that the iBot randomly sleeps after its connection is refused before it attempts to reconnect to the server to issue requests.
Maximum Request Sleep Seconds	The maximum number of seconds that the iBot randomly sleeps after its connection is refused before it attempts to reconnect to the server to issue requests.
Number of Delivery Tries	After a unique request has executed, the iBot tries to deliver the results to specified devices. Specifies the number of times Oracle BI Scheduler attempts to connect to the server to deliver the results.
Minimum Delivery Sleep Seconds	The minimum number of seconds that the iBot randomly sleeps after its connection is refused before it attempts to reconnect to the server to deliver results.
Maximum Delivery Sleep Seconds	The maximum number of seconds that the iBot randomly sleeps after its connection is refused before it attempts to reconnect to the server to deliver results.
Max Rows Times Columns	When iBots are chained, this value governs the size of filters passed between iBots. When you pass a filter to another iBot in a chain, Oracle BI Scheduler creates a union of the result sets for the Conditional Report for each personalized recipient. This report can grow very large in some cases (1000 users with 100 unique rows each with ten columns per report = 1,000,000 column values in the filter). The Oracle BI servers may not be able to handle such a large filter, so this field specifies the maximum number of rows*columns in the filter. The default is 10,000.
Debug Enabled	Select this to have Oracle BI Scheduler generate a log file for each iBot. This log file has useful logging messages when trying to diagnose a problem. This log file is stored in \$OracleBI\server\Log\iBots (where \$OracleBI is the location in which the Oracle BI software is installed).
Keep Error Log Files	Select this check box to generate an error log file for each iBot. This log file is created only when an iBot execution encounters errors and contains only error messages. The file is stored in \$OracleBI\server\Log\iBots (where \$OracleBI is the location in which the Oracle BI software is installed).

Workflow Tab in Oracle BI Scheduler Job Manager

This topic is part of [“Configuration Dialog Box in Oracle BI Scheduler Job Manager.”](#)

Oracle Business Intelligence Delivers can trigger applications workflows for Oracle’s Siebel Customer Relationship Management versions 7.7 or higher.

NOTE: Before this triggering occurs, you must activate the EAI and Workflow component groups on the Siebel Enterprise Server, following the component activation instructions in *Siebel System Administration Guide*.

For more information on setting up operational application workflows, refer to [Chapter , “Configuring Oracle BI Scheduler Job Manager”](#).

Part of the workflow triggering process requires using the controls on the Job Manager Workflow tab. [Table 11 on page 29](#) describes each field in the Workflow tab.

Table 11. Job Manager Workflow Tab, Configuration Fields

Field	Description
Server	The name of the Enterprise Server containing the workflows. Oracle BI Scheduler uses the HTTP connector of the server to trigger the workflow. For example, http://localhost.
System	The system name of the application in your environment. For example, iae_enu.
Service	The name of the HTTP service as defined in the system configuration file. For example, ANALYTICS.
User	The user name that Oracle BI Scheduler uses to log on to the Enterprise Server.
Workflow Password/ Confirm Password	Password for the user account.

Java Extension Tab in Oracle BI Scheduler Job Manager

This topic is part of [“Configuration Dialog Box in Oracle BI Scheduler Job Manager.”](#)

The Java Extension feature enables the end user to specify a custom action using a Java interface that runs in the existing Java Host process. The results of the query are passed to the Java program and are returned as formatted output to the iBot execution engine. The Custom Java program may be specified for both when iBot conditions are satisfied and when no records are returned. This functionality allows an end user to develop custom actions in a platform-neutral language to all supported platforms. For more information about attaching a custom Java program to an iBot, see the *Business Intelligence Answers, Delivers, and Interactive Dashboards User Guide*.

Oracle BI Scheduler must be configured to be aware of some of the Java Host instance properties, such as the host it is running on and the port it is listening to. [“Oracle BI Scheduler Job Manager Java Extension Tab Configuration Fields” on page 30](#) describes the fields in the Java Extension tab.

Table 12. Oracle BI Scheduler Job Manager Java Extension Tab Configuration Fields

Field	Description
Java Host Servers	<p>The default server is localhost, and the default local host port is 9810.</p> <p>More than one server and port can be specified. Enter a comma-separated list of Java Hosts in the following format:</p> <p>hostname: port[, hostname: port]</p> <p>NOTE: If Oracle BI Scheduler is configured to use SSL, the Java Job uses SSL too.</p>

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 13 on page 31](#) describes the general job properties.

In addition, [Table 14 on page 33](#) describes the job action properties, and [Table 15 on page 35](#) describes the recurrent job triggers.

Table 13. General Oracle BI Scheduler Job Properties

Field	Description
Name	Short, descriptive name for the job. This field also appears 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 also appears 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 the Oracle BI Presentation Services, the UserID needs to be a valid Oracle Business Intelligence user ID. This field also appears 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 specified in this field.
Maximum Run Time MS	Specifies the maximum number of milliseconds this job should run before it is canceled forcibly. If a job exceeds its run time, it fails with a time-out reason code. To prevent the job from timing out, set this field to 0 (zero). NOTE: One second equals 1,000 milliseconds.
Last Run Time	Display-only field that shows the last time this job began execution. This field also appears in the Job List display in the right pane of the Job Manager window.
Next Run Time	Display-only field that shows recurrent jobs and the next time this job executes. The trigger is used to determine this value.
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.

Table 13. General Oracle BI Scheduler Job Properties

Field	Description
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 Oracle BI 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 Scheduler application stops), and if the job's next run time was missed, the job runs after Oracle BI Scheduler comes back up. If you do not select this option, the job executes at the its next run time, defined by its trigger.
Delete Script When Job is Removed	If you select this option, when a job is removed its associated job script is also removed. If many jobs reference the same job script, this option should not be set.

Job Actions in Oracle BI Scheduler Job Manager

Table 14 on page 33 describes the job action properties available in the Add Job and Modify Job dialog boxes. 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 14. 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 referenced by the Script field. The fields that appear depend upon the type of script that you specify.
Script Contains File Name	(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 field. Click this option if the Script field contains a file name.
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, enter a file name in this field, such as TestConnect.js. If no path is given, Oracle BI Scheduler examines the directory referred to in the Default Script Path configuration value in the Scheduler tab of the Job Manager Configuration dialog. If a path is given, 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	(Java, only) The fully qualified implementation class for the Java program action.
Class Path (jar file)	(Java, only) The name of the jar file containing the Java program.
Additional Class Path(s)	(Java, only) Other jar files containing the utility classes and libraries that the Java program needs to run properly. These are coma separated jar files.
DSN (Data Source Name)	(NQCmd, only) The data source name that OBISch uses to connect to OBIS. NOTE: Impersonation is used with NQCmd. The value in the User ID is the user that OBISch tries to impersonate when connecting to OBIS.

Table 14. Job Manager Job Action Fields

Field	Description
SQL Input File	(NQCmd, only) 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 generated by the aggregate persistence feature.
Additional Command Line Parameters	(NQCmd, only) Parameters that are passed to NQCmd. Enter one parameter per line. For example: -o D:\foo\bar.txt

Job Triggers in Oracle BI Scheduler Job Manager

This topic is part of [“General Oracle BI Scheduler Job Properties.”](#)

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 a job performs when it executes.

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

Single-Run Triggers

Use the Trigger Type drop-down list to choose 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 specified in the Begin Date and Start Time fields, which become active when you choose Run Once. An error occurs if the given time is in the past. If you select the option Set Start Time To Now, this trigger is equivalent to the Run Now trigger.

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 15 on page 35](#).

Recurrent Trigger Types are described in [Table 16 on page 37](#).

Table 15. Job Manager Recurrent Trigger Fields

Field	Description
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 option Set Start Time to Now.
End Date	Specifies the date when the last recurrent interval is run. Becomes active when the option Has End Date is set. If no end date is set, 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 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 specified in this value. If this value is less than the Start Time value, 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 specified in Begin Date at 11:00 P.M. and continues until 2:00 A.M. on the following day.

Table 15. Job Manager Recurrent Trigger Fields

Field	Description
Has End Date	<p>If you select this option, specify an End Date. If you do not select this option, the job stays scheduled.</p> <p>NOTE: The schedule is perpetual; the job instance is not. If you restart Oracle BI Scheduler, the next runtime will be set as dictated by the job schedule. If an instance is running while you shut down Oracle BI Scheduler, it will be canceled.</p>
Set Start Time To Now	<p>If you select this option, the Begin Date and Start Time fields are ignored and their values are populated with Oracle BI Scheduler's current date and time.</p>
Interval in Minutes	<p>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.</p>
Maximum Concurrent Instances	<p>If a job executes every n minutes (from the Interval in Minutes field), a long-running job may 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.</p>

Recurrent Trigger Types

The recurrent trigger types available from the Trigger Type drop-down list are described in [Table 16 on page 37](#). The fields described pertain to all recurrent triggers. Depending on the trigger type you choose, additional options become active. The examples in the table illustrate how these additional options can be used.

Table 16. 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>

Table 16. Job Manager Recurrent Trigger Types

Trigger Type	Description and Example
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, 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>
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>

4

Configuring Oracle BI Scheduler Job Manager

This topic area describes how to configure Oracle BI Scheduler using the Job Manager feature in the Oracle BI Server Administration Tool. The tasks include the following:

- [“Opening Oracle BI Scheduler Job Manager under Windows” on page 39](#)
- [“Adding an Oracle BI Scheduler Job in Job Manager” on page 40](#)
- [“Modifying an Oracle BI Scheduler Job in the Job Manager” on page 40](#)
- [“Re-Running a Job Instance” on page 41](#)
- [“Restoring or Changing Default Job Manager Values” on page 41](#)
- [“Modifying an iBot Job in Oracle BI Scheduler” on page 41](#)
- [“Re-Running a Job Instance” on page 41](#)
- [“Triggering Workflow Tasks in Other Applications” on page 44](#)

About Adding iBots Jobs

The Oracle BI Server passes iBots to Oracle BI Scheduler. You cannot add iBot jobs using the Job Manager. iBots jobs are most commonly added through Oracle BI Delivers.

You can, however, modify iBots using the Modify Job dialog box. For more information, refer to the topic [“Modifying an iBot Job in Oracle BI Scheduler” on page 41](#) and the *Oracle Business Intelligence Presentation Services Administration Guide*.

Opening Oracle BI Scheduler Job Manager under Windows

The Job Manager is the interface to Oracle BI Scheduler. When the Job Manager is opened, the BI administrator can use it to connect to, configure, and start and stop Oracle BI Scheduler, add and manage jobs, and manage job instances.

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

To open the Job Manager Configuration dialog box in Windows

- 1 Open the Oracle BI Server Administration Tool.
- 2 Choose Manage > Jobs from the toolbar.
- 3 In the Job Manager window, choose File > Configuration Options.

- 4 In the Machine Name dialog box, enter the machine name where Oracle BI Scheduler is located, and then click OK.

The Job Manager Configuration dialog box appears.

Adding an Oracle BI Scheduler Job in Job Manager

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

To add an Oracle BI Scheduler job in the Job Manager

- 1 In the Job Manager, choose Jobs > Add New Job.
- 2 Enter the appropriate information in the dialog box.

Refer to the following sections for field descriptions:

- [“General Oracle BI Scheduler Job Properties” on page 31](#)
- [“Job Actions in Oracle BI Scheduler Job Manager” on page 33](#)
- [“Job Triggers in Oracle BI Scheduler Job Manager” on page 35](#)

Modifying an Oracle BI Scheduler Job in the Job Manager

You can modify Oracle BI Scheduler jobs 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, Oracle BI Scheduler creates a file with an SCS extension in the path defined by this field. However, Oracle BI Scheduler's back-end database does not actually store the job scripts. In general, do not add or remove scripts from the \$OracleBI\server\Scripts\Scheduler directory (where \$OracleBI is the location in which the Oracle BI software is installed).

NOTE: directory.

To modify an Oracle BI Scheduler job in the Job Manager

- 1 In the Job Manager, select the job you want to modify.
- 2 Select Jobs > Modify Job.
- 3 In the Modify Job dialog box, change the job properties.

For field descriptions, refer to [“General Oracle BI Scheduler Job Properties” on page 31](#).

Restoring or Changing Default Job Manager Values

If you change any of the values in the Job Manager Scheduler tab fields, you can restore the default values as shown in the following procedure.

To restore or change the default values in the Job Manager Scheduler tab fields

- To restore the default values distributed with Oracle BI Scheduler, click Defaults.
- To revert to the most recently used settings, click Revert.

Modifying an iBot Job in Oracle BI Scheduler

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

To modify an iBot job

- 1 In the Job Manager, select the iBot you want to modify.
- 2 Choose Jobs > Modify Job.
- 3 In the Modify Job dialog box, modify the iBot properties, as described in the following table.

NOTE: A default value in these fields indicates that the value specified in the iBots tab of the Job Manager Configuration dialog box is active.

iBot Property	Description
Web Server	Specifies the Web server this iBot contacts when it runs. Do not change this setting, because the iBot may 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.

Re-Running a Job Instance

You can re-run any type of job, including an iBot job that failed. When you re-run a failed iBot job, only the failed items will be delivered. For example, if there are 1,000 recipients of an iBot, but 200 of those recipients belonged to a group that does not have the proper report viewing permissions, the Administrator can correct the permission settings and then re-run the job. In this example, only the 200 users in the modified group would receive the iBot (the iBot would not be re-delivered to the other 800 users who successfully received the iBot).

To re-run a job instance

- 1 Open the Oracle BI Server Administration Tool in Online mode.
- 2 Choose Manage > Jobs from the toolbar.
The Job Manager window appears.
- 3 Navigate to the Instance list.
- 4 Locate the job instance that you want to re-run.
- 5 Right-click on the job instance and select Re-run Instance(s) Now.

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:

- [“Viewing or Refreshing Oracle BI Scheduler Job Instances” on page 42](#)
- [“Canceling Oracle BI Scheduler Job Instances” on page 43](#)
- [“Purging Oracle BI Scheduler Job Instances” on page 43](#)

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 under the topic [“Instances Menu in Oracle BI Scheduler Job Manager” on page 16](#).
 - Instances properties are described in the table under the topic [“Instance Properties in Oracle BI Scheduler Job Manager” on page 16](#).

Viewing or Refreshing Oracle BI Scheduler Job Instances

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

NOTE: In some environments, if numerous instances have run and instances have not been purged in some time, this process may take a few seconds.

To view Oracle BI Scheduler job instance information

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

- 2 Choose Manage > Jobs from the toolbar.

The Job Manager window appears.

- 3 Navigate to the Instance list.

- 4 Select a particular job instance and choose Instance > View Instance.

A description of the Instance properties shown in the Instance window is given in the table under the topic [“Instance Properties in Oracle BI Scheduler Job Manager” on page 16](#).

To refresh Oracle BI Scheduler job instances

- In the Instance List, choose Instances > Refresh Instance List.

Canceling Oracle BI Scheduler Job Instances

Registered cancel methods are described under the command [“RegisterCancelCommand Method” on page 62](#). The cancel event is issued to 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 In the Instance List, Select a particular job instance.
- 2 Choose Instances > Cancel Instance(s).

Purging Oracle BI Scheduler Job Instances

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

- [Using the Job Manager Instances List](#)
- [Using the Job Manager Purge Instances Window](#)

Using the Job Manager Instances List

The following method 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 Server Administration Tool in online mode.
- 2 Choose Manage > Jobs from the toolbar.
The Job Manager window appears.
- 3 Navigate to the Instance list.
- 4 Select the instances from the Instance List and then press Delete.

Using the Job Manager Purge Instances Window

The following method 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 Server Administration Tool in online mode.
- 2 Choose Manage > Jobs from the toolbar.
The Job Manager window appears.
- 3 Navigate to the Instance list.
- 4 Click the Purge Instance(s) icon on the toolbar or choose Instances > Purge Instances to open the Purge Instances window.
You can purge instances by JobID, by UserID, or by End Time.
If you choose the End Time method, all jobs with an End Time less than or equal to the given time are purged.
- 5 Choose the purge method to use.
- 6 Click OK when you are done to return to the Job Manager window.

Triggering Workflow Tasks in Other Applications

Operational applications, such as Oracle's Siebel Enterprise family of applications, can be made to work with Oracle BI Delivers so that iBots can trigger workflow tasks in those applications.

When integrating iBots with application workflows, each row of an analytic query results set invokes a workflow. The row and other nondynamic properties are passed as a Property Set that is particular to each workflow. Using the Advanced tab of Delivers, you can configure the property set by manually typing in the name for a workflow and matching it with a column.

NOTE: Some tasks may need to be performed in the operational application before iBots can trigger workflows on its behalf.

The process of configuring Delivers to launch workflows consists of the following tasks:

- 1 Configure the operational application server for workflow application with Oracle Delivers.
This configuration step depends on the operational application being deployed. In general, you created a named subsystem for Oracle Business Intelligence, add name these subsystem components to the HTTP services section of a configuration file.
- 2 Configure the Workflow Triggers.
Choose from one of the following methods.
 - ["Configuring Oracle BI Scheduler to Trigger Workflows" on page 45.](#)
 - ["Third-Party Triggering of Oracle BI Scheduler Jobs" on page 45.](#)

- 3 Create workflow triggers in the iBots.

See the topics on using Oracle BI Delivers in the *Oracle Business Intelligence Answers, Delivers, and Interactive Dashboards User Guide*.

Configuring Oracle BI Scheduler to Trigger Workflows

This procedure explains how to configure Oracle BI Scheduler to launch a workflow.

To configure Oracle BI Scheduler to trigger workflows

- 1 In the Server Administration Tool, open the Job Manager.
- 2 Choose File > Configuration Options.
The Job Manager Configuration window opens.
- 3 Click the Workflow tab.
- 4 Fill out the fields using the information in [“Workflow Tab in Oracle BI Scheduler Job Manager” on page 29](#).

Third-Party Triggering of Oracle BI Scheduler Jobs

Third-party applications and scripts can launch an Oracle BI Scheduler job from the command line. They can also change the Job Parameters for a single instance. This change simulates third-party-triggered iBots. The interface for the command line is:

```
saschi nvoke.exe -u <Admin Name>/<Admin Password> (-j <job id> |  
-i <iBot path>) [-m <machine name>[:<port>]] ([-r <replace parameter filename>] |  
[-a <append parameter filename>])
```

The required parameters <Admin Name> and <Admin Password> are the same as those configured for Oracle BI Scheduler where you invoke the job. You can invoke the job either by the job ID or by the iBot path. Optionally, you can specify a machine and port for Oracle BI Scheduler server. If this is omitted, the invoker uses localhost and 9705 respectively.

The invoker also takes an optional job parameter file. Depending on the mode you choose, the parameters configured in Oracle BI Scheduler are changed in one of the following ways by the options in the job parameter file:

- Replace the existing parameters, using the following syntax:
[-r <replace parameter filename>]
- Append to the existing parameters, using the following syntax:
[-a <append parameter filename>]
- When using a parameter file, employ the following rules:
 - Use only one parameter per line.

- Do not ignore white space because it may be custom script-dependent.

Using Replace Mode

In replace mode, the file can specify to leave some parameters as they are in Oracle BI Scheduler. To specify this for a specific line, enter `$SCH_DEFAULT$` on the line. This text string acts as a variable and replaces the `$SCH_DEFAULT$` text with the text from the original Oracle BI Scheduler parameter.

For example, if the original parameter is hello, the line

```
$SCH_DEFAULT$ world, $SCH_DEFAULT$ again
```

is changed to

```
hello world, hello again
```

If you use the `saschinvoke` command, make sure the job parameters are correct. The `saschinvoke` command does not test the parameters for correctness. A job may not properly execute if invalid parameters are passed to it.

5

Using Oracle BI Scheduler Job Scripts

The Oracle Business Intelligence 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.

Using Oracle BI, an end user specifies a custom script action to be executed as part of an iBot. This custom script is passed to the results of the iBot report. Custom processing is performed on these results, which are then returned to the iBot execution engine output.

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

The following topics contain detailed information on job script properties:

- [“Read-Only Oracle BI Scheduler Script Object Properties” on page 53](#)
- [“Oracle BI Scheduler Read/Write Script Object Properties” on page 54](#)
- [“Oracle BI Scheduler Script-Defined Constants” on page 55](#)
- [“Oracle BI Scheduler Script Object Methods and Events” on page 59](#)

Job Manager Script for Cache Clearance

This topic provides an example of setting up a script for Oracle BI Scheduler. Oracle BI Scheduler can be used for general purpose scripts that extend the functionality of Oracle Oracle Business Intelligence.

The script purgeSASCache.js is used to periodically purge all of the cache from the Oracle BI Server:

```
////////////////////////////////////  
// 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\Bi n\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 outputStream = fso.OpenTextFile(fileName, 1);
    var output = outputStream.ReadAll();
    outputStream.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;
}
```

Script for iBot Chaining

This topic provides an example of setting up a script for Oracle BI Scheduler. The following script copies the results of an iBot to another directory. It copies the temporary file containing the results of the Conditional Request to the iBot log directory. The JobID, InstanceID, and UserID are used in the file name to guarantee that the result sets will not overwrite each other with each execution of the iBot, for each user, or for other iBots that share this script.

The example script uses the following setup:

- The iBot log directory on Oracle BI Scheduler machine is \$OracleBI\server\Log\iBots (where \$OracleBI is the location in which the Oracle BI software is installed).
- The iBot is run as Administrator
- The Custom Script Properties are set according to the table in [“Configuring Custom Script Properties for Oracle BI Delivers” on page 52](#)
- The job ID is 101 (assigned by Oracle BI Scheduler)
- The instance ID is 1208 (assigned by Oracle BI Scheduler)

The output of this example, after the iBot is run, is a file on the Oracle BI Scheduler machine called D:\OracleBI\Log\iBots\101-1208-Administrator-iBotScript1.PDF. This file contains the results of the Conditional Request in PDF format.

For all scripts from chained iBots, 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) = iBot Result File Path
// Parameter(1) = Last Part of Output File Name (no path)
//
////////////////////////////////////////

var FS0 = new ActiveXObject("Scripting.FileSystemObject");

var fileName = GetConfigurationValue("Log Dir", "iBots") +
  "\\\" + JobID + "-" + InstanceID + "-" + UserID + "-" +
  Parameter(1);

var fooFile = FS0.CopyFile(Parameter(0), fileName, true);

```

Configuring Custom Script Properties for Oracle BI Scheduler

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

NOTE: The script has to exist on Oracle BI Scheduler server machine before you can configure the properties.

To configure custom Oracle BI Scheduler script properties

- 1 Set the custom properties according to the discussion in [“Job Actions in Oracle BI Scheduler Job Manager” on page 33](#).

For example, for the script `purgeSASCACHE.js`, use the values shown in the following table. To view an example of the `SASCACHE.js` script, see [“Job Manager Script for Cache Clearance” on page 47](#).

Field	Value or Setting
Script Type	JScript
Script Contains File Name	Checked
Script	<code>purgeSASCACHE.js</code>
Parameters:	
Parameter(0): User	Administrator
Parameter(1): Password	SADMIN

- 2 Click OK.

Configuring Custom Script Properties for Oracle BI Delivers

You set the script properties on the Advanced tab of an iBot in Oracle BI Delivers. Refer to the *Oracle Business Intelligence Presentation Services Administration Guide* for details. Refer also to [“Modifying an iBot Job in Oracle BI Scheduler” on page 41](#).

NOTE: The script has to exist on the Oracle BI Scheduler server machine before you can create the iBot. Create the custom script, and then create the iBot to call the script.

To configure custom Delivers script properties

- 1 In the screen, select the iBot, and click the Advanced tab.
- 2 In the Custom Script Properties window, enter the properties.

For example, for the script shown in “Script for iBot Chaining,” you use the values shown in the following table:

Field	Value or Setting
Filename	createResultfile.js
Type	JScript
Results	Pass results of Conditional Request as a script Pass results as PDF
Other Parameters	iBotScript1.PDF

- 3 Click OK.
- 4 Run the iBot.

This script is run after the Conditional Request of the iBot is run.

Read-Only Oracle BI Scheduler Script Object Properties

Oracle BI Scheduler supports the read-only script object properties shown in [Table 17 on page 53](#).

Table 17. Scheduler Read-Only Script Object Properties

Object Property	Description	Return Value	Syntax
JobID	Returns the job identification number associated with this instance.	long	
InstanceID	Returns the instance identification number associated with this instance.		
ParameterCount	Returns the number of job parameters associated with the job script.	long	
Parameter (index)	Returns a specific parameter 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. Refer to "RegisterCancelCommand Method" on page 62 .	script object	
UserID	Returns the user identification number associated with the instance.	string	

Oracle BI Scheduler Read/Write Script Object Properties

Oracle BI Scheduler supports the read/write script object properties shown in [Table 18 on page 54](#).

Table 18. 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 ErrorMessage property can be used to convey meaningful error information. Setting this value changes the Error Message field of a Job Instance. without stopping execution of the current Job Script.</p> <p>In Windows environments, if the JScript throw() method is called and this property has been set, 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, 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 (see “Severity Constants” on page 55). By default, it is set to nqSeverityInformation.	string
ExitCode	Sets or returns the Exit Code property associated with the instance. The default is 0 (zero). Refer to the description of ExitCode instance properties in “Instance Properties in Oracle BI Scheduler Job Manager” on page 16 .	long

Oracle BI Scheduler Script-Defined Constants

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

- [“Severity Constants” on page 55](#)
- [“DayEnum Constants” on page 56](#)
- [“DayOfWeekEnum Constants” on page 56](#)
- [“JobFlagsEnum Constants” on page 57](#)
- [“MonthEnum Constants” on page 57](#)
- [“OccurrenceEnum Constants” on page 58](#)

Severity Constants

This topic is part of [“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 19 on page 55](#) describes Severity values.

Table 19. 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 will be 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 will be 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 will be set to Failed.

DayEnum Constants

This topic is part of [“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 20 on page 56](#) describes DayEnum values.

Table 20. DayEnum Constant Values

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

DayOfWeekEnum Constants

This topic is part of [“Oracle BI Scheduler Script-Defined Constants.”](#)

The DayOfWeekEnum values are used with the scheduling functions to identify days in a week. [Table 21 on page 56](#) describes DayOfWeekEnum values.

Table 21. DayOfWeekEnum Constant Values

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

JobFlagsEnum Constants

This topic is part of [“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 22 on page 57](#) describes JobFlagsEnum values.

Table 22. JobFlagsEnum Constant Values

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

MonthEnum Constants

This topic is part of [“Oracle BI Scheduler Script-Defined Constants.”](#)

The MonthEnum values are used with the scheduling functions to identify months. [Table 23 on page 57](#) describes MonthEnum values.

Table 23. MonthEnum Constant Values

Value	Description
nqJanuary	January
nqFebruary	February
nqMarch	March
nqApril	April

Table 23. MonthEnum Constant Values

Value	Description
nqMay	May
nqJune	June
nqJuly	July
nqAugust	August
nqSeptember	September
nqOctober	October
nqNovember	November
nqDecember	December

OccurrenceEnum Constants

This topic is part of [“Oracle BI Scheduler Script-Defined Constants.”](#)

The OccurrenceEnum values are used with the scheduling functions to identify the occurrence of a given day. [Table 24 on page 58](#) describes OccurrenceEnum values.

Table 24. OccurrenceEnum Constant Values

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

Oracle BI Scheduler Script Object Methods and Events

This process is part of [“Using Oracle BI Scheduler Job Scripts.”](#)

Oracle BI Scheduler script object methods and events, with their usage, syntax, and arguments, are the following:

- [“CreateArray Method” on page 60](#)
- [“DeregisterCancelCommand Method” on page 61](#)
- [“GetConfigurationValue Method” on page 61](#)
- [“GetTempFileName Method” on page 62](#)
- [“LaunchProcess Method” on page 62](#)
- [“RegisterCancelCommand Method” on page 62](#)
- [“ScheduleJobDaily Method” on page 63](#)
- [“ScheduleJobMonthlyDate Method” on page 64](#)
- [“ScheduleJobMonthlyDOW Method” on page 65](#)
- [“ScheduleJobNow Method” on page 67](#)
- [“ScheduleJobOnce Method” on page 67](#)
- [“ScheduleJobWeekly Method” on page 68](#)
- [“OnError Event” on page 69](#)

CreateArray Method

This topic is part of [“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 pass the array object to Script methods that accept an array as arguments.

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

```
var i;  
var array1= CreateArray(2);  
for (i = 0; i < array1. Si ze; i++)  
{  
    array1(i) = i;  
}  
  
    array1. Resi ze(4);  
for (i = 2; i < array1. Si ze; i++)  
{  
    array1(i) = i;  
}  
  
var array2 = CreateArray(0, 1, 2, 3);  
for (i = 0; i < array2. Si ze; i++)  
{  
    i f (array1(i) != array2(i))  
        break;  
}
```

Arguments: Refer to [Table 25 on page 61](#) for CreateArray method arguments.

Return Value: Returns an Array object.

Table 25. 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.

DeregisterCancelCommand Method

This topic is part of [“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 do not need to call this method if the script was canceled.

Syntax: DeregisterCancelCommand

GetConfigurationValue Method

This topic is part of [“Oracle BI Scheduler Script Object Methods and Events.”](#)

Returns the value in Oracle Business Intelligence Scheduler configuration relative to Oracle BI Scheduler’s root registry entry.

Usage: GetConfigurationValue() returns the string value for a registry setting relative to Oracle Business Intelligence Scheduler. The configKey and subkeyPath strings must be identical to those in the registry.

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

Arguments: Refer to [Table 26 on page 61](#) for GetConfigurationValue method arguments.

Return Value: Returns a string value.

Table 26. 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 Oracle BI Scheduler’s root path.

GetTempFileName Method

This topic is part of [“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 that can be used to create a file. Files created in job scripts are not deleted automatically when the script terminates.

Syntax: tfname = GetTempFileName()

Return Value: Returns a string value.

LaunchProcess Method

This topic is part of [“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 wait is set to True, this method returns the exit code returned by the process.

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

Arguments: Refer to [Table 27 on page 62](#) for LaunchProcess method arguments.

Return Value: Returns a long value.

Table 27. 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.

RegisterCancelCommand Method

This topic is part of [“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, 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: Refer to [Table 28 on page 63](#) for `RegisterCancelCommand` method arguments.

Table 28. RegisterCancelCommand Method Arguments

Argument	Description
source	An object whose method is being registered.
methodName	A string that specifies the method name.
arguments	Optional arguments to be passed into the method.

ScheduleJobDaily Method

This topic is part of “[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: Refer to [Table 29 on page 63](#) for `ScheduleJobDaily` method arguments.

Table 29. 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 associated with the job (either VBScript or JScript).
script	A String that specifies the script code or path (if the <code>nqJobScriptContainsPath</code> flag is set) associated with the job.
startDate	A date value that specifies the date the job is activated.
startTime	A date value that specifies the time the job is activated.
endTime	A date value that specifies the time the job is deactivated.

Table 29. ScheduleJobDaily Method Arguments

Argument	Description
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 passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags associated with the job. For valid settings, refer to “JobFlagsEnum Constants” on page 57 . 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 the job is deactivated.

ScheduleJobMonthlyDate Method

This topic is part of [“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: Refer to [Table 30 on page 64](#) for ScheduleJobMonthlyDate method arguments.

Table 30. 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 associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) associated with the job.
startDate	A date value that specifies the date the job is activated.

Table 30. ScheduleJobMonthlyDate Method Arguments

Argument	Description
startTime	A date value that specifies the time the job is activated.
endTime	A date value that specifies the time 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 the job runs. For valid settings, refer to “DayEnum Constants” on page 56 .
whichMonths	An integer value that specifies the months the job runs. For valid settings, refer to “MonthEnum Constants” on page 57 .
parameters	(Optional) A string array of parameter values passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags associated with the job. For valid settings, refer to “JobFlagsEnum Constants” on page 57 . 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 the job is deactivated.

ScheduleJobMonthlyDOW Method

This topic is part of [“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: Refer to [Table 31 on page 65](#) for ScheduleJobMonthlyDOW method arguments.

Table 31. 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.

Table 31. ScheduleJobMonthlyDOW Method Arguments

Argument	Description
scriptType	A string that specifies the script type associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) associated with the job.
startDate	A date value that specifies the date the job is activated.
startTime	A date value that specifies the time the job is activated.
endTime	A date value that specifies the time 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 the job runs. For valid settings, refer to “DayEnum Constants” on page 56 .
whichDays	An integer value that specifies the days of the week the job runs. For valid settings, refer to “DayOfWeekEnum Constants” on page 56 .
whichMonths	An integer value that specifies the months the job runs. For valid settings, refer to “MonthEnum Constants” on page 57 .
parameters	(Optional) A string array of parameter values passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags associated with the job. For valid settings, refer to “JobFlagsEnum Constants” on page 57 . 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 the job is deactivated.

ScheduleJobNow Method

This topic is part of [“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: Refer to [Table 32 on page 67](#) for ScheduleJobNow method arguments.

Table 32. ScheduleJobNow 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 associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the <code>nqJobScriptContainsPath</code> flag is set) associated with the job.
parameters	(Optional) A string array of parameter values passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags associated with the job. For valid settings, refer to “JobFlagsEnum Constants” on page 57 . The default is <code>nqJobNoFlags</code> .
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.

ScheduleJobOnce Method

This topic is part of [“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: Refer to [Table 33 on page 67](#) for ScheduleJobOnce method arguments.

Table 33. 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.

Table 33. ScheduleJobOnce Method Arguments

Argument	Description
scriptType	A string that specifies the script type associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) associated with the job.
startDate	A date value that specifies the date the job is activated.
startTime	A date value that specifies the time the job is activated.
parameters	(Optional) A string array of parameter values passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags associated with the job. For valid settings, refer to "JobFlagsEnum Constants" on page 57 . 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.

ScheduleJobWeekly Method

This topic is part of ["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: Refer to [Table 34 on page 68](#) for ScheduleJobWeekly method arguments.

Table 34. 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 associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) associated with the job.
startDate	A date value that specifies the date the job is activated.
startTime	A date value that specifies the time the job is activated.
endTime	A date value that specifies the time the job is deactivated.

Table 34. ScheduleJobWeekly Method Arguments

Argument	Description
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 the job runs. Refer to “DayOfWeekEnum Constants” on page 56 for valid settings.
parameters	(Optional) A string array of parameter values passed to the script. The default is an empty array.
flags	(Optional) A long value that specifies the flags associated with the job. For valid settings, refer to “JobFlagsEnum Constants” on page 57 . 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 the job is deactivated.

OnError Event

This topic is part of [“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 proper 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 if you want to perform some cleanup activities before the script terminates, such as deleting temporary files and releasing resources.

Syntax: OnError

Examples

■ Using VBScript:

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

■ Using JScript:

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

6

Configuring Java Job Properties for Oracle BI Scheduler

Oracle BI Scheduler Java Jobs

The Oracle BI Scheduler integrates with Oracle BI Java Host to support a custom Java program. The Oracle BI Scheduler publishes two Java interfaces, SchedulerJavaExtension and SchedulerJobInfo, as well as one Java class, SchedulerJobException.

NOTE: For more information about the Oracle BI Java Host, see the *Oracle Business Intelligence Presentation Services Administration Guide*.

Based on the Java interfaces and Java class from Oracle BI, an end user can implement the SchedulerJavaExtension interface. The compiled class can be executed as part of an iBot by setting up the iBot to use the name, path, and additional utility paths information of the compiled class.

This section contains the following topics:

- [“Configuring Java Job Properties for Oracle BI Scheduler” on page 71](#)
- [“Configuring Java Job Properties for Oracle BI Delivers” on page 72](#)
- [“Oracle BI Scheduler Custom Java Program Package” on page 73](#)
- [“Oracle BI Scheduler Java Extension Example” on page 76](#)

Configuring Java Job Properties for Oracle BI Scheduler

Use the following procedure to modify the Oracle BI Scheduler Java Job properties.

NOTE: The compiled Java class file has to exist on the Oracle BI Java Host machine before you can configure the properties.

To configure the Oracle BI Scheduler Java Job properties

- 1 Access the Job Manager and click Jobs > 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 properties, see [“Job Actions in Oracle BI Scheduler Job Manager” on page 33](#).

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
Class File (Jar File)	Sample

- 4 Click OK.

Configuring Java Job Properties for Oracle BI Delivers

Use the following procedure to set the Java Job properties on the Advanced tab of an iBot in Oracle BI Delivers. Before you perform this task, confirm that the compiled Java class file exists on the Oracle BI Java Host machine.

For more information, see [“Modifying an iBot Job in Oracle BI Scheduler” on page 41](#) and the *Oracle Business Intelligence Presentation Services Administration Guide*.

To configure an Oracle BI Delivers Java Job properties

- 1 Access the Delivers application and select the iBot to which you want to add the custom Java program.

- 2 Select the Advanced tab.

- 3 Click the Add Action Button and select Custom Java Program.

The Custom Java Program Properties window appears.

- 4 Enter the properties.

Example values and settings for a Java Job with the class name "test.Sample", class path "Sample", and no additional paths and parameters are included below.

Field	Value or Setting
Class Name	test.Sample
Class Path	Sample
Results	Pass Delivery content to Java program

- 5 Click OK.

6 Run the iBot.

The custom Java Program is run and delivery content is sent to the custom Java program.

Oracle BI Scheduler Custom Java Program Package

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

- [“SchedulerJavaExtension Interface” on page 73](#)
- [“SchedulerJobInfo Interface” on page 73](#)
- [“SchedulerJobException Class” on page 75](#)

SchedulerJavaExtension Interface

Note the following class of the SchedulerJavaExtension interface:

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

This interface has two methods: run and cancel. You can implement the interface and plug the compiled class file into the Java Host instance as part of iBot execution. The following table describes the methods:

Method	Description
run	This method is invoked by the Java Host. It takes one SchedulerJobInfo object (described below), which contains instance related properties like user ID, Job ID, and Instance ID and parameters. In addition, 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. User can imbed the resources that clean up the routine.

SchedulerJobInfo Interface

Note the following structure of the SchedulerJobInfo interface:

```

package com.siebel.analytics.scheduler.javahostpccalls;
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 hasResultSetSet();
    File getResultSetSetFile();
    String userID();
    int getExitCode();
    void setExitCode(int exitCode);
    int getStatus();
    void setStatus(int status);
    String getMessage();
    void setMessage(String message);

    void appendMessage(String message);
}

```

This interface is implemented by the parameter of the *run* method of any class implemented by SchedulerJavaExtension interface.

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 methods declared in the interface:

Method	Description
int jobID()	Returns the job ID associated with the iBot.
int instanceID()	Returns the instance ID associated with the iBot.
int parameterCount()	Returns how many parameters are associated with the iBot.
String parameter(int index)	Returns the indexed parameter for the iBot.(1).

Method	Description
boolean hasResultSet()	Specifies if there is a result set for this iBot.
File getResultSetFile()	Returns a file of result set for this iBot (2).
String userID()	Returns the user running the iBot.
int getExitCode()	Returns the exit code for the iBot.
void setExitCode(int exitCode)	User can set the exit code for the iBot.
int getStatus()	Returns the status code for the iBot.
void setStatus(int status)	User can set the status code for the iBot.
String getMessage()	Returns the message associated with the iBot.
void setMessage(String message)	User can set the message associated with the iBot. It replaces the existing message.
void appendMessage(String message)	User can append an additional message to the iBot.

SchedulerJobException Class

Note the following class of the SchedulerJobException.

```
package com.siebel.analytics.scheduler.javahostpccalls;
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. This class is a wrapper of error conditions. The following table describes the three members:

Members	Description
int m_exitCode	The framework would assign this exit code to the iBot.
int m_status	The framework would assign this status code to the iBot.
String m_message	The framework would assign this message to the iBot.

Oracle BI Scheduler Java Extension Example

The following example illustrates how to use the previously described interfaces and class to create a custom implementation.

NOTE: When the compiled class is run as part of an iBot, it collects the user who ran the iBot, job ID of the iBot, instance ID of the iBot, as well as all possible parameters into an output file.

```
package sample;

import java.io.*;
import java.lang.Thread;

import com.siebel.analytics.scheduler.javahostpccal.s.SchedulerJavaExtension;
import com.siebel.analytics.scheduler.javahostpccal.s.SchedulerJobException;
import com.siebel.analytics.scheduler.javahostpccal.s.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 ()
{
}
}
```


Index

Symbols

Job Manager Configuration dialog box
29

A

Administrator Name configuration field 22
Administrator Password/Confirm Password configuration field 22
Advanced Mail configuration tab 26
Advanced Scheduler configuration tab application 23 9

B

Bulk Fetch Buffer Size (bytes) configuration field 21
Business Intelligence Scheduler
databases supported 10
overview 9

C

CA Certificate Directory configuration field 23, 26
CA Certificate File configuration field 23, 26
Call Interface configuration field 20
canceling instances 16
class, Java
SchedulerJobException 75
Cluster Monitor Port configuration field 23
concurrent connections, maximum number 20
CreateArray object method 60

D

daily trigger type 37
Data Source Name configuration field 20
Database Scheduler configuration tab 20
Database Type configuration field 20
database, back-end
bulk fetch buffer size 21
call interface type 20
configuration options 19
data source logon name 20
database type 20

default configuration options, restoring 41
job script path 21
logon username 20, 29
maximum concurrent connections field 20
password field 20
purging 21
Scheduler requirement 10
timeout specification 20

DayOfWeek Enum values 56
Default Script Path configuration field 21
DeregisterCancelCommand object method 61

E

ExitCode instance property 17
ExitCode script object property 54
extensions, Java
example 76
overview 71

G

General Mail configuration tab 25
General Scheduler configuration tab 21
GetConfigurationValue object method 61
GetTempFileName object method 62

I

iBots
default values, restoring 27
definition of 9
Job Manager Configuration fields 27
jobs, adding and modifying 39
Keep Error Log Files configuration field 28
Log Directory configuration field 27
log file location 27
Max Rows Times Columns configuration field 28
Maximum Delivery Sleep Seconds configuration field 28
Maximum Global Sleep Seconds configuration field 27
Maximum Request Sleep Seconds configuration field 28
Minimum Delivery Sleep Seconds configuration field 28
Minimum Global Sleep Seconds configuration

- field 27
- Minimum Request Sleep Seconds configuration field 28
- modifying 41
- number connection attempts 27
- number delivery attempts 28
- Number of Delivery Tries configuration field 28
- Number of Global Tries configuration field 27
- Number of Request Tries configuration field 28
- Oracle BI Scheduler, role of 9
- property fields 41
- Purge files older than (days) configuration field 27
- rerunning 41
- sleep time 27
- Web Server Machine Name configuration field 27
- InstanceID instance property** 16
- InstanceID script object property** 53
- instances**
 - canceling 16
 - definition of 42
 - ExitCode property 17
 - InstanceID property 16
 - job instances, managing 42
 - JobID property 16
 - Message property 17
 - purging 16
 - Status property 17
 - viewing 16
- interface, Java**
 - SchedulerJavaExtension 73
 - SchedulerJobInfo 73
- J**
- Java Jobs**
 - example 76
 - overview 71
- job action**
 - Parameters (script) field 33
 - Script Contains File Name field 33
 - Script field 33
 - Script Type field 33
- job actions, defining** 33
- job instances, managing** 42
- Job Manager**
 - File menu options, table of 14
 - iBot jobs, adding and modifying 39
 - Instances menu options, table of 16
 - job instances, managing 42
 - Jobs menu options, table of 15
 - overview 39
 - Service Management menu options, table of 15
 - shutting down 14
- Job Manager Configuration dialog box**
 - accessing 18
 - iBots tab fields 27
 - Java Extension tab fields 29
 - opening 14
 - role of 18
 - Scheduler Tab fields 19
- job parameters, description of** 13
- job script path** 21
- job scripts**
 - day-of-month flags 56
 - day-of-week flags 56
 - definition of 47
 - Delete Script When Job is Removed 32
 - deleting at end of job 32
 - Disabled field 32
 - disabling 32
 - locating 11
 - month-of-year constants 57
 - occurrence constants 58
 - Script object 47
 - Script object read/write properties 54
 - Script object read-only properties 53
 - Script-defined constants 55
- job trigger, definition of** 35
- job triggers**
 - recurrent trigger fields 35
 - recurrent trigger types 37
 - setting 35
 - single-run triggers 35
- JobID instance property** 16
- JobID script object property** 53
- jobs**
 - about setting up 13
 - adding or modifying jobs 40
 - automatic deletion after execution 31
 - Delete Job When Done field 31
 - Execute When Missed field 32
 - job property configuration fields 31
 - last run time and next run time fields 31
 - Last Run Time field 31
 - Maximum Run Time MS field 31
 - number of instances running 31
 - re-running 41
 - Running Instance Count field 31
 - scheduling and canceling 15
 - time out setting 31

L

LaunchProcess object method 62
login username, back-end database 20

M**Mail tab**

Authenticate against SMTP Server
 configuration field 26
 From Display Name configuration field 25
 Maximum Recipients field 25
 Number of Retries Upon Failure configuration
 field 25
 Password configuration field 26
 Sender Address configuration field 25
 SMTP Port configuration field 25
 SMTP Server configuration field 25
 Use Bcc Instead of To field 25
 Username configuration field 26

managing job instances 42

Maximum Connections configuration
 field 20

Maximum Execution Threads configuration
 field 22

Message instance property 17

Message script object property 54

methods, Java

run and cancel 73

Minimum Execution Threads configuration
 field 22

Monthly by Date trigger type 38

Monthly by DOW trigger type 38

month-of-year constants 57

N

nqJobDeleteScriptWhenDone script-defined
 constant 57

nqJobDeleteWhenDone script-defined
 constant 57

nqJobDisabled script-defined constant 57

nqJobExecuteWhenMissed script-defined
 constant 57

nqJobHasEndDate script-defined
 constant 57

nqJobNoFlags 57

nqJobScriptContainsPath script-defined
 constant 57

nqJobStartNow script-defined
 constant 57

NQScheduler.log file 11

O

occurrence constants, table of values 58

OnError object event 69

Oracle BI Delivers
 Java Job properties 72

Oracle BI Scheduler
 Java Job properties 71
 role of 9

P

ParameterCount script object property 53

Participant in Cluster configuration
 field 23

Password/Confirm Password configuration
 field 20

Pause When Service Starts configuration
 field 22

pausing and continuing the Scheduler 15

Purge DB every X minutes configuration
 field 22

Purge Old instances after X days
 configuration field 21

purging instances 16

R**recurrent trigger**

Begin Date 35
 End Date 35
 End Time 35
 Has End Date 36
 Interval in Minutes 36
 Maximum Concurrent Instances 36
 Set Start Time To Now 36
 Start Time 35

recurrent triggers

fields related to 35
 trigger types 37

RegisterCancelCommand object
 method 62

running job, definition of 17

S

S_NQ_ERR_MSG 11

S_NQ_INSTANCE 11

S_NQ_JOB 11

S_NQ_JOB_PARAM 11

ScheduleJobDaily object method 63

ScheduleJobMonthlyDate object
 method 64

ScheduleJobMonthlyDOW object
 method 65

ScheduleJobNow object method 67

ScheduleJobOnce object method 67

ScheduleJobWeekly object method 68
Scheduler

- connection, opening and closing 14
 - execution threads 22
 - Job Manager, role of 39
 - pausing and continuing 15
 - Scheduler tables, about database support 10
 - scheduling and removing jobs 15
 - schema tables 11
 - schema tables, configuring 21
 - stopping 15
 - Scheduler configuration**
 - configuration options 18
 - Job Manager Configuration dialog box, opening 14
 - Scheduler Tab options 19
 - Scheduler Log File** 11
 - Scheduler Script Path configuration field** 21
 - Scheduler tables**
 - about 11
 - Script object**
 - overview 47, 71
 - read/write properties 54
 - read-only properties 53
 - Script Server Port configuration field** 23
 - script-defined constants**
 - about 55
 - day-of-month 56
 - day-of-month flags 56
 - day-of-week 56
 - day-of-week constants 56
 - job status flags 57
 - JobFlagsEnum 57
 - month-of-year 57
 - occurrence flags 58
 - severity 55
 - scripted jobs**
 - definition of 9
 - scripts**
 - object methods 59
 - Server Port Number configuration field** 21
 - Severity script object property** 54
 - Single-Run Triggers** 35
 - SSL Certificate File Path configuration field** 23
 - SSL Certificate Private Key File configuration field** 23
 - SSL Certificate Verification Depth configuration field** 23, 26
 - SSL Cipher List configuration field** 23, 26
 - SSL File Containing Passphrase configuration field** 23
 - SSL Require Client Certificate configuration field** 23
 - SSL Trusted Peer DNs configuration field** 23
 - Status instance property** 17
 - stopping the Scheduler** 15
- ## T
- Temporary File Path configuration field** 21
 - temporary files, location of** 21
 - Timeout (Minutes) configuration field** 20
- ## U
- Use Secure Socket Layer configuration field** 23, 26
 - User name configuration field** 20
 - UserID script object property** 53
 - username** 20
- ## V
- viewing instances** 16
- ## W
- weekly trigger type** 37
 - Windows**
 - Scheduler tables, about 11
 - Scheduler, about database support 10
 - Workflow tab**
 - logon username, back-end database 29
 - username 29
 - Workflow tab fields** 29