



Oracle Knowledge Analytics Administrator's Guide

Release 8.5

Document Number OKIS-ADEV85-01

November, 2012

COPYRIGHT INFORMATION

Copyright © 2002, 2012, Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT RIGHTS

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

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

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

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

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

Contents

About This Guide	1
In This Guide	1
Screen and Text Representations	2
References to External Web Content	2
Introduction to Oracle Knowledge Analytics	3
Analytics Architecture	3
Oracle Knowledge Applications	4
JMS Queue and Event Router	5
Data Warehouse	5
Oracle Data Integrator	5
Oracle Business Intelligence Enterprise Edition	5
Administrating Analytics	7
Configuring Analytics for Initial Use	7
Operating and Managing Analytics	7
Configuring Analytics	8
Defining Reporting User Groups	8
Importing Information Manager Reference Data	8
Applying Analytics Styles to OBIEE	9
Copying the Analytics Style Directories	9
Installing and Deploying Analytics Styles	9
Making the Analytics Styles Available to OBIEE	11
Using the Packaged Encryption Utility	11
Managing Users and Security	13
Managing Administration Users and Security	13
Managing Reporting Users and Security	13
Creating General Analytics Report Users	14
Configuring and Managing Data Acquisition	15
Configuring Applications to Send Events to Analytics	15
Validating Messaging for Application Instances	17
Validating Staging Table Data	18
Configuring and Managing Data Transformation	20
Analytics Event Processing	20
Checking for Unprocessed Events	21
Creating Event Batches for Transformation	21

Loading Dimension and Staging Event Data	22
Completing Daily Batch Processing	22
Processing Fact and Aggregate Data	22
Logging Processing Activity	22
Logging Processing Exceptions	23
Purging the ODI Execution Logs	23
Managing and Monitoring Event Processing	23
Starting Event Processing	23
Event Processing Components	24
Event Processing Control Properties	24
Event Processing Settings	25
Monitoring Event Processing and Recovering from Failures	25
Re-Processing Events	27
Analytics Data Aggregation	28
Monitoring Data Aggregation	28
Identifying Data Aggregation Exceptions	28
Weekly and Monthly Aggregation Tables	29
How Analytics Uses Aggregated Data in Reporting	29

Configuring and Managing Data Access and Storage 30

Viewing the Analytics Database Schema	30
Configuring and Managing Data Purging	30
Enabling and Scheduling an ODI Data Management Agent	31
Enabling the Agent	31
Specifying Agent Connectivity Information	31
Scheduling the Purge Process	32
The Default Purging Interval	34
Purge Settings	34
Modifying the Purging Process	35
Monitoring the Purge Process	35
Diagnosing and Resolving Purge Errors	35
Restoring Default Purge Settings	36

About This Guide

This guide provides information for application administrators who need to understand the basic architecture of the Oracle Knowledge Analytics application and perform the various tasks associated with configuring, deploying, and maintaining it. The guide describes the initial configuration processes as well as ongoing operational tasks.

In This Guide

This guide is divided into the following sections:

<i>Chapter 1, Introduction to Oracle Knowledge Analytics</i>	This section provides an overview of the Analytics application, its architecture and data flow, and describes the major application components.
<i>Chapter 2, Administrating Analytics</i>	This section provides information on the initial configuration tasks and ongoing operational and management tasks required to administrate an Analytics application.
<i>Chapter 3, Configuring Analytics</i>	This section describes required and optional Analytics configuration tasks, and also includes instructions for encrypting passwords to enable automated access to various application instances as required for deploying Analytics.
<i>Chapter 4, Managing Users and Security</i>	This section provides general information on managing security for administration functions and general reporting functions.
<i>Chapter 5, Configuring and Managing Data Acquisition</i>	This section describes configuration and administration processes that support Analytics data acquisition.
<i>Chapter 6, Configuring and Managing Data Transformation</i>	This section describes the configuration and administration processes that support Analytics data transformation.
<i>Chapter 7, Configuring and Managing Data Access and Storage</i>	This section describes the configuration and administration processes that support the Analytics data warehouse.

Screen and Text Representations

The product screens, screen text, and file contents depicted in the documentation are examples. We attempt to convey the product's appearance and functionality as accurately as possible; however, the actual product contents and displays may differ from the published examples.

References to External Web Content

For your convenience, we refer to Uniform Resource Locators (URLs) for resources published on the World Wide Web when appropriate. We attempt to provide accurate information; however, these resources are controlled by their respective owners and are therefore subject to change at any time.

Introduction to Oracle Knowledge Analytics

Oracle Knowledge Analytics is a business intelligence application that provides insight into the effectiveness and performance of Oracle Knowledge Intelligent Search and Information Manager implementations.

Analytics includes intuitive dashboards and packaged reports that provide insight into the most important aspects of search and content performance and user interaction.

Analytics features near-realtime data integration, easy end-user access to application data for creating custom reports, and a comprehensive set of reporting tools packaged within Oracle's Business Intelligence presentation environment.

You can use Analytics to:

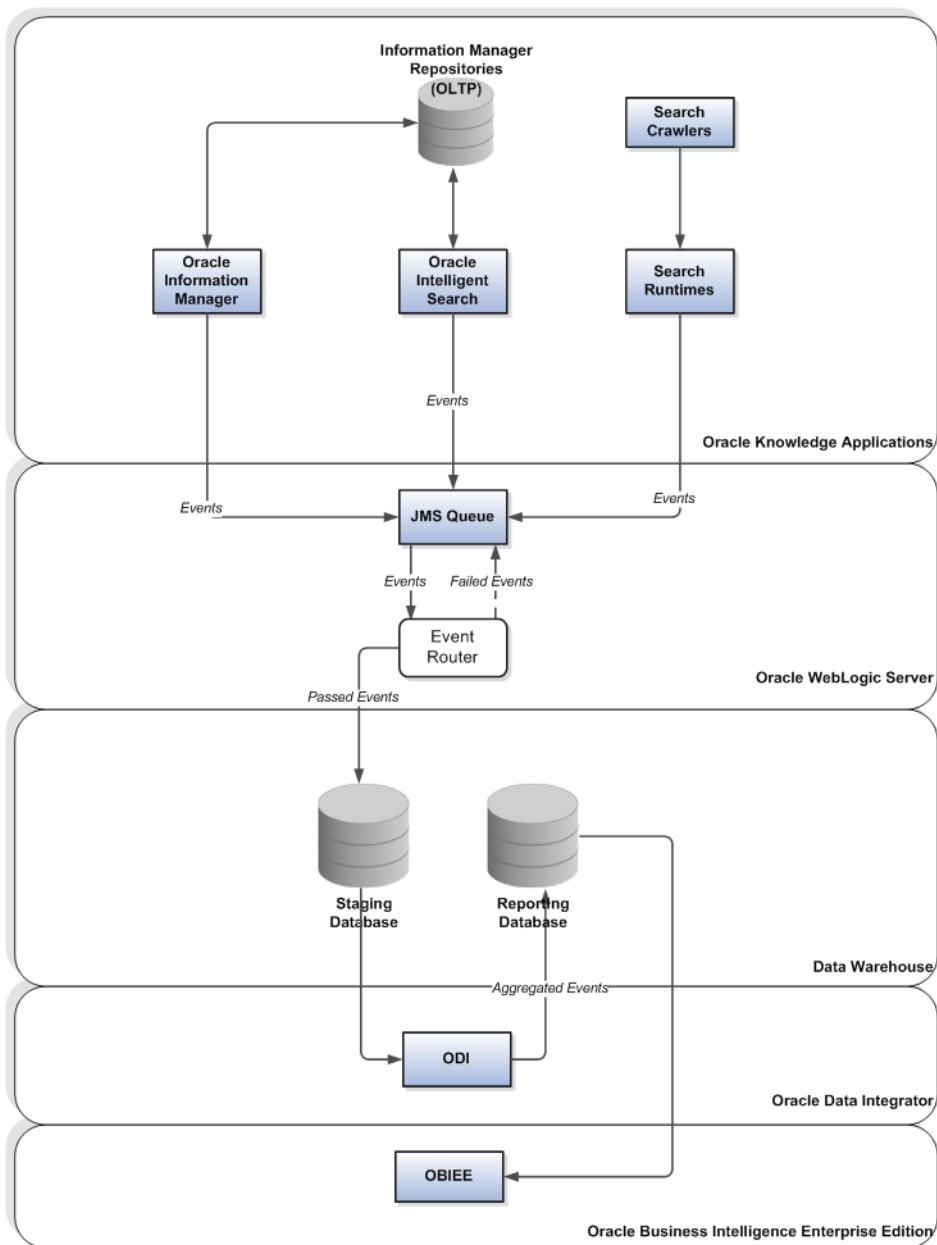
- understand user behavior, such as why users visit your site, and what they try to achieve
- assess the quality of Oracle Knowledge articles and answers, and determine whether Intelligent Search and Information Manager are effectively addressing users' needs
- determine if important information is missing from your application content

Important! Analytics requires installed and configured Oracle Knowledge software supplementary software as described in the Oracle Knowledge Installation Guide.

Analytics Architecture

An Analytics application consists of one or more instances of the following components, which together generate and process the various types of Intelligent Search and Information Manager user interactions, which are captured as events, transformed and stored as data for analysis, and accessed and presented within the business intelligence platform.

- Intelligent Search and Information Manager applications
- JMS Queue and Event Router installed on WebLogic Server
- Data Warehouse
- Oracle Data Integrator (ODI)
- Oracle Business Intelligence Enterprise Edition (OBIEE)



Oracle Knowledge Applications

Intelligent Search and Information Manager applications that are configured for Analytics generate events based on various user interactions, such as asking questions or publishing articles to the knowledge base. An Analytics client that is installed on each Intelligent Search and Information Manager instance generates the events and sends them to the Java Messaging Service (JMS) Queue.

JMS Queue and Event Router

You install the JMS Queue and Event Router in a WebLogic Server instance. The Event Router retrieves events in batches from the JMS Queue and delivers these events to the staging database.

For more information on configuring Oracle Knowledge applications for Analytics and configuring and managing the JMS Queue and Event Router, see “Configuring and Managing Data Acquisition” on page 15.

Data Warehouse

The data warehouse consists of a staging database and a reporting database. The staging database stores the raw data from incoming events, and the reporting database stores the data that the Data Integrator (ODI) transforms for reporting purposes, including weekly and monthly aggregated data.

For more information on configuring and managing the Analytics data warehouse, see “Configuring and Managing Data Access and Storage” on page 30.

Oracle Data Integrator

Oracle Data Integrator (ODI) transforms the staged event data and transforms it into reporting data. The transformation process includes calculating metrics and populating facts and dimensions in the reporting database.

The installation process installs and configures packages within ODI that control transformation and other data management tasks, such as deleting (purging) obsolete data.

Note: For more information on configuring and managing data transformation, see “Configuring and Managing Data Transformation,” on page 20.

You must install and configure an instance of ODI separately from the Oracle Knowledge Analytics installations. ODI is not included in the Oracle Knowledge product distribution.

Oracle Business Intelligence Enterprise Edition

Oracle Knowledge Analytics uses Oracle Business Intelligence Enterprise Edition to present Analytics data for analysis. The installation process defines and configures OBIEE features, including packaged reports, that enable you to begin using Analytics immediately.

Note: Analytics packaged reports use implementation-specific data dimensions, such as channels, locales, users, and workflows that you define for your specific business environment. You add this dimensional data to the database by importing it from the Information Manager application that you have configured for Analytics.

You can also use additional OBIEE capabilities to create custom analyses, dashboards, scorecards, key performance indicators (KPIs), and other reporting features.

For more information on working with Analytics packaged reports and additional OBIEE capabilities, see the *Analytics User's Guide* and the OBIEE product documentation set on the Oracle Technology Network.

Note: You must install and configure an instance of OBIEE separately from the Oracle Knowledge installations. OBIEE is not included in the Oracle Knowledge product distribution.

Administering Analytics

Analytics administration requires initial configuration tasks as well as ongoing operational and management tasks.

Configuring Analytics for Initial Use

You must configure some Analytics functionality prior to using the application. Initial configuration tasks include:

- applying the Analytics style definitions to the OBIEE user interface as described in “Applying Analytics Styles to OBIEE” on page 9
- defining users to access Analytics reports as described in “Creating General Analytics Report Users” on page 14
- defining Reporting User Groups as described in “Defining Reporting User Groups” on page 8
- importing lookup (reference) data from Information Manager as described in “Importing Information Manager Reference Data” on page 8
- enabling and scheduling the ODI data management agent to perform data purging as described in “Configuring and Managing Data Purging” on page 30

Operating and Managing Analytics

You perform ongoing operational and management tasks using the processes described below:

Operational and Management Tasks	Description
Chapter 4, Managing Users and Security	This section provides general information on managing security for administration functions and general reporting functions.
Chapter 5, Configuring and Managing Data Acquisition	This section describes configuration and administration processes that support Analytics data acquisition.
Chapter 6, Configuring and Managing Data Transformation	This section describes the configuration and administration processes that support Analytics data transformation.
Chapter 7, Configuring and Managing Data Access and Storage	This section describes the configuration and administration processes that support the Analytics data warehouse.

Configuring Analytics

This section describes required and optional Analytics configuration tasks, and also includes instructions for encrypting passwords to enable automated access to various application instances as required for deploying Analytics.

Defining Reporting User Groups

Analytics can report on groups of Information Manager users. Analytics uses specific user groups that you define within the configured Information Manager application.

Reporting User Groups are independent of any other groups to which an Information Manager user may belong. A user can belong to only one Reporting User Group.

You define Reporting User Groups using the Information Manager Management Console.

For more information on the using the Information Manager Management Console to define Reporting User Groups, see the *Information Manager Administration Guide*.

Importing Information Manager Reference Data

You must populate the dimensional data that Analytics uses, such as the information about the channels, content, locales, recommendations, user groups, users, workflow, and workflow steps that are currently defined in your environment. You make this data available to Analytics by:

- importing the reference data from the Information Manager database to the Analytics staging database
- resetting the event timestamp of the imported dimensional data to the current system time and date

Note: Analytics uses this data to populate only dimensional data in the reporting database; it does not use time-related attributes.

To import the Information Manager reference data:

- log into the Oracle Information Manager Administrator console using a repository administrator or a super-user role
- select Configure under System in the Tools tab
- select the Current Configuration option
- select Create Analytics Lookup Data

The import process begins importing the data into the staging database. The import process creates events in the staging database related to channels, content, locales, recommendations, user groups, users, workflow, and workflow steps, which are then available for ODI to process and move to the reporting database for use in Analytics.

Note: The data import process cannot import data older than 10 years. If your Information Manager database contains data older than 10 years, the import process will ignore this data.

To reset the reference data timestamp:

- execute the following SQL to reset the timestamp to SYSDATE of all the reference data or look-up events that you import in Analytics:

```
UPDATE DW_STAGE.DW_STG_DATA SET TIMESTAMP = Trunc(SYSDATE);
```

The SQL resets the event timestamp to the current system data and time.

Applying Analytics Styles to OBIEE

Analytics requires a specific set of styles and formats to display the reports as they are designed to be viewed. You configure and deploy the Analytics styles by:

- copying the Analytics style directories from the installation location to the deployment location
- installing and deploying Analytics styles on the application server
- make the Analytics styles available to OBIEE
- activating and validating the new configuration

Copying the Analytics Style Directories

The Analytics installation process creates `S_OracleKnowledge` and `SK_OracleKnowledge` directories at the location you specify in the *Choosing the OBIEE Components Location* step during the installation. These directories contain the style and formatting information required for viewing Analytics reports on OBIEE.

- copy the `S_OracleKnowledge` and `SK_OracleKnowledge` directories to the following location of the OBIEE instance:

```
<OBIEE_instance_installation_directory>\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips1\analyticsRes
```

Installing and Deploying Analytics Styles

You install and deploy the Analytics styles as an application within WebLogic Server. You install and deploy applications using the WebLogic Administration Console.

Note: For more information on working with WebLogic Administration Console, see the *Oracle Fusion Middleware Administrator's Guide* and other WebLogic Server documentation on the Oracle Technology Network.

To install and deploy the styles:

- start the WebLogic Server Administration Console at:

`http://hostname:port/console`

- select Deployments in the Domain Structure pane

WebLogic Server displays the Summary of Deployments page.

- select Lock and Edit in the Change Center
- select the Install option

WebLogic Server displays the Install Application Assistant page.

- specify the path to the S_Oracle Knowledge directory using the following fields:

	specify the path to the parent of the AnalyticsRes directory where you copied the S_OracleKnowledge and SK_OracleKnowledge directories, for example:
Path	<code><OBIEE_instance_installation_directory>\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips1</code>
Current Location	ensure that the server name and pecified name and the specified path are correct

The Install Application Assistant displays a list of eligible applications.

- select the AnalyticsRes directory, which contains the S_OracleKnowledge and SK_OracleKnowledge directories, then select Next

The Install Application Assistant prompts you to choose the targeting style.

- select the Install this deployment as an application option
- select Next

If you have server clusters defined in your environment, the Install Application Assistant prompts you to choose a deployment target:

- select the appropriate server in the Clusters section

Important! In single-server environments, the Install Application Assistant will not display the Clusters section; you do not need to select a server.

- continue the installation process by accepting the defaults on the subsequent screens until the Install Application Assistant displays the following option:

I will make the deployment accessible from the following location

- select this option, and ensure that the Location field displays the correct path:

`C:\OBIEE\instances\instance1\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips1\analyticsRes`

- select Finish

The Install Application Assistant displays the deployed application, which is started with the status OK.

Making the Analytics Styles Available to OBIEE

You must make the styles that you have deployed available to the OBIEE presentation server by editing the OBIEE configuration. To edit the OBIEE configuration:

- edit the `InstanceConfig.xml` file at the following location:

```
C:\OBIEE\instances\instance1\config\OracleBIPresentationServicesComponent\coreapplication_obips1
```

- add the following markup to the `InstanceConfig.xml` file, within `<ServerInstance>` and `<WebConfig>` tags.

```
<URL>

<CustomerResourcePhysicalPath>C:\OBIEE\instances\instance1\bifoundation\OracleBIPresentationServicesComponent\coreapplication_obips1\analyticsRes</CustomerResourcePhysicalPath>

<CustomerResourceVirtualPath>/analyticsRes</CustomerResourceVirtualPath>

</URL>

<UI>

<DefaultStyle>OracleKnowledge</DefaultStyle>

<DefaultSkin>OracleKnowledge</DefaultSkin>

</UI>
```

Using the Packaged Encryption Utility

You can create encrypted passwords to use when you need to configure automated access to various applications within the Oracle Knowledge environment using the encryption utility that is available in the ICE environment.

To create an encrypted password:

- open a terminal window and change to the instances folder
- execute the command to start ICE

```
./setenv.sh
```

```
setenv.bat
```

- enter the encryption command, using your password as the argument to the command:

```
./encrypt.sh <your_password>
```

```
encrypt.bat <your_password>
```

The encryption utility produces an encrypted form of the password, for example:

c6ypWN0kqs7qoM0V58qif5zi

- copy the encrypted password for use.

Managing Users and Security

You can manage users and security for both administration and general users of Analytics. This section provides general information on managing security for administration functions that require access to WebLogic Server and ODI, and managing security for general reporting users who require access to OBIEE.

You must define general reporting users so that they have the privileges associated with the default BI Consumer role, as described in “Creating General Analytics Report Users” on page 14.

Managing Administration Users and Security

You can manage users, roles, and privileges for Analytics administrators and other users who need access to WebLogic Server and ODI to install, configure, or deploy Analytics.

For information on managing security for WebLogic Server, see <cite> Oracle Fusion Middleware Securing Oracle WebLogic Server.

For information on managing security for Oracle Data Integrator, see <cite> Managing Security in Oracle Data Integrator in *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

Managing Reporting Users and Security

You can manage users, roles, and privileges for general Analytics users who need to use OBIEE to report on the performance of Oracle Intelligent Search and Oracle Information Manager applications using the OBIEE security facility. Oracle Business Intelligence 11g is tightly integrated with the Oracle Fusion Middleware Security architecture and delegates core security functionality to components of that architecture. You can use the following tools, as required, to configure security in Oracle Business Intelligence:

Security Tools	OBIEE Security Components
Oracle WebLogic Server Administration Console	Embedded WebLogic LDAP Server (Users and Groups)
Oracle Fusion Middleware Control	Credential Store (Credentials) Policy Store (Application Roles)
Oracle BI Administration Tool	Repository RPD (Permissions)
Oracle BI Presentation Catalog Administration Page	Oracle BI Presentation Catalog (Presentation)

For more information on security architecture, default users, groups, and application roles, see the *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

Creating General Analytics Report Users

OBIEE uses groups, roles, and users for user security and privileges. A standard installation of OBIEE creates default administrator, author, and consumer groups and roles:

Role	Description
BI Administrator	defines default privileges for all members of the BI Administrator group
BI Author	defines default privileges for all members of the BI Author group
BI Consumer	defines default privileges for all members of the BI Consumer group

Oracle Knowledge Analytics reports and supporting data objects, such as dimensions and measures, are created using the BI Consumer role. You need to define users so that they have the privileges associated with the BI Consumer role in order to grant them complete access to the packaged reports and the Analytics data objects. Users having the BI Consumer role will have complete access to the packaged reports, and will be able to use Analytics data to create custom reports and leverage other OBIEE features for use with Analytics.

Configuring and Managing Data Acquisition

This section describes the configuration and administration processes that support Analytics data acquisition. You configure and manage data acquisition for Analytics by:

- configuring Oracle Knowledge applications to send events to the Analytics application
- validating the data flow among the JMS Queue, Event Router, and staging database tables

Configuring Applications to Send Events to Analytics

You configure applications to send events to Analytics by configuring the JMS Queue and Event Router. You install and configure the JMS Queue and Event Router in an instance of WebLogic server as part of the Analytics installation process.

You can configure Intelligent Search and Information Manager instances to send event data to Analytics as part of the installation process, or as a separate post-installation process.

Note: You use the same process to configure Event Messaging for both Information Manager and Intelligent Search applications; however, the location of the `ok_jms.properties` file is different.

When you install Oracle Information Manager and Oracle Intelligent Search applications, the installer prompts you to specify whether to capture events for use by Analytics. To configure an instance for Analytics during installation, you must have an installed and configured Analytics instance, and access to the JMS configuration information.

If you specify to enable events for Analytics, the installer sets the property `is_enabled` in the `ok_jms.properties` file to `true`; the installer then prompts you to specify the JMS Queue properties. These properties determine how the events generated by Oracle Knowledge applications are logged in the Oracle WebLogic JMS queue.

If you specify to not enable events for Analytics during installation, the installation process sets the property `is_enabled` in the `ok_jms.properties` file to `false`; the installer does not collect the JMS Queue properties, and events are not written to the JMS queue;

however, all the required components of the analytics client are still installed for each application.

You can enable or disable logging for an installed instance by editing the `is_enabled` property in the `ok_jms.properties` file.

Configuring the JMS Queue in Intelligent Search and Information Manager

You configure the JMS Queue in an Intelligent Search or Information Manager instance by editing the configuration property settings in the `ok_jms.properties` file. The `jms.properties` file is located at:

`<Intelligent_Search_Install_Directory>/conf`

or

`<Information_Manager_Install_Directory>/lib`

To configure the JMS Queue:

- edit the following properties:

Property	java.naming.security.principal
Parameter	JMS_USERID
Configuration	Specify an encrypted user name to connect to the WebLogic JMS Queue. The user name must be encrypted using the encryption utility within ICE as described in "Using the Packaged Encryption Utility" on page 11.
Property	java.naming.security.credentials
Parameter	JMS_PASSWORD
Configuration	Specify an encrypted password. The password must be encrypted using the encryption utility within ICE as described in "Using the Packaged Encryption Utility" on page 11.
Property	java.naming.provider.url
Parameter	JMS_URL
Configuration	Specify the URL required to connect to the WebLogic JMS queue. For example: <code>t3://<Analytics_Server_Name>:8220</code>
Property	ok.queue.connection.factory.name
Parameter	JMS_FACTORY_NAME
Configuration	Specify the connection factory name created in WebLogic. The Analytics installer creates the connection factory using the name <code>jms/AnalyticsConnectionFactory</code> by default.

Property	ok.queue.name
Parameter	JMS_QUEUE_NAME
Configuration	Specify the name of the JMS queue. The Analytics installer creates the queue using the name jms/AnalyticsQueue by default.

Property	ok.analytics.logging.enabled
Parameter	true
Configuration	Specify the value of logging.enabled as true to enable event generation from this instance.

- save the file and restart the application to activate the configuration changes

Validating Messaging for Application Instances

You can verify that each configured application is sending the events to the JMS Queue. You verify messaging for an instance by:

- stopping the Event Router
- monitoring the JMS Queue to determine whether events are collecting in the queue
- re-starting the Event Router

Stopping the Event Router

To stop the Event Router:

- start the WebLogic Server Administration console that hosts the Analytics Event Router and JMS queue at:

`http://<hostname>:<port>/console`

where:

`<hostname>` is the DNS name or IP address of the Administration Server

`<port>` is the address of the port on which the Administration Server is listening for requests (7001 by default)

- select Deployments in the Domain Structure pane in the left portion of the screen
- select AnalyticsEventRouterEA-8.5.0.0 deployment
- select the option When Work Completes from the Stop drop-down menu
- check that the status of the AnalyticsEventRouterEA-8.5.0.0 deployment is now pending

Monitoring the JMS Queue

You can test and monitor the JMS Queue to determine whether configured instances are sending messages to the queue. You test the queue by:

- initiating an event from within a configured application
- monitoring the JMS queue from the WebLogic Administration console

To initiate an event:

- log onto a configured application instance to start a session, and optionally ask a question, or view an article

Note: Each of these activities will generate events.

- log onto the WebLogic Administration console
- select Service from the Domain Structure menu, then select Messaging and Modules
- select the SystemModule-OracleKnowledgeModule item
- select the Queue-AnalyticsQueue item
- select the Monitoring tab

The Message Count displays the messages that are currently in the queue.

- select the checkbox for the queue, then select the Show Messages button

The Message Count displays the list of messages that arrived in the queue. You can select each message to view details.

If messages are not arriving in the queue, validate the settings in the `ok_jms.properties` file and verify the connection from the configured instance to the address and port of the JMS Queue.

When you have validated messaging for the instance, restart the Events Router.

Restarting the Event Router

To restart the Event Router from the WebLogic Administration console:

- select Deployments in the Domain Structure pane in the left portion of the screen
- select the AnalyticsEventRouterEA-8.5.0.0 deployment checkbox
- select Start - Servicing All Requests
- verify that the status of the AnalyticsEventRouterEA-8.5.0.0 deployment is Active.

Validating Staging Table Data

You can validate that the Event Router is sending data to the staging tables for processing by ODI.

To validate the staging table data:

- connect to the DW_STAGE schema on the analytics database using an SQL Client.
- execute the following query to return a list of all events in the staging tables:

```
Select * from dw_stage.dw_stg_data;
```

- compare the event types and timestamps of the events in the staging tables with those in the JMS queue

If the events are not present in the staging database, verify the Connection Pool settings.

Verifying Connection Pool Settings

If you are unable to validate the staging table data by observing that the Event Router is sending data to the staging tables, verify the Connection Pool settings for the jdbc/AnalyticsDataSource as follows:

- log into the WebLogic Administrator console
- select Data Sources in the Domain Structure pane in the left portion of the screen
- select the jdbc/AnalyticsDataSource item
- select the Connection Pool tab
- verify the connection information

Configuring and Managing Data Transformation

This section describes the configuration and administration processes that support Analytics data transformation. Analytics uses Oracle Data Integrator (ODI) to transform dimension and event data from the staging database format to the reporting database format, and also to aggregate daily data into weekly and monthly totals for use in reports.

ODI performs the following tasks as part of Analytics operation:

- transforming staged dimension and event data into reporting data
- aggregating daily data into weekly and monthly data for reporting

ODI transforms and aggregates data using packages that are installed and configured with Analytics. You configure and manage data transformation for Analytics by:

- managing event processing as described in “Analytics Event Processing” on page 20
- managing data aggregation as described in “Analytics Data Aggregation” on page 28

Analytics Event Processing

Analytics uses an ODI package, `PKG-ORACLE_KNOWLEDGE_MAIN` to automate data transformation and additional supporting processes. When configured and started, the event processing package runs continuously to:

- check for unprocessed events as described in “Checking for Unprocessed Events” on page 21
- create event batches for transformation as described in “Creating Event Batches for Transformation” on page 21
- load transformed dimension and staging data into the reporting database as described in “Loading Dimension and Staging Event Data” on page 22

When all events for the current day have been processed as described in “Completing Daily Batch Processing” on page 22, it loads fact and aggregate data into the reporting database as described in “Processing Fact and Aggregate Data” on page 22. In addition, the event processor also:

- logs information about each batch and records the batch number associated with each event in the staging database, as described in “Logging Processing Activity” on page 22
- logs any event processing exceptions as described in “Logging Processing Exceptions” on page 23
- purges ODI execution logs after a specified time period as described in “Purging the ODI Execution Logs” on page 23

Checking for Unprocessed Events

Event processing starts when you start the ODI event processing package, `PKG-ORACLE_KNOWLEDGE_MAIN`, as described in “Starting Event Processing” on page 23.

The event processing package operates continuously, and queries the staging database to determine whether there are unprocessed events associated with the current date. If there are no unprocessed events, the process will wait for the interval of time specified for the `SLEEP_INTERVAL` property, as described in “Event Processing Control Properties” on page 24.

Creating Event Batches for Transformation

The event processor processes events in batches. Each time that it queries the staging database, it evaluates unprocessed events against a set of criteria to determine whether it will create an event batch for processing.

The package evaluates the session data that have been loaded into the staging database, and creates an event batch for processing as follows:

If...	And...	Then...
the number of unprocessed events is less than the minimum batch size...	all unprocessed events occurred within the current processing day...	the package adds the event to the current batch, and waits for the specified sleep interval, then checks the staging table for more events to be processed. See “Event Processing Control Properties” on page 24 for more information on the sleep interval.
the number of unprocessed events is less than the minimum batch size	the current processing day has ended, which is indicated by the presence of one or more events timestamped within the next processing day...	the package ignores the minimum batch size criteria and creates a batch containing all of the unprocessed events for the current day.
the number of unprocessed events is equal to or greater than the minimum batch size...	all unprocessed events occurred within the current processing day...	the package creates a batch containing up to the maximum (<code>BATCHSIZE</code>) number of events.

Ensuring Session Integrity in Reporting Data

The event processing package ensures that all events from a single session are processed within the same batch by evaluating each event timestamp against the value of the buffer time property.

The package excludes all events within sessions that contain any events more recent than the value of the (`BUFFER_TIME`) property. The buffer time is set by default to 30 minutes. For example, an event that occurred at 12:10:00 will be excluded from a batch created at 12:30:00, since the time elapsed between the event and the current time is less than the buffer time. The `BUFFER_TIME` and `BATCH_MIN_SIZE` properties help to ensure that event processing does not separate events from a single session into multiple batches.

Loading Dimension and Staging Event Data

The event processing package pre-processes the staged dimension and event data, including calculating and aggregating activity and fact data. It then loads the finalized data into the reporting database.

Completing Daily Batch Processing

When the event processing package completes a batch, it checks for the next available events in the staging database. If no sessions and events are available within the current processing day, it checks for available data for the next processing day (`CURRENT_BATCH_PROCESS_DATE + 1`).

If sessions and events exist for the next day, the package:

- updates the `INTERFACE_PACKAGE_PROCESS` table as described in “The Interface Package Process Table” on page 25
- stops querying the staging database for new events
- processes facts and aggregates weekly and monthly data as described in “Processing Fact and Aggregate Data” on page 22

Processing Fact and Aggregate Data

When all available events have been processed for the current day, the package suspends event processing, and processes fact data and weekly and monthly aggregate data for the current day. When the package completes all facts and aggregate data processing for the current day, it updates the `INTERFACE_PACKAGE_PROCESS` table as described in “Logging Processing Activity” on page 22.

Logging Processing Activity

The event processing package records processing activity at the batch and event level. Each time it creates a batch, the process:

- creates a record of the batch in the `INTERFACE_BATCH_PROCESS` table
- identifies the batch as the first batch of the day in the `INTERFACE_PACKAGE_PROCESS` table, if applicable

Each time it completes a batch, the event processing package executes an ODI procedure named `UPDATE_INTERFACE_BATCH_PROCESS_SUCCESS`. The procedure updates the staging database for each processed event with:

- a record that the event was processed (`IS_PROCESSED`)
- the processing date and time (`DATE_PROCESSED`)
- It also records the batch completion time and other details in the `INTERFACE_BATCH_PROCESS` table

If the batch is the final batch for the processing day, the process records the start and end times for the data load process in the `INTERFACE_PACKAGE_PROCESS` table.

When the package completes all fact and aggregate data processing for the current day, it finalizes (publishes) the data so that it is available for use in reporting, and records the report start and completion times (REPORT_PUBLISH_START_TIME and REPORT_PUBLISH_COMPLETION_TIME) in the INTERFACE_PACKAGE_PROCESS table.

Logging Processing Exceptions

The event processing package starts an ODI procedure, UPDATE_INTERFACE_BATCH_PROCESS_EXCEPTION, which records any errors that occur within the event processing interfaces, packages, or procedures as exceptions in the INTERFACE_PACKAGE_PROCESS table and the DW_STG_DATA table. The process records the batch number and the Exception ID for each exception that occurs during event processing. See “Monitoring Event Processing and Recovering from Failures” on page 25 for more information.

Purging the ODI Execution Logs

The event processor also purges the ODI execution logs. The event processing package runs the ODI execution log purge process daily after it loads fact and aggregate data. The execution log purging process is controlled by the ODI_LOG_PURGE_DURATION property, which is set in the ANALYTICS_PROPERTIES table.

The purging process uses an internal utility, OdiPurgeLog, to delete all execution logs that are older than the specified purge duration interval. The purge interval is set to 7 days by default.

We recommend that you validate the execution log purging process by reviewing the execution logs in the ODI Operator tab. For more information on purging ODI logs, refer to the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

Managing and Monitoring Event Processing

Analytics processes events using components that you install and configure as part of the standard installation process. You manage event processing by starting the process, which then operates automatically using the various components and configuration settings described in this section.

Starting Event Processing

You start event processing by executing the PKG-ORACLE_KNOWLEDGE_MAIN package within ODI. Once the process starts, it runs continuously. To start the event processing package:

- log onto to ODI as a SUPERVISOR
- navigate to the Designer tab
- expand the DW_Data_Loads project
- select PKG-ORACLE_KNOWLEDGE_MAIN from the Packages folder

- right-click the package and select **Execute** from the options

Note: You can stop event processing using the process described in “Stopping Event Processing” on page 27

Event Processing Components

This section provides reference information about the various components that Analytics uses to operate, track, and control event processing.

Component	Description
PKG-ORACLE_KNOWLEDGE_MAIN	This is the main event processing package, which runs continually to perform event processing as described in “Analytics Event Processing” on page 20
ANALYTICS_PROPERTIES	This table contains various properties that control event processing as described in “Event Processing Control Properties” on page 24.
INTERFACE_BATCH_PROCESS	This table contains detailed information about batches of events that have been processed, including start and end time, and exception information.

Event Processing Control Properties

Analytics uses a table called `ANALYTICS_PROPERTIES` to store batch control properties, including:

Property	Description
BATCHSIZE	Specifies the maximum number of events to include in a batch, as described in “Creating Event Batches for Transformation” on page 21. The default value is 10000.
BATCH_MIN_SIZE	Specifies the minimum number of events to include in a batch, as described in “Checking for Unprocessed Events” on page 21. This property helps to ensure that all events from a single session are processed within the same batch and available for reporting at the same time. The default value is 5000.
BUFFER_TIME	Specifies a floating time period relative to the current time. The event processing package will not include events from within this time period in a batch as described in “Ensuring Session Integrity in Reporting Data” on page 21. This property helps to ensure that all events from a single session are processed within the same batch and available for reporting at the same time. The default value is 30 minutes.
CURRENT_BATCH_PROCESS_DATE	This property specifies the date from which the initial batch will start. The default value is <td>.

Property	Description
ODI_LOG_PURGE_DURATION	This property specifies a floating time period, in days, relative to the current time. Event processing will automatically delete (purge) all ODI execution logs older than the value of this property, as described in “Purging the ODI Execution Logs” on page 23. The default value is 7 days.
SLEEP_INTERVAL	This property specifies the time interval that the event processing package will wait before querying the staging database for unprocessed events, as described in “Checking for Unprocessed Events” on page 21. The default value is 10 minutes.

The Interface Batch Process Table

The `INTERFACE_BATCH_PROCESS` table contains detailed information about batch start time, end time, and additional processing details, including exception (error) information, as described in “Logging Processing Activity” on page 22.

The Interface Package Process Table

The `INTERFACE_PACKAGE_PROCESS` table contains start time and end time information for the data load processes (dimension, staging, fact, and aggregate) as described in “Logging Processing Activity” on page 22, and start time and end time information for publishing to the reporting database.

Event Processing Settings

This section describes the settings that control event processing. You can access these settings by logging onto the Analytics database as an administrator.

The Initial Event Processing Date

The initial event processing date is controlled by the `CURRENT_BATCH_PROCESS_DATE` property, specified in the format `YYYYMMDD`. The date should reflect the date of the earliest data that is available in the table `DW_STAGE.DW_STG_DATA`. You can specify the initial processing date when you need to re-process failed events as described in “Monitoring Event Processing and Recovering from Failures” on page 25.

Monitoring Event Processing and Recovering from Failures

You can monitor the following processes and conditions to identify batch processing issues:

- purging the ODI logs as described in “Purging the ODI Execution Logs” on page 23
- event processing package execution as described in “Monitoring Event Processing Package Execution” on page 26
- exceptions in the staging tables as described in “Monitoring Staging Exceptions” on page 26

- exceptions in the reporting tables as described in “Monitoring Reporting Exceptions” on page 26
- batch processing errors as described in “Monitoring Batch Processing Status in ODI” on page 27

Monitoring Event Processing Package Execution

You can monitor event processing by querying various tables in the Analytics database. You can monitor event processing progress by querying:

- `DW_REPORTING.ANALYTICS_PROPERTIES` (see “Event Processing Control Properties” on page 24 for more information)
- `DW_REPORTING.INTERFACE_BATCH_PROCESS` (see “The Interface Batch Process Table” on page 25 for more information)
- `DW_REPORTING.INTERFACE_PACKAGE_PROCESS` (see “The Interface Package Process Table” on page 25 for more information)

Monitoring Event Processing Exceptions

You can check for event processing exceptions, which are logged as described in “Logging Processing Exceptions” on page 23. To check for exceptions, query the batch number in the Exception ID field.

Monitoring Staging Exceptions

You can check for exceptions in the staging database, `DW_STAGE`, using the following SQL:

```
SELECT batch_num, is_processed, date_processed, exception_id FROM
dw_stage.dw_stg_data WHERE exception_id NOT IN (0, -1);
```

The results will include any events that have an exception ID other than 0 (zero) and -1, which indicate a batch processing issue. An exception ID value of -1 indicates events that have not been processed. An exception ID value of 0 (zero) indicates events that have been processed successfully.

If the SQL returns exception IDs other than 0 and -1, note the affected batch number and re-process the events.

Monitoring Reporting Exceptions

You can check for exceptions in the reporting database using the following SQL:

```
SELECT * FROM dw_reporting.interface_batch_process WHERE
exception_id NOT IN (0, -1);
```

The results will include any events that have an exception ID other than 0 (zero) and -1, which indicate a batch processing issue. An exception ID value of -1 indicates events that have not been processed. An exception ID value of 0 (zero) indicates events that have been processed successfully.

If the SQL returns exception IDs other than 0 and -1, note the affected batch number and re-process the events as described in “Re-Processing Events” on page 27.

Monitoring Batch Processing Status in ODI

You can check ODI batch processing status using the ODI client. To check batch processing status:

- log onto to ODI as a SUPERVISOR
- select the Operator tab
- check for warnings and errors as follows:

Errors	are indicated in red.
Warnings	are indicated in yellow
Success	is indicated in green

If a load fails:

- log into the ODI work schema
- check the E\$ and I\$ tables for error details using the following SQL:

```
SELECT 'SELECT * FROM ' || table_name || ';' FROM user_tables WHERE
table_name LIKE 'E$%' OR table_name LIKE 'I$%';
```

This command generates a set of SQL statements specific to the dynamically created tables. You can execute these statements to locate exceptions within the tables.

Re-Processing Events

You can re-process batches of events as part of an event processing failure recovery process. You re-process batches by:

- stopping event processing as described in “Stopping Event Processing” on page 27
- repairing events to be processed as described in “Repairing Events that Failed Processing” on page 27
- resetting the processing date as described in “Resetting the Initial Event Processing Date” on page 28
- re-starting event processing as described in “Starting Event Processing” on page 23

Stopping Event Processing

To stop event processing, log into ODI and navigate to the Operator tab. Right-click the package PKG-ORACLE_KNOWLEDGE_MAIN and select Stop Normal or Stop Immediate.

Repairing Events that Failed Processing

You can repair events so that they can be re-processed. The repair script operates only on a data for a single day. You must run the process once for each day that contains events that you want to re-process. To run the repair process, execute the following commands, specifying the appropriate date:

```
UPDATE DW_STAGE.DW_STG_DATA
set batch_num=-1,
is_processed=-1,
```

```
DATE_PROCESSED=NULL,  
exception_id=-1  
WHERE to_char(timestamp, 'YYYYMMDD')='20121201'
```

Resetting the Initial Event Processing Date

You schedule events for re-processing by setting the initial event processing date to the date of the data containing the events that you want to re-process. To reset the processing date, execute the following commands, specifying the appropriate date:

```
TRUNCATE TABLE DW_REPORTING.INTERFACE_BATCH_PROCESS;  
TRUNCATE TABLE DW_REPORTING.INTERFACE_PACKAGE_PROCESS;  
UPDATE DW_REPORTING.DBO.ANALYTICS_PROPERTIES SET VALUE = 'YYYYMMDD'  
WHERE PROPERTY = 'CURRENT_BATCH_PROCESS_DATE';
```

Analytics Data Aggregation

You can view weekly and monthly totals for Analytics data within reports. Analytics automatically aggregates the data stored in the reporting data warehouse into weekly and monthly time dimensions. By default, the aggregation process is run daily.

Analytics uses a package within ODI called `LOAD_FACT_AGG_TABLES` to perform data aggregation. This aggregation package is configured such that it runs after all events (batches) have been processed for a given day. This package then initiates the interfaces required to aggregate data for each fact table in the data warehouse.

The aggregation process stores the aggregated weekly and monthly data in two sets of tables. For each fact table in the data warehouse, there are corresponding weekly and monthly aggregate tables, as described in “Weekly and Monthly Aggregation Tables” on page 29.

Monitoring Data Aggregation

The definitions that control data aggregation are stored in the `INTERFACE_AGG_PROCESS` table. You must monitor this table periodically for exceptions.

Each row in this table contains a predefined date. When an exception occurs, the details are recorded in the row containing the date on which the exception occurred.

Identifying Data Aggregation Exceptions

The `INTERFACE_AGG_PROCESS` table stores information about any exceptions that occur during the aggregation process.

Datekey	Stores each day till the year 2020 as of now.
Interface_Name	Contains each Interface run for aggregation

Month_Exception_ID

Is populated with an exception ID if there is an issue running monthly aggregation. Exception ID of 0 (zero) implies a successful run.

Week_Exception_ID

Is populated with an exception ID if there is an issue running Weekly aggregation. Exception ID of 0 (zero) implies a successful run.

Weekly and Monthly Aggregation Tables

For each fact table, there is a weekly aggregate table and a monthly aggregate table:

FACT_table	AGG_WEEK_	AGG_MONTH_
FACT_CASE	AGG_WEEK_CASE	AGG_MONTH_CASE
FACT_CONTENT	AGG_WEEK_CONTENT	AGG_MONTH_CONTENT
FACT_QUESTNS	AGG_WEEK_QUESTNS	AGG_MONTH_QUESTNS
FACT_RECOMMENDATION	AGG_WEEK_RECOMMENDATIONS	AGG_MONTH_RECOMMENDATIONS
FACT_RESPONSES	AGG_WEEK_RESPONSES	AGG_MONTH_RESPONSES
FACT_TOTAL	AGG_WEEK_TOTAL	AGG_MONTH_TOTAL
FACT_TOTAL_SEARCH	AGG_WEEK_TOTAL_SEARCH	AGG_MONTH_TOTAL_SEARCH
FACT_USR	AGG_WEEK_USR	AGG_MONTH_USR
FACT_USR_SEARCH	AGG_WEEK_USR_SEARCH	AGG_MONTH_USR_SEARCH
FACT_WORKFLOW	AGG_WEEK_WORKFLOW	AGG_MONTH_WORKFLOW
BRIDGE_QUESTN_RESPONSE	AGG_WEEK_BRG_QUESTN_RESPONSE	AGG_MONTH_BRG_QUESTN_RESPONSE
BRIDGE_QUESTN_RESPONSE_STG	AGG_WEEK_BRG_QUESTN_RESPONSE_STG	AGG_MONTH_BRG_QUESTN_RESPONSE_STG

How Analytics Uses Aggregated Data in Reporting

OBIEE contains a hierarchy that maps the aggregation tables to each fact. When users report on monthly or weekly data, OBIEE uses the data stored in the AGG_Month tables or AGG_Week tables, respectively.

Aggregation Level	OBIEE Reporting Hierarchy
Daily	FACT_
Weekly	AGG_WEEK_
Monthly	AGG_MONTH_

Important! If the aggregation process does not complete, the tables are empty and the reports do not display any data. Even though data for days within the specified time-period may exist in the daily data tables, monthly and weekly data is available only if the aggregation process has completed successfully and the aggregate data tables are populated.

Configuring and Managing Data Access and Storage

The Analytics data warehouse stores the staging and reporting data, as well as additional metadata, used to create the reports. You configure the data warehouse for an Analytics application during the installation process.

You can view details of the staging and reporting database schema as described in “Viewing the Analytics Database Schema” on page 30. You can manage the data in the staging tables by configuring the application to purge data from the staging tables as described in “Configuring and Managing Data Purging” on page 30.

Viewing the Analytics Database Schema

The Analytics installation process includes html data dictionary framesets that you can use to view the details of the staging and reporting database schema. The framesets are located in:

```
<install_dir>/inquirasql/documentation
```

To view the staging database schema:

- open the DW_STAGE folder
- open the index.html file in a browser

The browser displays the Analytics staging data dictionary frameset.

To view the reporting database schema:

- open the DW_REPORTING folder
- open the index.html file in a browser

The browser displays the Analytics reporting data dictionary frameset.

Configuring and Managing Data Purging

Analytics includes an automated process to remove outdated data from the staging database tables. The purging process is an ODI package that automatically deletes data from various tables. It purges data from a defined set of tables according to the schedule described in “The Default Purging Interval” on page 34.

You configure and schedule the purging process using an ODI data management agent. You enable, schedule and start the ODI agent using the process described in “Enabling and Scheduling an ODI Data Management Agent” on page 31.

The purging process is controlled by an ODI package, `PKG-PURGE_PROCESS_MAIN`, which includes the interface `INT_Purge_Data`. When the purging process starts, the interface does the following:

- determines the purge settings for each table
- updates the `PURGE_PROCESS` table
- deletes data based on each table's purge settings
- logs information about the purge and any exceptions that occur to the `PURGE_PROCESS` table

Enabling and Scheduling an ODI Data Management Agent

Analytics includes a defined ODI data management agent that you must enable, schedule, and start in order to operate the automated data purge process.

Note: For more information on working with agents in ODI, see the Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator.

Enabling the Agent

You enable the agent to connect to your ODI repository by specifying the following connectivity information in the ODI parameters file:

- repository connection information
- user authorization information
- jdbc and schema connectivity information

You specify repository connection and authorization information using an encrypted form of the ODI master repository password.

Specifying Agent Connectivity Information

To specify connectivity information for the ODI agent, edit the ODI parameters file:

- navigate to the `ODI_HOME` directory
- open the `agent\bin` directory, for example:

```
<ODI_home\Oracle_ODI1\oracledi\agent\bin>
```

or

```
<ODI_home_dir>\Middleware\Oracle_ODI1\oracledi\agent\bin
```

- open the `odiparams` file in an editor

You can now use the encryption utility to encrypt the master repository password.

Encrypting the ODI Master Repository Password

To encrypt the ODI master repository password:

- navigate to the directory:

```
ODI_HOME\agent\bin
```

- enter the following command as appropriate for your environment:

```
encode <ODI_MASTER_REPOSITORY_PASSWORD>
```

```
./encode <ODI_MASTER_REPOSITORY_PASSWORD>
```

The utility produces an encrypted password, for example:

```
c6ypWN0kqs7qoM0V58qif5zi
```

- copy the encrypted password.

Specifying Repository Connection and User Authorization Information

To specify repository connection and user authorization information, add the encrypted master repository password to the parameters file:

- locate the parameter `ODI_MASTER_ENCODED_PASS` in the Repository Connection Information section of the ODI parameters file and paste the encrypted password as the value:

```
set ODI_MASTER_ENCODED_PASS=c6ypWN0kqs7qoM0V58qif5zi
```

- locate the parameter `ODI_SUPERVISOR_ENCODED_PASS` in the User Login section of the ODI parameters file and paste the encrypted password as the value:

```
set ODI_SUPERVISOR_ENCODED_PASS=c6ypWN0kqs7qoM0V58qif5zi
```

Specifying JDBC Connection and Repository Schema Information

To specify JDBC connection and repository schema information, edit the values for the `ODI_MASTER_URL` and the `ODI_MASTER_USER` parameters in the ODI parameters file:

```
ODI_MASTER_URL=jdbc:oracle:thin:@<host>:<port>:<sid>
```

```
ODI_MASTER_USER=<Schema_name_of_ODI_Repository>
```

- save and Close the document

Scheduling the Purge Process

You schedule the purge process by:

- adding a scenario to the purge package using the ODI Studio
- starting the agent from a command prompt
- updating the agent's schedule using ODI Studio

Adding a Scenario to the Purge Package

To add the scenario:

- start ODI Studio:

start > All Programs > Oracle > Oracle Data Integrator > ODI studio

- navigate to:

Designer > DW_Data_Loads (folder) > PKG-PURGE_PROCESS_MAIN

- expand the package by selecting

'+') > Expand Scenarios > PKG_PURGE_PROCESS_MECHANISM_MAIN Version 001
> Scheduling > open DEVELOPMENT / ODI_LOGICAL_AGENT

- select DEVELOPMENT / ODI_LOGICAL_AGENT

The Scheduling DEVELOPMENT / ODI_LOGICAL_AGENT window opens.

- set the following scheduling parameters:

Parameter	Value
Context	Development
Agent	ODI_LOGICAL_AGENT
Status	Active
Execution	specify the desired interval, e.g. daily, weekly, etc. and specify the time to initiate the process.

- save and close 'DEVELOPMENT / ODI_LOGICAL_AGENT'

Starting the Agent

You start the agent by executing the agent script from the ODI /bin directory.

To start the agent:

- open a command prompt and navigate to the directory:

ODI_HOME\oracledi\agent\bin

- execute the appropriate script and arguments for your environment:

agent.bat "-NAME=ODI_AGENT" "-PORT=20910"

./agent.sh -PORT=20910 -NAME=ODI_AGENT

Updating the Agent Schedule

You update the agent schedule using ODI Studio.

To update the agent schedule:

- select Topology > Expand 'AGENTS' in the Physical Architecture > Open ODI_AGENT
- select 'UPDATE SCHEDULE'

ODI Studio displays the Select Repositories dialog.

- select the appropriate work repository, then select OK
- select View Schedule

ODI Studio displays the agent's schedule.

The Default Purging Interval

The data purging process is configured by default to delete data that is older than the specified purge interval from the defined set of tables. If the purge interval for a table is set to seven days, the purge process deletes any data older than seven days each time that it runs. You can change the purging interval as described in “Modifying the Purging Process” on page 35.

The purging process is configured by default to include the following tables in the Staging (DW_STAGE) and Reporting (DW_REPORTING) schema:

Schema	Table	Database Table Type	Table Description	Purge Interval
DW_STAGE	SW_STG_DATA	STAGE	Stores event-level data temporarily until the data is transformed and loaded into Fact and Workflow Fact tables.	7
DW_REPORTING	INTERFACE_BATCH_PROCESS	INTERFACE	Stores details about event batches.	720
DW_REPORTING	INTERFACE_PACKAGE_PROCESS	INTERFACE	Stores details on the data load processes for dimensions, staging, fact, and aggregate tables.	720

Note: Dimension tables, which store information about Channels, Locales, User Groups, and Users, are exempt from the automatic purging process.

Purge Settings

The purge settings for each table are stored in the PURGE_SETTING table within the DW_REPORTING schema. The PURGE_SETTING table contains the following columns:

Column	Description
Row Num	Specifies the order in which the process purges data in the DW_STAGE and DW_REPORTING schema.
Schema Name	Specifies the schema to which the table belongs.
Table Name	Specifies the table to purge.
Type	Specifies the type of table. Possible values are: STAGE INTERFACE
Duration	Specifies the number of days that data is retained.
Enabled	Specifies whether the process purges data for a table. Possible values are Y (data purging is enabled) and N (data purging is disabled).

Modifying the Purging Process

You can modify the purging process by editing the `PURGE_SETTING` table to:

- exempt tables from the purging process by setting the value of the `Enabled` column to `No`
- change the purging intervals for tables by editing the value of the `Duration` column
- add tables to the purging process by adding a row that identifies the table and specifies the desired `Duration` and `Enabled` status.

Monitoring the Purge Process

You can monitor the purge process by examining the contents of the `PURGE_PROCESS` table. The purge process records information about each purge operation in the `PURGE_PROCESS` table. Each row represents a table that was included in the purge.

The `PURGE_PROCESS` table includes the following columns:

Column	Description
<code>BEGINNING_DATETIME</code>	The earliest timestamp in the purged data.
<code>ENDING_DATETIME</code>	The most recent timestamp in the purged data.
<code>SCHEMA_NAME</code>	The schema that contains the table from which data was purged.
<code>TABLE_NAME</code>	The table from which data was purged.
<code>ROW_COUNT</code>	The number of records removed from the table.
<code>EXCEPTION_ID</code>	The indicator of success or error during the purge process. An exception ID of 0 indicates success; an ID of 66666 indicates an error. See “Diagnosing and Resolving Purge Errors” on page 35 for more information.
<code>PROCESSING_TIME</code>	The time that the purge process occurred.

Diagnosing and Resolving Purge Errors

When the purge process completes an operation to delete data from a table, it updates the `PURGE_PROCESS` table with one of the following exception IDs:

Exception ID	Description
0	Indicates that the purge process succeeded
66666	Indicates that there was an exception within ODI, and that the purge process failed

You can diagnose and resolve purge errors using your preferred SQL tool to:

- generate a set of select statements by querying the master ODI repository
- execute the select statements on the ODI temporary tables to identify specific ODI exceptions

You query the ODI repository using the following SQL:

```
select 'SELECT * FROM ' || 'ODI_STANDBY.' || table_name || ';' FROM
all_TABLES where owner='ODI_REPOSITORY_NAME';
```

where:

ODI_REPOSITORY_NAME is the name of the master repository as defined during the installation process.

The SQL generates a set of select statements. You can execute the select statements against the ODI temp tables to identify the specific ODI exceptions, which you can then diagnose and resolve.

Restoring Default Purge Settings

You can restore the automated purge settings to their defaults by running the following commands:

```
TRUNCATE TABLE DW_REPORTING.PURGE_SETTING;
```

```
Insert into PURGE_SETTING (ROW_NUM,SCHEMA_NAME,TABLE_NAME,TYPE,DURA-
TION,ENABLED) values (1,'DW_STAGE','DW_STG_DATA','STAGE',7,'Y');
```

```
Insert into PURGE_SETTING (ROW_NUM,SCHEMA_NAME,TABLE_NAME,TYPE,DURA-
TION,ENABLED) values
(2,'DW_REPORTING','INTERFACE_BATCH_PROCESS','INTERFACE_BATCH',720,'N'
);
```

```
Insert into PURGE_SETTING (ROW_NUM,SCHEMA_NAME,TABLE_NAME,TYPE,DURA-
TION,ENABLED) values
(3,'DW_REPORTING','INTERFACE_PACKAGE_PROCESS','INTERFACE',720,'Y');
```

These commands are packaged as a section within the script named `dw_populate.sql`. You can copy the commands from the script for convenience.

Important! Do not run the `dw_populate.sql` script to restore the default purge settings. The `dw_populate.sql` script will reset the data warehouse tables to their installation default values.