

# Oracle® Fusion Middleware

## Monitoring Business Activity with Oracle BAM



14c (14.1.2.0.0)  
F81874-02  
December 2024



Oracle Fusion Middleware Monitoring Business Activity with Oracle BAM, 14c (14.1.2.0.0)

F81874-02

Copyright © 2006, 2024, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

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

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

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

Intel and Intel Inside 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, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

## Preface

---

Audience	xix
Documentation Accessibility	xix
Related Documents	xix
Conventions	xx

## Part I Introducing Oracle Business Activity Monitoring 14c

---

### 1 Understanding Oracle Business Activity Monitoring

---

What You Can Do with BAM	1-1
Event Stream Analytics and Alerts	1-1
Process Analytics and Dashboards	1-2
Business-Friendly Graphs and Charts	1-2
Enterprise Grade Infrastructure	1-2
Concepts and Terms in BAM	1-3
BAM Architecture	1-4
BAM Components	1-5
Oracle Metadata Services Repository	1-6
Persistence Engine	1-6
Continuous Query Service	1-6
Report Cache	1-6
Alert Service	1-6
Data Objects and the Process Star Schema	1-7
BAM Dashboards	1-7
BAM Main Pages and User Task Summaries	1-8
Viewers	1-8
Creators	1-8
Architects	1-8
Administrators	1-9
Navigating in BAM Composer	1-9
Browser Settings	1-10
Logging In	1-10

Selecting the Home, Administrator, or Designer Page	1-10
Using the Navigation Pane	1-11
Expanding and Collapsing Panes	1-11
Selecting a Project	1-12
Showing and Hiding UI Text	1-12
Closing Tabs	1-12
Setting Language, Time Zone, and Accessibility Preferences	1-12
Sample BAM Projects	1-13
Installing or Upgrading to BAM Release 14c	1-13
Moving BAM to Another Environment	1-14

## 2 Using Oracle Business Activity Monitoring as a Viewer

---

Understanding Dashboards and Alerts	2-1
Using the Home Page	2-3
Selecting a Project	2-3
Choosing Carousel or List View for Most Recent and Favorites	2-3
Opening a Dashboard	2-4
Adding a Dashboard to the Favorites List	2-4
Opening a Dashboard when Oracle BAM Is Started	2-4
Viewing Alert History	2-4
Filtering the Alert History List	2-5
Clearing the Alert History List	2-5
Viewing Dashboards	2-5
Dashboard Parameters	2-6
Saving a Dashboard to an HTML File	2-6
Filtering Runtime Data in a View	2-6
Modifying Active Data Properties in a View	2-7
Changing Dimensions in a View	2-9
Exploring View Data	2-9
View Actions	2-10

## 3 Tuning Oracle Business Activity Monitoring 14c

---

Tuning Schema Design	3-1
Tuning Dashboard Design and Deployment	3-2
Tuning Query Design	3-2
Tuning Data Archiving and Purging	3-3
Tuning Scalability and Performance	3-3
Real-Time Data Sources for Oracle BAM	3-4
Tuning Production Roll-out	3-4

### 4 Planning and Creating Projects

---

Planning a Project	4-1
Choosing Bottom-Up or Top-Down Project Building Steps	4-2
Creating a Project	4-2
Opening a Project	4-3
Renaming a Project	4-3
Deleting a Project	4-4
Securing a Project	4-4
Copying a Project	4-5
Managing Changes in a Project	4-6

### 5 Working with Data Objects

---

Understanding Data Objects	5-1
Data Object Columns	5-1
Designer Access to Data Objects	5-2
Data Object Types	5-2
Data Object Prerequisites and Uses	5-4
Adding a Data Object to a Project	5-5
Removing a Data Object from a Project	5-5
Viewing Data Object Information	5-6
The General Tab	5-6
The Columns Tab	5-7
The Tables Tab	5-8
The Indexes Tab	5-8
The Hierarchies Tab	5-9
The Retention Tab	5-9
The Row Security Tab	5-9
The Data Tab	5-9
Modifying Data Tab Preferences	5-9
The Filter Tab	5-10
The Calculated Fields Tab	5-10

### 6 Creating Business Queries

---

Understanding Business Queries	6-1
Business Query Prerequisites and Uses	6-2
Creating a Flat SQL Query	6-3
Creating a Group SQL Query	6-4

Creating a Tree Model Query	6-6
Creating a Continuous Query	6-7
Using the Trending Detection Template	6-9
Using the Duplicate Detection Template	6-10
Using the Monitor Count Template	6-10
Using the Moving Aggregation Template	6-11
Using the Missing Event Template	6-12
Using the Top N Template	6-13
Using the KPI Alert Template	6-14
Creating a Pre-defined SQL Query	6-16
Editing a Business Query	6-18
Renaming a Business Query	6-19
Deleting a Business Query	6-19
Securing a Business Query	6-20
Filtering Data	6-21
Adding a Group Data Filter	6-22
Filter Comparison Operations	6-22
Exceptions for Datetime Filter Comparisons	6-23
Using Top N Data	6-23
Applying Aggregate Functions to Groups	6-23
Using a Time Group or a Time Series	6-24

## 7 Creating KPIs

---

Understanding KPIs	7-1
KPI Prerequisites and Uses	7-2
Creating a Realtime KPI	7-2
Creating a Scheduled KPI	7-4
Editing a KPI	7-5
Renaming a KPI	7-6
Deleting a KPI	7-6
Securing a KPI	7-6

## 8 Creating and Using Business Views

---

Understanding Business Views	8-1
Business View Prerequisites and Uses	8-4
Creating Business Views	8-4
Creating Chart and Graph Business Views	8-5
Creating a Business View: Steps	8-10
Creating Table Business Views	8-11
Understanding Table Business Views	8-11

Creating Table Business Views: Steps	8-12
Adjusting Column Width in Table Business Views	8-14
Reordering Columns in Table Business Views	8-14
Sorting Data in Table Business Views	8-14
Selecting Editable Fields in an Action List	8-14
To Enable or Disable Pivot Table Features	8-15
To Pivot the View of Data	8-15
To Add A Category Aggregate Function to a Pivot Table	8-16
Creating KPI Watchlist Business Views	8-16
Creating Gauge Business Views	8-17
Creating Geo Map Business Views	8-19
Creating Scatter and Bubble Business Views	8-21
Creating Treemap Business Views	8-22
Map Points Properties	8-24
Zoom Properties	8-24
Formatting Business Views	8-24
General Properties	8-25
General Properties for Business Views	8-25
Scatter and Bubble General Properties	8-27
List General Properties	8-27
Gauge General Properties	8-27
General Properties Specific to Pivot Table Business Views	8-28
General Properties Specific to Geo Map Business Views	8-28
Label, Text, Title, and Value Properties	8-31
Axis Properties	8-32
Axis Properties Specific to Dial and Statusmeter Business Views	8-32
X-Axis Properties	8-33
Y-Axis Properties	8-33
Legend Properties	8-34
Target Properties	8-35
Animation Properties	8-36
Chart Animation Properties	8-36
Gauge Animation Properties	8-37
Slice Properties	8-37
Text Properties	8-37
Layout Properties	8-38
List Layout Properties	8-38
Treemap Layout Properties	8-39
Editable Fields Properties	8-39
Pivot Table Data Properties	8-39
Pivot Table Label Properties	8-40
Thresholds Properties	8-40

Collapsed List Thresholds Properties	8-40
Gauge Thresholds Properties	8-41
Treemap Thresholds Properties	8-41
Node Depth Properties	8-42
Labels Properties	8-43
Tooltip Labels Properties	8-43
Runtime Interactions with Business Views	8-44
Using Drilling	8-44
Drilling Down	8-44
Drilling Up	8-45
Drilling Through	8-46
Drilling Across	8-46
Enabling Drilling Down	8-46
Enabling Drilling Through to Details	8-47
Configuring Drill-Across Targets	8-48
Using Actions in Business Views	8-49
Using Active Data	8-51
Editing, Renaming, Deleting, and Securing Business Views	8-53
Editing a Business View	8-53
Renaming a Business View	8-54
Deleting a Business View	8-54
Securing a Business View	8-55

## 9 Creating Dashboards

---

Understanding Dashboards	9-1
Dashboard Prerequisites and Uses	9-2
Creating a Dashboard and Adding Views	9-3
Creating a Springboard	9-5
Process Analytics Springboard	9-5
Configuring a Process Health Dashboard	9-6
Open Process Summary Dashboard	9-6
Configuring the Open Process Summary Dashboard	9-7
Workload Analysis Dashboard	9-7
Cycle Time Dashboard	9-8
Configuring the Cycle Time Dashboard	9-8
Closed Process Analytics Dashboard	9-8
Configuring the Closed Process Analytics Dashboard	9-9
Configuring the Workload Analysis Dashboard	9-9
Adding Other Content to Dashboard Cells	9-9
Formatting Dashboard Cells	9-10
Formatting the Dashboard	9-10

Creating a Tabbed Dashboard	9-11
Editing a Dashboard	9-12
Renaming a Dashboard	9-12
Deleting a Dashboard	9-13
Securing a Dashboard	9-13
Displaying a Dashboard URL	9-14
Viewing a Dashboard	9-15

## 10 Creating Alerts

---

Understanding Alerts	10-1
Alert Prerequisites and Uses	10-2
Creating an Alert	10-3
Activating and Deactivating an Alert	10-5
To Change the Activity Status from the Navigator	10-6
To Change the Activity Status from the Alert Tab	10-6
Creating Dependent Alerts	10-6
Editing an Alert	10-7
Renaming an Alert	10-7
Deleting an Alert	10-7
Securing an Alert	10-8
Viewing Alert History from the Home Page	10-8
Launching Alerts by Invoking Web Services	10-9
Alert and Web Service Example	10-9
Sending Alerts to External Email Accounts	10-10
Alert Events	10-10
In a specific amount of time	10-10
At a specific time today	10-10
On a certain day at a specific time	10-10
Every interval between two times	10-10
Every date interval starting on certain date at a specific time	10-11
Select a CQL Query	10-11
When a data field in a data object meets specified conditions	10-11
When this rule is launched	10-12
Alert Conditions	10-12
If it is between two times	10-13
If It is between two days	10-13
If it is a particular day of the week	10-13
Alert Actions	10-13
Send a dashboard link via email	10-13
Send a message via email	10-14
Delete rows from a data object	10-15

Insert values into a data object	10-15
Send a parameterized message	10-16
Launch a rule	10-20
Run a web service	10-20
Launch a Custom Action	10-22

## 11 Creating Parameters

---

Understanding Parameters	11-1
Parameter Prerequisites and Uses	11-1
Creating a Value Parameter	11-2
Creating a List Parameter	11-3
Using a Parameter in a Query	11-4
Editing a Parameter	11-5
Renaming a Parameter	11-5
Deleting a Parameter	11-6
Securing a Parameter	11-6
Using Parameters in Dashboard URLs	11-7
Driving Parameters from Other Views	11-7
Using Prompts in Dashboards	11-9

## Part III Integrating Oracle Business Activity Monitoring with Other Applications

---

### 12 Integrating with Oracle Business Process Management

---

Understanding Oracle BAM and BPM Integration	12-1
Configuring Cross-domain Communication for BAM, SOA, and BPM	12-1
Enabling Process Metrics	12-2
Understanding the Pre-assembled BPM Dashboards	12-3
Process Analytics Dashboards Dashboard	12-4
Data Visibility Rules for the Process Analytics Dashboards Dashboard	12-5
Process Summary Dashboard	12-5
Process Summary View	12-6
Process Summary Query	12-7
Analysis of Open Processes Dashboard	12-7
Analysis of Open Processes View	12-8
Analysis of Open Instances Query	12-9
Analysis of Closed Processes Dashboard	12-9
Analysis of Processes Closed Today View	12-10
Historical Process Completion Times View	12-11
Analysis of Instances Closed Today Query	12-11

Historical Process Completion Times Query	12-12
Bottleneck Analysis of Open Processes Dashboard	12-12
Bottleneck Analysis of Open Processes View	12-13
Bottleneck Analysis of Open Processes Query	12-14
Process Summary at a Glance Dashboard	12-14
Process Summary at a Glance View	12-15
Process Summary at a Glance Query	12-15
Trend Analysis for Processes Dashboard	12-15
Process Cycle Time Trend View	12-16
Process Inflow Trend View	12-17
Process Outflow Trend View	12-17
Process Cycle Time Trend Query	12-17
Process Inflow Trend Query	12-18
Process Outflow Trend Query	12-18
Process Dashboard for Process Dashboard	12-19
Processes Opened Today Total View	12-20
Processes Closed Today Total View	12-20
Processes Faulted Today Total View	12-20
Processes Not Due Soon Total View	12-21
Processes Due Soon Total View	12-21
Processes Overdue Total View	12-21
Bottleneck Analysis of Open Tasks View	12-22
Processes Opened Today Total Query	12-22
Processes Closed Today Total Query	12-22
Processes Faulted Today Total Query	12-23
Processes Not Due Soon Total Query	12-23
Processes Due Soon Total Query	12-23
Processes Overdue Total Query	12-24
Bottleneck Analysis of Open Tasks Query	12-24
Process Activity Analysis Dashboard	12-24
Tasks Opened Today Total View	12-26
Tasks Closed Today Total View	12-26
Tasks Not Due Soon Total View	12-26
Tasks Due Soon Total View	12-26
Tasks Overdue Total View	12-27
Tasks Not Due Soon View	12-27
Tasks Due Soon View	12-27
Tasks Overdue View	12-28
Assignees Not Due Soon View	12-28
Assignees Due Soon View	12-29
Assignees Overdue View	12-29
Tasks Opened Today Total Query	12-29

Tasks Closed Today Total Query	12-30
Tasks Not Due Soon Total Query	12-30
Tasks Due Soon Total Query	12-30
Tasks Overdue Total Query	12-31
Tasks Not Due Soon Query	12-31
Tasks Due Soon Query	12-31
Tasks Overdue Query	12-32
Assignees Not Due Soon Query	12-32
Assignees Due Soon Query	12-33
Assignees Overdue Query	12-33
Task Trend Analysis for Process Dashboard	12-33
Task Cycle Time Trend View	12-34
Task Inflow Trend View	12-35
Task Outflow Trend View	12-35
Task Cycle Time Trend Query	12-35
Task Inflow Trend Query	12-36
Task Outflow Trend Query	12-36
BPM Process View Dashboard	12-37
Assignees Not Due Soon PV View	12-38
Assignees Due Soon PV View	12-38
Assignees Overdue PV View	12-39
Assignees Not Due Soon PV Query	12-39
Assignees Due Soon PV Query	12-39
Assignees Overdue PV Query	12-40
BPM Dashboard Parameters	12-40
Assignee Is Group Parameter	12-41
Process Name Parameter	12-41
Select a Process Parameter	12-41
Select data after Parameter	12-41
Select data before Parameter	12-42
Time Period (Days) Parameter	12-42
Understanding the Oracle BAM Data Objects for BPM	12-42
Process Data Object	12-43
Activity Data Object	12-45
HWF Task Data Object	12-45
HWF Task Assignment Data Object	12-46

## 13 Integrating with Oracle SOA Suite

---

Understanding Oracle BAM and Oracle SOA Suite Integration	13-1
Understanding the Preassembled Dashboards	13-1

## Part IV Using Oracle Business Activity Monitoring as an Administrator

---

### 14 Creating and Managing Oracle BAM Data Objects

---

Understanding Oracle BAM Data Objects	14-1
Creating a Data Object	14-3
Creating a Simple Data Object	14-3
Creating a Derived Data Object	14-4
Creating an External Data Object	14-5
Creating a Logical Data Object	14-6
Adding Columns and Hierarchies to a Data Object	14-7
Column Data Types	14-7
Adding Columns to a Simple or Derived Data Object	14-8
Adding Columns to a Logical Data Object	14-9
Adding Index Columns	14-10
Adding Hierarchies	14-11
Creating a Non-Datetime Hierarchy	14-12
Creating a Datetime Hierarchy	14-12
Adding Calculated Fields	14-13
Adding Data Object Parameters	14-14
Viewing Maximum Numbers of Columns in a Derived Data Object	14-15
Securing a Data Object	14-16
Setting Security Permissions for a Data Object	14-16
Setting Row Security for a Data Object	14-17
Managing Data in Data Objects	14-18
Adding Rows to a Data Object Automatically	14-18
Adding Rows to a Data Object Manually	14-19
Editing Rows in a Data Object	14-19
Purging Data in a Data Object	14-20
Exporting Data from a Data Object	14-20
Setting Data Retention in a Data Object	14-21
Scheduling Automatic Data Purges for a Data Object	14-21
Filtering Data in a Data Object	14-22
Specifying Slow-Changing Dimensions for a Data Object	14-22
Using Data Object Folders	14-23
Editing, Renaming, and Deleting Data Objects	14-23
Editing a Data Object	14-23
Renaming or Moving a Data Object	14-24
Deleting a Data Object	14-24

Operators and Functions for Calculated Fields	14-24
Operators	14-26
Ago	14-27
Avg	14-27
Case	14-27
Ceiling	14-28
Concat	14-28
Count	14-28
CountDistinct	14-28
DateAdd	14-29
DateDiff	14-29
DayName	14-29
DayOfMonth	14-30
DayOfWeek	14-30
DayOfYear	14-30
Default	14-30
Else	14-30
Floor	14-31
Hour	14-31
If	14-31
IfNull	14-32
Length	14-32
Lower	14-32
Max	14-32
Median	14-33
Min	14-33
Minute	14-33
Month	14-33
MonthName	14-33
Noop	14-34
Now	14-34
PercentOfTotal	14-34
Quarter	14-34
Rank	14-35
Repeat	14-35
Replace	14-35
Round	14-35
Second	14-36
StdDev	14-36
StdDevsFromMean	14-36
Substring	14-36
Sum	14-37

Switch	14-37
Then	14-37
TrimEnd	14-37
TrimStart	14-38
Upper	14-38
Variance	14-38
VariancesFromMean	14-38
Week	14-39
Year	14-39

## 15 Creating Oracle BAM Enterprise Message Sources

---

Understanding Enterprise Message Sources	15-1
Creating Enterprise Message Sources	15-2
Preparing to Create an Enterprise Message Source	15-2
Creating an Enterprise Message Source: Steps	15-3
Configuring a DateTime Specification	15-9
Oracle AQ JMS DateTime Considerations	15-11
Using Advanced XML Formatting	15-12
Configuring EMS Error Handling	15-12
Using Enterprise Message Sources	15-14
Editing an Enterprise Message Source	15-14
Starting and Stopping an Enterprise Message Source	15-14
Monitoring Enterprise Message Source Metrics	15-15
Copying an Enterprise Message Source	15-15
Securing an Enterprise Message Source	15-16
Deleting an Enterprise Message Source	15-17
Creating an Outbound Connection JNDI	15-17
Updating the Oracle BAM Adapter Deployment	15-19
Using Foreign JMS Providers	15-19
Configuring the JCA Alerts Mechanism in EMS	15-20

## 16 Monitoring and Managing Oracle BAM

---

Understanding Oracle BAM Monitoring and Management	16-1
Monitoring Continuous Queries	16-1
Monitoring Viewsets	16-2
Monitoring the State of the BAM Server and Services	16-3
Configuring BAM Server MBean Properties	16-4
Using the BAM Diagnostic Framework	16-5
Monitoring Oracle BAM Performance	16-5
Tuning Oracle BAM Performance	16-6

Monitoring Oracle BAM Logs	16-6
Monitoring Oracle BAM Web Services	16-7
Managing Oracle BAM Availability	16-7
Configuring Oracle User Messaging Service	16-8

## 17 Securing Oracle BAM

---

Understanding Oracle BAM Security	17-1
Understanding Oracle BAM Permissions	17-1
Managing Oracle BAM Users	17-2
Predefined Groups and Roles	17-2
Adding Users to Groups	17-3
Creating Custom Groups	17-4
Creating Custom Roles	17-4
Configuring Oracle BAM Server Security	17-4
Configuring Credential Mapping	17-4
Configuring Secure Socket Layer	17-5

## 18 Using Commands with Oracle BAM

---

Using BAMCommand	18-1
Executing BAMCommand	18-1
Using the BAMCommand Configuration File	18-2
Running BAMCommand Remotely	18-2
Specifying the Command and Parameter Syntax	18-2
Specifying General Parameters	18-2
Specifying Security Credentials	18-4
Specifying Object Names	18-4
Summary of Individual Operations	18-5
Detailed Operation Descriptions	18-6
Clear	18-6
Delete	18-7
Export	18-8
Import	18-10
Rename	18-13
Command File Structure	18-14
Operation IDs	18-14
Continue On Error	18-15
Log File Structure	18-15
Export File Structure	18-16
Migrate Considerations	18-18
Upgrade Considerations	18-19

Regular Expressions	18-20
Using the Loadgen Utility	18-22
Overview of Loadgen	18-22
Loadgen Features	18-23
Configuration and Execution	18-23
Command-Line Syntax	18-24
Usage Examples	18-25
Setting the -frequency to Inject Data in Fixed Periods at a Stable Rate	18-25
Inject at a Rate From Initial Rate to Final Rate with a Single File	18-25
Multiple Threads for Higher Throughput	18-26
Multiple XML Input Files	18-26
Batching Operations to Group Multiple Events Into a Single Operation	18-27
Simulate Data for Functional Testing	18-27
Operating on an Archive Relation DO and a Stream DO	18-27
Running Loadgen With a Command Line in a Control File	18-28
CRUD XML Insert, Update, and Delete Examples	18-28

## 19 Using Oracle BAM Web Services

---

Understanding Oracle BAM Web Services	19-1
Using the Legacy Data Object Web Services	19-2
How to Use the Legacy Data Object Web Services	19-2
DataObjectOperationsByName	19-3
Delete	19-3
Get	19-3
Insert	19-4
Update	19-4
Upsert	19-5
DataObjectOperationsByID	19-5
Batch	19-5
Delete	19-6
Insert	19-6
Update	19-7
Upsert	19-7
Using the Data Object Web Services	19-8
How to Use the Data Object Web Services	19-8
DOOperations	19-9
Batch	19-9
Delete	19-10
Get	19-10
Insert	19-11
Update	19-11

Upsert	19-11
DODefinition	19-12
Create	19-12
Delete	19-15
Get	19-15
Update	19-16
Using the ManualRuleFire Web Service	19-16
How to Use the ManualRuleFire Web Service	19-17
ManualRuleFire Operations	19-17
FireRuleByName	19-17

# Preface

*Monitoring Business Activity with Oracle BAM* describes:

- Using Oracle Business Activity Monitoring (Oracle BAM) to monitor business processes for making tactical and strategic decisions.
- Creating dashboards that contain graphical views of data updated either in real time as streams or on a scheduled basis.
- Integrating Oracle BAM with both the Oracle SOA Suite and the Oracle Business Process Management (Oracle BPM) products.
- Alerting capabilities for business users to monitor business events, manage business exceptions, and continuously optimize their processes.

## Audience

This manual is intended for all users of Oracle Business Activity Monitoring, including those who view data, those who design data views, and those who administer Oracle Business Activity Monitoring.

## Documentation Accessibility

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

### Accessible Access to Oracle Support

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

## Related Documents

Refer to the [Oracle Fusion Middleware library on the Oracle Help Center](#) for additional information.

- For Oracle BAM information, see Oracle Business Activity Monitoring.
- For Oracle SOA Suite information, see Oracle SOA Suite.
- For adapters information, see On-Premises Integration Adapters.
- For versions of platforms and related software for which Oracle products are certified and supported, review the [Certification Matrix on OTN](#).

---

## Conventions

The following text conventions are used in this document:

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

---

# Part I

## Introducing Oracle Business Activity Monitoring 14c

This part presents an overview of Oracle Business Activity Monitoring (BAM) and explains how to use the BAM Viewer (Home page) to view dashboards and alert history.

This part includes the following chapters:

- [Understanding Oracle Business Activity Monitoring](#)
- [Using Oracle Business Activity Monitoring as a Viewer](#)
- [Tuning Oracle Business Activity Monitoring 14c](#)

# 1

## Understanding Oracle Business Activity Monitoring

Oracle Business Activity Monitoring (BAM) 14c monitors your business processes in real time to help you make informed tactical and strategic business decisions. This chapter gives a brief overview of Oracle BAM and what you can do with it.

This chapter includes the following sections:

- [What You Can Do with BAM](#)
- [Concepts and Terms in BAM](#)
- [BAM Architecture](#)
- [BAM Components](#)
- [BAM Main Pages and User Task Summaries](#)
- [Navigating in BAM Composer](#)
- [Sample BAM Projects](#)
- [Installing or Upgrading to BAM Release 12c](#)
- [Moving BAM to Another Environment](#)

### What You Can Do with BAM

Oracle BAM offers *right-time* operational intelligence for mission critical business processes. Oracle BAM analyzes data before, during, and after business events.

- Analytics *before* a risk to a business goal materializes warn you that course correction could be needed.
- Analytics *during* a business event enable you to take immediate action.
- Analytics *after* a business event enable a full examination of the causes and aftermath of the event.

Oracle BAM offers these high-level sets of new features for Release 14c:

- [Event Stream Analytics and Alerts](#)
- [Process Analytics and Dashboards](#)
- [Business-Friendly Graphs and Charts](#)
- [Enterprise Grade Infrastructure](#)

### Event Stream Analytics and Alerts

Oracle BAM provides real-time pattern matching, trend analysis, rolling-window computations, and both static and dynamic thresholds. The built-in Continuous Query Language (CQL) engine evaluates incoming event streams against business requirements and data patterns of interest.

You can define these patterns with easy-to-use business query and Key Performance Indicator (KPI) templates, insulated from the underlying CQL. Furthermore, you can archive the event streams for fuller analysis later. See [Creating Business Queries](#) and [Creating KPIs](#) for more information.

Based on these data patterns, you can implement a wide variety of alerts and actions, such as emailing particular users, recording data, or launching a web service. You can even daisy-chain alerts and define custom actions. See [Creating Alerts](#) for more information.

## Process Analytics and Dashboards

Oracle BAM provides a rich set of ready-to-use dashboards for all major components of Oracle Business Process Management (BPM) and Oracle Service-Oriented Architecture (SOA) Suite.

Process Analytics dashboards include metrics such as task queue depth and bottleneck analysis of business processes. See [Integrating with Oracle Business Process Management](#) and [Integrating with Oracle SOA Suite](#) for more information.

You can create your own dashboards to present the information you need in a visually effective, easy-to-read format. See [Creating Dashboards](#) for more information.

## Business-Friendly Graphs and Charts

Business-friendly graphs and charts within Oracle BAM help with viewing your data in various visual formats.

With the adoption of Oracle Application Development Framework (ADF) Data Visualization Tool (DVT) components, Oracle BAM offers a wide range of intuitive visualizations that you can include in dashboards, including treemaps, scatter charts, bubble charts, and KPI watchlists. New DVT charts within Oracle BAM provide added chart capabilities apart from the old graph components. See [Creating and Using Business Views](#) for more information.

Oracle BAM also integrates with Oracle JavaScript Extension Toolkit (JET) to provide JET-based visualizations for some chart views. You can access JET dashboards in the same way you access ADF dashboards. See [Displaying a Dashboard URL](#) for more information.

### Note:

Some business views based on ADF components may display formatting issues. For instance, table business views may have incorrect value formatting values while using active data. These errors are ADF-based.

## Enterprise Grade Infrastructure

The Oracle BAM 14c architecture has been completely reengineered to be highly scalable, easily managed, and more tightly integrated with Oracle BPM and Oracle SOA Suite.

Various Oracle BAM components at both design time and run-time enable this scalability to help achieve flexible integration. See [BAM Architecture](#) for more information.

Oracle BAM incorporates components such as the Oracle Complex Event Processing Service Engine for real-time data analysis, the Oracle Application Development Framework for data presentation, and Oracle Coherence for active data caching. See [BAM Components](#) for more information.

# Concepts and Terms in BAM

Learn a comprehensive list of all terms and concepts within Oracle BAM.

In alphabetical order, the terms are as follows.

**aggregation**

A calculation performed on a measure and used for evaluation, such as an average, sum, count, maximum, or minimum.

**analytics**

Operational data captured from a business process to measure its effectiveness in attaining a business goal and to determine possible corrective actions.

**attribute**

A data field that can be used for filtering or references.

**business indicator**

A data field used for analysis, such as a measure, dimension, or counter.

**business intelligence**

Insight based on historical, data-centered analytics.

**counter**

A data field that can be counted. For example, the number of rows of a process name data field with the name of a specific process can be counted.

**data field**

A column in a data object.

**data object**

A model of data for use and monitoring within BAM. The data can be saved in raw format in a database or in an external table. The data can also be a stream from an ongoing activity.

**dimension**

A data field that provides a type of grouping on which aggregation is performed, used for drilling. For example, sales dimensions might be geographic region, company, and product.

**drilling**

Analyzing how data is broken down by dimensions. For example, sales can be broken down by geographic region, company, and product.

**key performance indicator (KPI)**

An important measure that is compared to a specific business goal, such as monthly sales.

**measure**

A numeric data field that can be measured and used in comparisons to evaluate business processes.

**measurement**

Capturing the value of a measure at a specific point in time.

**operational intelligence**

Insight based on real-time, dynamic, process-centered analytics.

**process**

A sequence of business tasks that results in a well-defined outcome. For example, a sales quote process includes tasks such as creating the quote, reviewing business practices, reviewing terms, and approving the quote, resulting in a quote that is either approved or rejected.

**risk indicator**

An important measure that provides early warning signs to identify events that may harm existing processes.

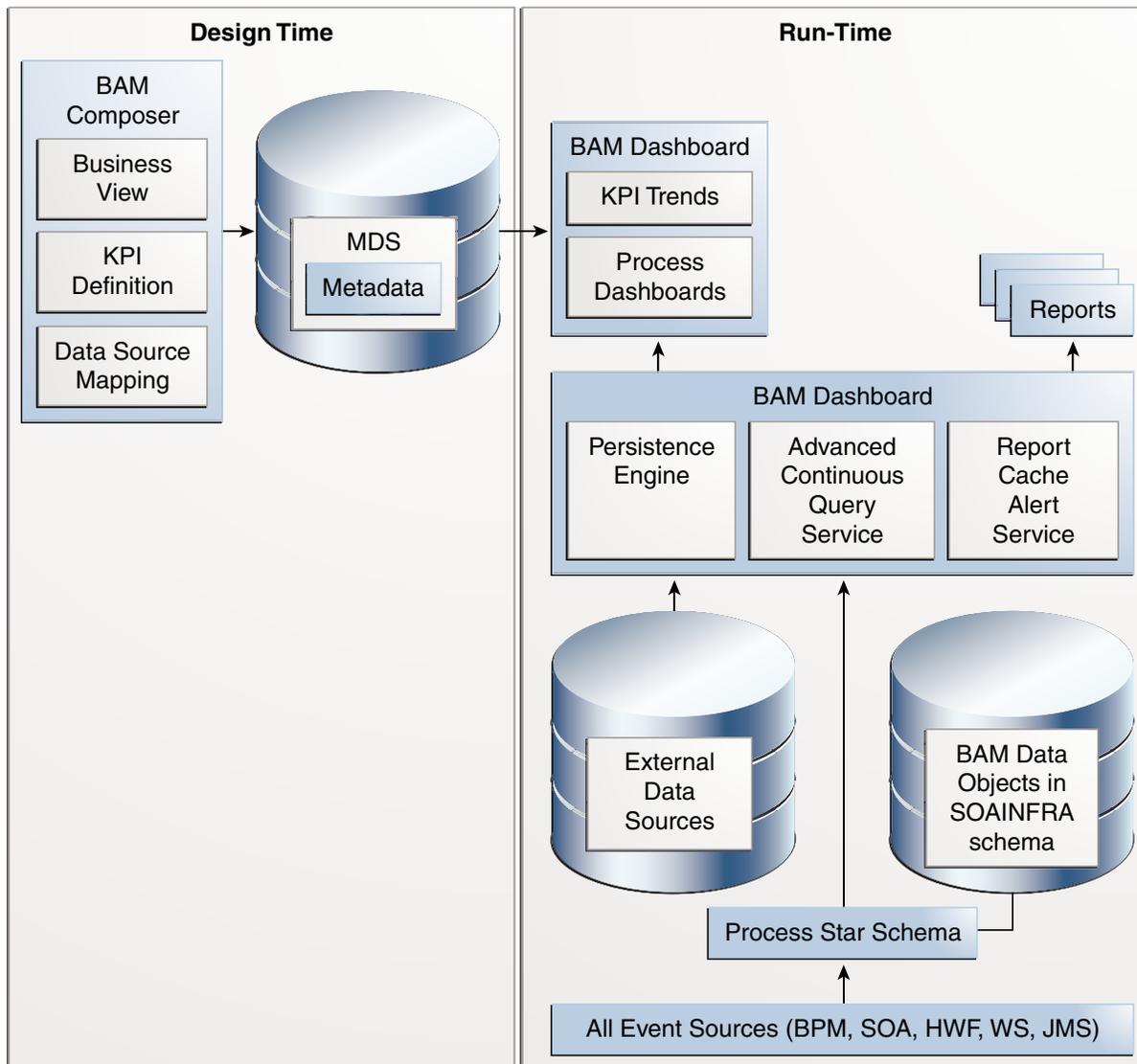
## BAM Architecture

Oracle BAM has a seamless architecture that allows for easy integration with your enterprise grade infrastructure.

[Figure 1–1](#) shows the main components of Oracle Business Activity Monitoring, which include the Composer browser interface, the Metadata Store (MDS), the persistence engine, the Continuous Query Service (CQS), the report cache and alert service, data objects and the process star schema, and BAM dashboards. Oracle BAM can consist of multiple managed servers that provide high availability.

For details about the Composer, see [Navigating in BAM Composer](#). For details about the other components, see [BAM Components](#).

Figure 1-1 Oracle BAM Architecture



## BAM Components

Oracle BAM comprises of many components such as the Metadata Services Repository, Dashboards, BAM Composer, Persistence Engine and many other components.

The following topics follow:

- [Oracle Metadata Services Repository](#)
- [Persistence Engine](#)
- [Continuous Query Service](#)
- [Report Cache](#)
- [Alert Service](#)
- [Data Objects and the Process Star Schema](#)

- [BAM Dashboards](#)

For details about the Composer, see [Navigating in BAM Composer](#).

## Oracle Metadata Services Repository

The Oracle Metadata Services (MDS) repository contains metadata for deployed Oracle Fusion Middleware component applications, including BPM and SOA applications.

Oracle BAM uses the MDS to store metadata for BAM entities such as data objects, dashboards, and alerts.

## Persistence Engine

The persistence engine is responsible for handling data object modifications.

Based on Oracle Coherence, the persistence engine receives data object modifications (insert, update, and delete), caches them, and passes them to the Continuous Query Service. Modifications to archived stream and archived relation data objects are persisted.

## Continuous Query Service

The Continuous Query Service (CQS) is a BAM-specific wrapper around the Continuous Query Language (CQL) engine within the Oracle Complex Event Processing Service Engine.

The CQS is a pure *push* system: query results are delivered automatically. The CQS supports both stream (non-persistent) and archived relation (persistent) data objects.

When you create a query, the CQS sets up tables in the CQL engine, registers the query, and listens for data changes from the persistence engine. The query result is processed in the CQL engine, then pushed to the CQS and on to the report cache.

## Report Cache

The report cache receives and caches query results, then pushes the results to dashboards.

The report cache holds the query results as viewsets. A viewset includes a data snapshot and the changes since the last snapshot for business views in the same project that use the same query and row security filters. The first view opens in a dashboard for a given query creates the master viewset. Opening other views that use the same query creates slave viewsets. Viewsets minimize time-consuming query reexecution and ensure that views are incrementally updated using a push-based mechanism.

You can configure `ReportCache` snapshot wait time in seconds using `RCSnapshotWaitTime` property in Enterprise Manager. It configures the `ReportCache` to maximum wait time before proceeding to open the viewset.

## Alert Service

The alert service is responsible for handling query results.

The alert service compares query results to alert events and conditions. If these events and conditions are satisfied, the alert service executes the corresponding alert actions and notifications. For email notifications, the alert service uses the Oracle User Messaging Service (UMS).

## Data Objects and the Process Star Schema

Data objects in BAM are based on the process star schema. This is the star schema standard adapted for use in process analytics.

Fact tables contain measures such as process running times, and dimension tables contain dimensions such as process names. BAM provides the following:

- Twenty seven ready-to-use fact and dimension physical data objects and logical data objects created as part of BPM, SOA, and BAM installation
- Custom derived data objects created automatically when users enable metrics collection for a process in BPM or SOA
- Application roles at the composite level attached to data objects automatically
- Row-level security with custom BPM application role tokens, or calling out to the BPM security system for backward compatibility with BAM 11g
- Analytics data populated during deployment for dimensions and during runtime for measures
- Support for stream (inserts only, not persisted), archived stream (inserts only, persisted), and archived relation (all modifications, persisted) data objects.
- Support for Business Process Management Notation (BPMN), Business Process Expression Language (BPEL), Oracle Human Workflow(HWF), and Case Management (CASE) processes
- Support for any data source that can be configured in Oracle WebLogic Server
- Data simulation using `loadgen`

See the following for more information:

- See [Working with Data Objects](#) for information on working with data objects in the Designer role.
- See [Integrating with Oracle Business Process Management](#) for more information on ready-to-use data objects for BPM.
- See [Integrating with Oracle SOA Suite](#) for more information on ready-to-use data objects for SOA.
- See [Creating and Managing Oracle BAM Data Objects](#) for information on creating and managing data objects in the Administrator role.
- See [Using the Loadgen Utility](#) for more information about data simulation.

## BAM Dashboards

Oracle BAM ships with easy to use dashboards that can be customized to include various visualizations.

See the following for more information:

- [Creating Dashboards](#)
- [Integrating with Oracle Business Process Management](#)
- [Integrating with Oracle SOA Suite](#)

## BAM Main Pages and User Task Summaries

BAM Composer has three main pages, which correspond to the four main BAM user roles which is Viewers, Creators, Architects, and Administrators.

The main pages are as follows:

- The **Home** page is for [Viewers](#), who view data in dashboards and receive alerts
- The **Designer** page is for [Creators](#), who create dashboards and alerts
- The **Administrator** page is for [Architects](#), who create data objects, and [Administrators](#), who maintain BAM servers

See [Managing Oracle BAM Users](#) for more information.

### Viewers

Viewers cannot create or modify data.

Viewers perform these tasks:

- View dashboards, which display collections of tables and charts with real-time data
- Type parameter values in dashboards to filter data with different values, such as dimensions and date ranges
- Receive alerts, which notify users of out-of-range data
- View a history of recent alerts

See [Using Oracle Business Activity Monitoring as a Viewer](#) for more information on how to use the Home page.

### Creators

Creators perform a variety of Designer tasks.

Creators perform these tasks:

- All Viewer tasks, for the dashboards and alerts to which they have access
- Create projects, which contain sets of related BAM entities
- Add existing data objects to projects
- Create queries and KPIs to fetch data from data objects
- Create business views to display data in a variety of tables and charts
- Create dashboards to display collections of related business views
- Create alerts to initiate actions and notify users based on out-of-range data
- Create parameters to allow users to filter data using different values in dashboards, such as dimensions and date ranges

### Architects

The Architects mainly use the Administrator page to perform management and configuration tasks.

Architects perform these tasks:

- Create and manage BAM data objects
- Configure Enterprise Message Sources (EMSs)
- Run data simulators
- Import, export, and migrate data, data object metadata, and Enterprise Message Sources (EMSs)
- Use BAM web services for managing data objects

## Administrators

Administrators mainly use the Administrator page for monitoring, configuring, and testing BAM.

Administrators perform these tasks:

- All Viewer, Creator, and Architect tasks, for the projects to which they have access
- Monitor continuous queries
- Monitor viewsets
- Configure BAM server properties
- Configure BAM server availability
- Configure BAM diagnostics
- Configure BAM security
- Run BAM test-to-production scripts
- Import, export, and migrate data, data object metadata, EMSs, projects, and project entities
- Use BAM web services for managing data objects and firing alerts

## Navigating in BAM Composer

Learn how to navigate in and operate BAM Composer.

The following topics illustrate the navigation process:

- [Browser Settings](#)
- [Logging In](#)
- [Selecting the Home\\_ Administrator\\_ or Designer Page](#)
- [Using the Navigation Pane](#)
- [Expanding and Collapsing Panes](#)
- [Selecting a Project](#)
- [Showing and Hiding UI Text](#)
- [Closing Tabs](#)
- [Setting Language\\_ Time Zone\\_ and Accessibility Preferences](#)

## Browser Settings

You must configure your browser to share these settings that enable you to run Oracle BAM in the most optimal conditions.

The following browser settings are recommended for use with Oracle BAM:

- Allow pop-ups from the BAM server. Pop-up windows from the BAM server host name must be permitted for dashboards to be opened in the Viewer (the Home page).
- Open Separate Browser Windows For New URLs.

As a default behavior of many browsers, if the browser is open and you click a link in an email or an alert, the link opens in the most recently used browser window. You can change this behavior so that a separate browser window opens each time a link outside of the current browser is clicked.

Changing this setting is recommended if you often have browsers running that should remain open when you click links in emails or alerts.

- Better quality printing results if you include background colors and images, which is not a default setting.
- To ensure that the most recent file versions are displayed, disable caching to automatically check for updated files.
- You can alter your browser security settings to allow Oracle BAM to hide the address bar.

Supported browser versions are listed on the Oracle Technology Network web site at <http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>. Click the link for System Requirements and Supported Platforms for Oracle Fusion Middleware 14c (14.1.2). Then click the Client Certification link on the spreadsheet.

## Logging In

**To access BAM Composer:**

1. Type a URL of the following form in your browser:

```
http://host:port/bam/composer/
```

For example:

```
http://www.example.com:7001/bam/composer/
```

The default port is 7001.

2. Type your user name in the **User ID** text box.
3. Type your password in the **Password** text box.
4. Click **Sign In**.

BAM Composer opens in the browser.

## Selecting the Home, Administrator, or Designer Page

On the right-hand side of the top banner of BAM Composer are links to the top-level pages. Which of these pages are accessible depends on your user role. For more information, see [BAM Main Pages and User Task Summaries](#). The top-level pages are:

- **Home** — Use this page to view dashboards and alerts.
- **Administrator** — Use this page to create data objects and manage BAM servers.
- **Designer** — Use this page to create dashboards, alerts, and components thereof.

See [Using Oracle Business Activity Monitoring as a Viewer](#) for more information on how to use the Home page.

## Using the Navigation Pane

The Designer and Administrator pages have a left navigation pane, which is displayed by default.

The navigation pane lists entity categories. To display all the entities in a category, click the arrow to the left of the category. For example, to display all dashboards, click the arrow to the left of the **Dashboards** category.

The **Data Objects** category can have folders under it. To display the contents of a folder, click the arrow to the left of the folder icon.

Across the top of the navigation pane are icons for the most basic operations. To perform an operation on an entity, select the icon for the entity and then select the icon for the operation.

- **Open** — Opens the selected dashboard for viewing data. Applies only to dashboards and is present only on the Designer page.
- **Create** — Creates a new entity within the selected category or folder and opens a new editor tab for it. On the Designer page, for data objects only, adds data objects to the current project. This is the default operation when you click on a category with no entities under it.
- **Edit** — Opens the selected entity in a new tab for viewing and editing its configuration. This is the default operation when you click on an entity name instead of an entity icon.
- **Delete** — Deletes the selected entity. If a data object folder is selected, deletes the folder and its contents.
- **Refresh** — Updates the navigation pane with entities created by other users and to which you have access.

## Expanding and Collapsing Panes

Expand and collapse icons appear on the edges of panes that can be hidden. These can be vertical:



Or horizontal:



You can collapse a pane to provide more space for other parts of the window. You can expand the pane again if you need to select something in it.

When you open a dashboard, the navigation pane is collapsed to give you more room to edit the dashboard. You can expand the navigation pane by clicking the Expand icon on the left edge of the dashboard.

## Selecting a Project

On the Designer page, above the navigation pane, is the name of the current project. The default project is Process Analytics. See [Planning and Creating Projects](#) for more information about how to create or open a project.

## Showing and Hiding UI Text

Some editor tabs have explanatory text at the top. To make more space for viewing the settings, you can hide this text by clicking the X in the top right corner.

## Closing Tabs

After you have opened tabs using the **Edit** operation, you can close tabs in one of the following ways:

- To close the current tab, click the X on the tab.
- To close the current tab, click the X in the top right corner of BAM Composer.
- To close all tabs, click the down arrow next to the X in the top right corner of BAM Composer and select **Close All** from the menu.
- To close all except the current tab, click the down arrow next to the X in the top right corner of BAM Composer and select **Close Others** from the menu.

## Setting Language, Time Zone, and Accessibility Preferences

**To change personal settings:**

1. On the right side of the top banner of BAM Composer, click **Preferences**.
2. On the left side of the Preferences window, select **Language**.
3. Select a language from the **Language** drop-down list.

No Preference defaults to English.

4. Select a time zone for dashboards and alerts from the **Time Zone** drop-down list.

No Preference defaults to the server time zone. However, in dashboards and alerts, dates and times are converted to the user's preferred time zone.

### **Note:**

Make sure that times on all Oracle BAM Server instances in a cluster are synchronized to UTC. Time differences can result in event losses for continuous queries.

5. On the left side of the Preferences window, select **Accessibility**.
6. Select an option from the **Mode Settings** drop-down list:
  - No Preference

- Enable screen reader mode
- Disable screen reader mode

No Preference defaults to screen reader mode disabled.

7. Select an option from the **Contrast Settings** drop-down list:

- No Preference
- Use high contrast
- Use normal contrast

No Preference defaults to normal contrast.

8. Select an option from the **Font Settings** drop-down list:

- No Preference
- Use large fonts
- Use normal fonts

No Preference defaults to normal fonts.

9. Click **Apply**, then click **OK**.

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

## Sample BAM Projects

Oracle BAM provides a rich set of ready-to-use dashboards for all major components of Oracle Business Process Management (BPM) and Oracle Service-Oriented Architecture (SOA) Suite.

These dashboards can be used as is or modified as per your organization's needs. See [Integrating with Oracle Business Process Management](#) and [Integrating with Oracle SOA Suite](#) for more information.

For additional sample BAM projects that you can download and import, see <http://www.oracle.com/technetwork/middleware/bam/learnmore/index.html>.

## Installing or Upgrading to BAM Release 14c

You can upgrade Oracle BAM from previous releases to 14c.

You can upgrade the following from BAM 11g to BAM 14c using the BAMCommand `import` command:

- Data object data
- Data object metadata
- Enterprise Message Source (EMS) metadata

See [Import](#) for more information.

For information on installing BAM 12c, see Configuring the Oracle Business Activity Monitoring Domain in *Installing and Configuring Oracle SOA Suite and Business Process Management*.



**Note:**

Oracle BAM does not support dynamic clusters and server templates in Release 12.2.1.2.0. For more information, refer Defining Server Templates in *Installing and Configuring Oracle SOA Suite and Oracle Business Process Management*.

For information on upgrading BAM, see Upgrading to Oracle SOA Suite with Business Activity Monitoring (BAM) 12c (12.2.1.1) from a Previous 12c Release in *Upgrading Oracle SOA Suite and Business Process Management*.

Supported system configurations are listed on the Oracle Technology Network web site at <http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>.

Oracle Business Process Management 12c and Oracle SOA Suite 12c depend on BAM 12c for analytics. See [Integrating with Oracle Business Process Management](#) and [Integrating with Oracle SOA Suite](#) for more information.

## Moving BAM to Another Environment

You can move BAM from a source environment, such as a test environment, to a target environment, such as a production environment.

Oracle Fusion Middleware provides scripts to copy an Oracle home and domains, as well as the configuration of BAM, from one environment to another. As part of the procedure, you extract a move plan from the source environment, which contains properties that you can then edit to provide the values for the target environment.

For more information, see Moving from a Test to a Production Environment in *Administering Oracle Fusion Middleware*.

# 2

## Using Oracle Business Activity Monitoring as a Viewer

One of the roles of Oracle BAM is that of a Viewer. Learn how to use the Oracle Business Activity Monitoring (Oracle BAM) Home page, also called the BAM Viewer, to view dashboards and alerts.

This chapter includes the following sections:

- [Understanding Dashboards and Alerts](#)
- [Using the Home Page](#)
- [Viewing Dashboards](#)

### Understanding Dashboards and Alerts

A *dashboard* is a collection of related graphs, tables, and gauges, called *business views*. Dashboards can display point-in-time or real-time information (active data).

A *tabbed dashboard* contains other complete dashboards, which you view by clicking the tabs. A *thumbnail* is an image showing the entire dashboard in miniature, which you can click to open the dashboard.

An *alert* performs one or more actions when launched by an event and filtered by one or more conditions. An *event* can be based on a date or time, a change in data, or a manual trigger. A *condition* restricts the alert to a date or time range or to a specific day of the week. An *action* can send a notification, perform a data operation, or invoke an external application. For example, an alert might send you an e-mail every week with a link to a dashboard that displays sales totals.

Oracle BAM integrates with Oracle Application Development Framework and Oracle JavaScript Extensibility Toolkit to provide various business views in dashboards. While ADF dashboards are compatible with all business views, JET dashboards only support some of the following business views:

- An **Area** view shows a graphed line of values with the area filled in for visual comparison. JET dashboards display Area and Stacked Area business views.
- A **Bar** view shows vertical columns to represent summarized values. JET dashboards display Bar and Stacked Bar business views.
- A **Horizontal Bar** view shows horizontal columns to represent summarized values. JET dashboards display Horizontal Bar and Stacked Horizontal Bar business views.
- A **Line** view shows a graphed line of values or compared values. JET dashboards display Line and Stacked Line business views.
- A **Pie** view displays values in segments of a circle. JET dashboards display the simple Pie business view.
- A **Combo** view shows a combination of values as areas, bars, and lines in the same view. JET dashboards display Combo business views.

- A **Scatter** view shows X,Y coordinates on a rectangular graph. JET dashboards display Scatter business views.
- A **Bubble** view is like a Scatter view, except that X,Y coordinates appear as bubbles that vary in size. JET dashboards display Bubble business views.
- A **Treemap** view shows hierarchical data groupings arranged in rectangles within rectangles. For example, a hierarchy might consist of the country, state or province, and city. The count of sales might determine the size of the rectangles, and the sum of sales might determine their colors.
- A **Point Theme Geo Map** represents cities as points on a map.
- A **Color Theme Geo Map** represents geographic regions as colors on a map.
- A **GPS Theme Geo Map** represents moving locations tracked by GPS as points on a map.
- A **List** displays a table you can sort. JET dashboards display List table business views.
- An **Action List** has values in some or all table columns that are editable. It also displays data changes without requiring a refresh. JET dashboards display Action List table business views.
- A **Collapsed List** displays table rows of aggregated data values grouped by one or more dimensions. It may also have color-coded thresholds for data values. JET dashboards display Collapsed List table business views.
- A **Pivot Table** combines rows and columns to display a multi-dimensional view of aggregated data values. A Pivot Table is summarized vertically and horizontally for columns and rows.
- A **KPI Watchlist** displays a table of the following information for one or more key performance indicators (KPIs):
  - **KPI** lists KPI names.
  - **Risks** list the color-coded status of risk indicators associated with each KPI.
  - **Actual** shows the most recent KPI value.
  - **Today** shows a line graph of the last 10 KPI values. Older KPI values are on the left and newer values are on the right.
  - **Target** shows a horizontal bar that compares the current KPI value to a target value.
- A **Dial Gauge** view indicates the current value in a curved numeric marked gauge. JET dashboard displays the Dial Gauge business view.
- A **Dial with Thresholds** view is similar to the dial gauge and indicates the current value in a curved numeric marked gauge with color-coded low, medium, and high ranges.
- A **Statusmeter Gauge** indicates the current value in a straight numeric marked gauge with color-coded low, medium, and high ranges.
- A **Vertical Statusmeter** gauge indicates the current value in a vertical numeric marked gauge with color-coded low, medium, and high ranges.
- An **Arrow Gauge** is an arrow shape that is color-coded according to whether the value is low, medium, or high. The direction of the arrow indicates whether the current value is higher (up), the same (right), or lower (down) than the previous value.
- A **Circular Status Meter Gauge** indicates the current value in a gauge representing the numerical sum along a circular axis.
- A **Horizontal Status Meter Gauge** indicates the current value in a gauge representing the numerical sum along a horizontal axis.

 **Note:**

Both the Circular and Horizontal Status Meter Gauges are preview features. To be able to use them, you must enable them when you install Oracle BAM.

When you open JET-based dashboards that contain business view types or subtypes that are not supported by JET, you see the following error message:

VIEW\_NOT\_SUPPORTED: The *{Business View}* type business view *{Business View Name}* is not supported.

If a specific business view subtype is unsupported, you see the following error message:

VIEW\_SUBTYPE\_NOT\_SUPPORTED: The *{Business View}* type business view *{Business View Name}* of subtype *{Business View Subtype}* is not supported.

## Using the Home Page

The Oracle BAM Home page, also called the BAM Viewer, is where you view the dashboards and alert history to which you have access.

This section contains the following topics:

- [Selecting a Project](#)
- [Choosing Carousel or List View for Most Recent and Favorites](#)
- [Opening a Dashboard](#)
- [Adding a Dashboard to the Favorites List](#)
- [Opening a Dashboard when Oracle BAM Is Started](#)
- [Viewing Alert History](#)

## Selecting a Project

A project contains a group of related dashboards. The current project determines which dashboards you can open.

The project name appears in the top left corner of the Dashboards tab, which is on the right side of the Home page. BPM Analytics is the initial and default project in BAM.

**To open a project:**

1. Click the down arrow to the right of the project name.
2. Select **Open Project** from the pop-up menu.

The Open Project dialog appears.

3. Click the name of the project you want to open.

The project opens, and the project name appears in the top left corner of the Dashboards tab.

## Choosing Carousel or List View for Most Recent and Favorites

To change how the Most Recent and Favorites lists are displayed, select one of the following:

- **Carousel View** — Displays each list separately as a horizontal series of thumbnails, with a horizontal slider for selecting a position in the list. To update the list with the latest dashboards, click **Refresh**. To open a dashboard, double-click its thumbnail.
- **List View** — Displays both lists as a vertical series of table rows, with a thumbnail, dashboard information, and icons in each row. To update the list with the latest dashboards, click **Refresh**.

To expand the list, click **Load More Items**. To open a dashboard, click its thumbnail.

## Opening a Dashboard

You can open a dashboard from the Dashboards tab, the Most Recent list, or the Favorites list.

- From the Dashboards tab, click the dashboard name from the drop-down list and click the **Open** icon, or right-click it and select the **Open** menu item. The dashboard opens in a new browser window.
- From the Most Recent or Favorites list, click the dashboard thumbnail in the Carousel View once to bring it to the center and again to open the dashboard. Click the thumbnail once in the List View to open the dashboard.

The dashboard opens in a new browser window.

## Adding a Dashboard to the Favorites List

You can add dashboards that you need to find quickly to the Favorites list.

- From the Dashboards tab, click the dashboard icon and click the **Favorites** (star) icon, or right-click it and select the **Favorites** menu item.  
To see the dashboard in the Favorites list in the Carousel View immediately, click the **Refresh** icon for the Favorites list.
- From the List View, click the **Favorites** (star) icon for the dashboard. The icon changes from grayed out to gold.

## Opening a Dashboard when Oracle BAM Is Started

You can choose which dashboards to open automatically when BAM is started.

- From the Dashboards tab, click the dashboard icon and click the **Start on Launch** icon, or right-click it and select the **Start on Launch** menu item.
- From the Most Recent or Favorites list in the Carousel View, click the dashboard thumbnail, then click the **Most Recent** or **Favorites** header and select the **Start on Launch** menu item.
- From the List View, click the **Start on Launch** icon. The icon changes from green to grayed out.

When you log into Oracle BAM and go to the Home page, the dashboard is automatically launched in another browser window.

## Viewing Alert History

You can view a recent history of alert activity on the Alerts tab. The Alert History list is divided into three sections: Today, This Week, and Older.

For each alert instance, the Alert History list displays the alert name, any associated messages, and the time and date that the alert was triggered.

You can display alerts for all users. See [Filtering the Alert History List](#).

A successful alert is shown with a green checkmark next to the message. An unsuccessful alert is displayed with a red **x** icon and a message indicating how the alert failed at the time of loading or processing. Click the **x** icon for additional information about the error.

## Filtering the Alert History List

If you are interested in specific alerts, you can filter the Alert History list by alert name or a date range.

### To filter the Alert History list:

1. On the Alerts tab, click the arrow to the right of **Filter by**.

Additional filter settings appear.

2. To filter by alert name, type the name in the **Alert Name** text box and press Enter. Typing part of a name lists all alert names containing that part. You can then select a name from the list.

The Alert History list shows only the alerts matching the name you entered.

3. To display alerts for all users, check the **All Users' Alert History** box. This check box is visible only to users that have an administrator role.

The Alert History list shows alerts sent and received by all users.

4. To filter by date, type dates in the **From** and **To** text boxes, or click each Select Date icon and select a date from the pop-up calendar.

The Alert History list shows only the alerts in the date range you specified.

## Clearing the Alert History List

When many alerts are actively launching and the Alert History list becomes long, you can clear the list.

### To clear the Alert History list:

1. On the Alerts tab, click **Clear All**.

A message asks you to confirm clearing the Alert History list.

2. Click **OK**.

The Alert History list is deleted. Only new alerts launched after clearing appear in the list.

## Viewing Dashboards

A dashboard opens in a separate browser window. Dashboard windows have their own set of viewing options, separate from those of the Home page.

This section contains the following topics:

- [Dashboard Parameters](#)
- [Filtering Runtime Data in a View](#)
- [Enabling Active Data in a View](#)

- [Changing Dimensions in a View](#)
- [Exploring View Data](#)
- [View Actions](#)

## Dashboard Parameters

A *parameter* is a variable representing the value of a data field. A *prompt* is a request for the user to specify the value of a parameter. For example, you can display product sales by country in a dashboard and prompt the user to choose the country parameter value.

You can pass in the parameter values you want to specify as part of dashboard URL.



### Note:

This section is not applicable to Oracle JavaScript Enterprise Toolkit (JET)-based dashboards.

Parameter types are as follows:

- A *value parameter* for a non-datetime field is a text box that accepts only one value.
- A *value parameter* for a datetime field has drop-down lists for the year, month, day, hour, minute, and second, and a **Refresh** button.
- A *list parameter* can present and in some cases accept multiple values with check boxes in a drop-down list.

## Saving a Dashboard to an HTML File

To display a static version of the dashboard that you can save as an HTML file, click the **Save Offline in a New Tab** icon in the top right corner of the dashboard.

The HTML file version opens in a new browser window. You can save it using the **Save Page As** command in the browser.

## Filtering Runtime Data in a View



### Note:

This section is not applicable to Oracle JavaScript Enterprise Toolkit (JET) dashboards.

You can apply conditions to the data displayed in a view. For example, you can display only processes with a priority of 2 or greater.

You cannot filter runtime data if Active Data is enabled.

**To filter runtime data in a view:**

1. Click the **Runtime Filters and Active Data Properties** (gear) icon in the top right corner of the view.
2. Select the **Filters** tab.

3. Select the type of filtering from the **Filters** drop-down list: By Number, By String (text) or By DateTime.

 **Note:**

Filtering by numeric fields is supported with only two digits of precision. If the value in the field is actually more precise, it cannot be filtered exactly. For example, you can filter by comparing to a value of 12.34, but not a value of 12.345.

4. Check the boxes to the left of the data fields to filter.
5. For each checked data field, select an operator from the drop-down list. Select a comparison operator, **null**, or **not null**. For filtering by number or datetime, you can also select **between**. For filtering by string or datetime, you can also select **like** or **not like**.

 **Note:**

The **like** and **not like** string searches are case sensitive and require a wildcard. For example, the search **US** would not yield the same results as **US%** or **us%**.

6. If you selected a comparison operator, **like**, or **not like**, type a value to which to compare the data. If you selected **between**, type two values to define a range. For filtering by datetime, you can click the **Select Date and Time** icon instead of typing.
7. Click **OK**.

A filter icon appears in the top right corner of the view to indicate that data is filtered.

## Modifying Active Data Properties in a View

Active Data continuously aggregates data or displays a segment of data within a defined time window. As time passes in a time window, older data is removed from the view and newer data is added.

Active data is supported only for basic Area, Bar, Line, Combo, Pie, Scatter, and Bubble views. None of the other variations of these view types support active data. However, active data is supported for all variations of Gauge views and for Point Theme and GPS Theme Geo Map views.

Active data is not supported for Horizontal Bar, KPI Watchlist, Treemap, or Color Theme Geo Map views.

Active data is not supported for external data objects.

The time zone in your Preferences does not control the time slice for Active Data. Active Data time is based on UTC.

**To configure Active Data:** **Note:**

Active Data properties can be modified at run time only if runtime modification is enabled at design time. See [Using Active Data](#) for more information.

1. Click the **Runtime Filters and Active Data Properties** (gear) icon in the top right corner of the view.
2. Select the **Make Active** tab.
3. To make data aggregation (computation of averages, sums, and so on) active, specify the following settings:
  - **Active Data Collapsing** — Optionally check this box to enable continuous data aggregation.
  - **Interval** — Specify how often the aggregation result is updated in the view.
  - **Units** — Select Day, Hour, Minute, or Second for the Interval.

If you change the **Interval** and **Units**, the new data snapshot is rendered only when the interval ends. For example, if the interval is set to one minute, the new snapshot is shown after one minute and then active data collapsing occurs.

4. To define a time window, specify the following settings:
  - **Use a time window** — Optionally check this box to define a period over which and a frequency at which data is updated. For example, you could update the number of calls completed in the last hour every ten minutes.
  - **Sliding Range Based on** — A date or time column upon which the rolling window calculation is based. For example, choosing Call Start Time enables the system to determine whether a given call is within the rolling window period.

This setting is displayed only if the selected data object is a relation. It is not supported for stream data objects.

 **Note:**

In JET—based views, the sliding tool may shrink if you adjust it manually as and when active data keeps populating in the view. This might invalidate the time window you have selected resulting in the business view not displaying any data. To correct this, set the business view to refresh automatically. Alternatively, you can set the entire dashboard to auto refresh.

- **Range Length** — The period over which data is displayed, from an amount of time in the past to the present. Examples of ranges are the last 30 minutes or the last 24 hours.
  - **Update Interval** — The frequency at which the result within the rolling window is updated.
  - **Units** — Select Day, Hour, Minute, or Second for the Range Length and Update Interval.
5. Click **OK**.

## Changing Dimensions in a View

 **Note:**

This section is not applicable to Oracle JavaScript Enterprise Toolkit (JET) dashboards.

Dimensions are the categories by which numeric data field values are grouped for aggregations such as averages or sums. For example, sales totals can be grouped by geographic region or by salesperson.

You cannot change dimensions if Active Data is enabled.

You cannot change dimensions in a view based on a query that uses a time series. See [Using a Time Group or a Time Series](#) for more information.

### To change dimensions in a view:

1. Click the **Change Dimension** (down arrow) icon in the top right corner of the view.
2. Check the boxes to the left of the dimensions to include.  
Datetime fields are not supported.
3. Click **OK**.

The X-axis of the view changes according to the dimensions you selected.

## Exploring View Data

There are four types of drilling in Oracle BAM views:

 **Note:**

Only Drilling Through is applicable to Oracle JavaScript Enterprise Toolkit (JET) dashboards.

- **Drilling Down** — displays hierarchical data one level down. For example, drilling down in a sales total for a country can display a breakdown of sales totals for states or provinces. To drill down:
  - Right-click the data point, bar, node, pie slice, or Collapsed List row and select the drill level from the menu.
- **Drilling Up** — displays hierarchical data one level up. After drilling down, you can go back up. To drill up:
  - Right-click the data point, bar, node, pie slice, or Collapsed List row and select the drill level from the menu.
  - Right-click the data point, bar, node, pie slice, or Collapsed List row and select **Home** to go to the initial drill level of the view.
- **Drilling Through** — displays actual data at the lowest level of the hierarchy. For example, drilling through in a sales total can display individual sales. To drill through:

- Right-click the data point, bar, pie slice, Collapsed List row, or gauge, and select **Show Details** from the menu.
- **Drilling Across** — displays a target dashboard or URL in a new browser window. Not all views have drill-across targets. To drill across, right-click the view and select the target name. Left-click drilling is supported for graph views if enabled at design time. See [Configuring Drill-Across Targets](#) for more information.

Active data stops during drilling and then restarts.

 **Note:**

If you drill down in an Area, Line, or Combo view to a level at which only one data point is included, no line is drawn, and you cannot drill up. To restore the view, remove the filter that the drilling applies. See [Filtering Runtime Data in a View](#) for more information.

## View Actions

If one or more custom actions are configured for a view, a square icon containing a green triangle appears in the upper right corner. Click the icon to display the names of the actions, then select an action.

All views can have these action types:

- **Open Dashboard** opens a specified dashboard in a new browser window.
- **Open URL** opens a specified URL in a new browser window.
- **Refresh View** redisplay the view with the latest data and properties.
- **Show Confirmation Message** displays a confirmation message dialog before performing the action.

Action List views can have these additional action types:

- **Insert** inserts data.
- **Update** updates data.
- **Save** saves all changes made to the Action List.
- **Delete** deletes selected rows from the Action List.

# 3

## Tuning Oracle Business Activity Monitoring 14c

Oracle Business Activity Monitoring 14c provides methods that will ensure Oracle BAM's optimal usage while curbing common errors.

This chapter includes the following sections:

- [Tuning Schema Design](#)
- [Tuning Dashboard Design and Deployment](#)
- [Tuning Query Design](#)
- [Tuning Data Archiving and Purging](#)
- [Tuning Scalability and Performance](#)
- [Real-time Data Sources for Oracle BAM](#)
- [Tuning Production Roll-out](#)

### Tuning Schema Design

Oracle BAM 14c supports a star schema. It supports joins through Logical Data Objects (LDOs). An LDO allows you to join one FACT table with multiple DIM or Dimension tables, which allows you to run queries on top of the LDO.

If Active Data is used on top of an LDO with Archived Relation, some considerations need to be taken into account to achieve the best performance. These conditions are outlined as follows. Please note that these are best practices to help enhance your performance when you switch to 14c and must be treated as recommendations.

- Oracle recommends that you mark your DIM table as 'Slow Changing Dim' if the changes to the DIM table are infrequent (once in a day or so). When you mark it with 'slow-changing dim', a join-memory enhancement comes into play that reduces the amount of heap required and also increases the query start up time.
- Oracle recommends that you keep the DIM table size reasonable (for example, less than 10k – only used for look up) and batch updates to DIM tables to avoid frequent updates so you can mark the DIM table as 'Slow Changing Dim'. In the case where the dim table is growing at the same pace as the fact table, it is better to keep the dimension columns in the FACT table.
- Oracle recommends that you only join the FACT table with required DIMs as multiple DIMS requires complex n joins. Consolidate DIM tables to reduce number of joins.
- Oracle recommends that you design an LDO with filters to reduce the amount of data returned. This reduces the TEMP tablespace required to run the archiver query.
- Oracle recommends that you use Calculated Fields in your query to reduce the number of columns required in the FACT DO. Keep only what's required as columns in the FACT table. This will help Active and CQL template queries to be more efficient as they do not need to bring additional columns into memory.

- If you are using 11g Oracle DB, Oracle recommends that you use Secure Files to improve performance.
- Oracle recommends that you create indexes for key fields. You can use the BAM Data Object to create indexes. See [Selecting an Index Strategy](#) for more information.

## Tuning Dashboard Design and Deployment

Oracle Business Activity Monitoring 14c comes with enhanced Dashboard capabilities for optimal performance.

The following best practices are recommendations too ensure optimal performance. For optimal dashboard performance, please treat the following best practices as recommendations.

- Ensure that a dashboard is designed for a specific user group. Do not try to create an all-purpose dashboard.
- Create a report which is “management by exception”.
- Allow the people handling various user roles to understand the health of the operations at a glance; then provide drill-down.
- Use the Logical Data Object filter and dashboard parameters to filter data before dashboard is loaded.
- Limit a dashboard to 4-6 views or one view with multiple tabs.
- Make effective use of Dashboard Action button present in each Dashboard cell.

Use summary or aggregation as much as possible and provide drill down to detail records from there to avoid many detail records.

- Table and list views have some performance limitations. Oracle recommends that you limit List View listings to 2000 records, although the Oracle BAM system has an upper limit of 10420 records. If applicable, define a dimension or time hierarchy which allows drill down until you reach the node that returns detailed records. Also ensure that you increase the memory settings (Oracle recommends up to 6GB) for the BAM server while retrieving many records.
- Use Oracle Tech Network demos for design and best practices.
- Choose active data interval and throttling carefully because it has memory implications.

## Tuning Query Design

Active and non-active queries are subject to best practices in Oracle BAM 14c.

For optimal performance of your business queries, Oracle recommends the following best practices.

- For tactical (non-active), queries, examine the query string generated by the business query and run the DB AWR report to tune the index or change your query to improve performance.
- For an active query, always specify a window size (range) because these reduce amount of memory required. Also, use the output window (sliding) judiciously to collapse results and to not overwhelm the dashboards.
- Use the aggregation query and bind it to a chart. Then perform drill down to detail records. This organizes the data into smaller chunks and improves performance.

- Test your query with your production data set to ensure that the performance is reasonable.
- Familiarize yourself with the differences between scheduled query and real time active query and use them as appropriate.

## Tuning Data Archiving and Purging

Oracle BAM 14c recommends some best practices for archiving and purging data for optimal performance.

- Keep only the necessary data in your Data Object. Any data that is relevant for operational monitoring and taking actions is a candidate for data object storage.
- Do not keep older data for historical purposes in the same data object. If you need the historical data to be available, transfer the data to another data object that is used for specific reporting.
- Schedule the Data Retention at non-peak hours using alerts. You can shut down all BAM Viewsets and Continuous Queries if you're an administrator during this task as they hold state in memory.
- If it is a requirement to archive the data prior to purging, then use an Oracle Data Integrator service with BAM 14c.
- Use alerts to control filter-based purging. Custom actions can also be provided to the alert to perform more customized purging.
- Use the Database Partition capability to improve performance if purging and data injections are happening concurrently. For more information, see *Partitioning Concepts in the Oracle Database VLDB and Partitioning Guide*.

## Tuning Scalability and Performance

Oracle BAM 14c recommends some best practices for optimal performance and scalability.

- Oracle BAM 14c is designed for High Availability and Scalability. High Availability and Scalability can be increased by adding additional servers. Additional servers increase High Availability by providing healthier servers to failover to in the event one server fails. As long as one BAM server is running, BAM will continue to function. Additional servers provide greater throughput as work is load-balanced across all available servers in the BAM cluster. For more information, see the *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite*.
- For input scalability, multiple EMS can be started on different servers to improve throughput.
- Customers should consider using batching capabilities to improve performance. See [Batching Operations to Group Multiple Events Into a Single Operation](#) for more information.
- CQL queries are load balanced using project as a base. All queries for a given project would reside on the same server. When a server crashes, CQL queries are moved onto a different server.
- Limit the number of CQL queries as they tend to require large quantities of heap memory.
- Ensure that Oracle BAM is in its own cluster of dedicated machines. If the solution includes BPEL or BPM, use separate machines for BPEL and its storage to avoid contention.
- For High Availability solutions, make sure that your servers are on the same network.

- Ensure that the application context is the same for Oracle BAM Web Services if the web service interface is used for pushing data into BAM.
- Tune the database completely, including adding indexes, partition table by day, parallel execution, and use DBIM (Database In-Memory). DBIM accelerates access to the data because only the required columns (columnar storage) are visited. In addition, for cases where filtering occurs but the filter is not selective, DBIM is a little like having an index per column. For the queries that perform simple aggregations, DBIM can also accelerate aggregation.

## Real-Time Data Sources for Oracle BAM

Oracle BAM provides real-time data sources that you can use in the Oracle WebLogic Server (WLS).

These data sources are as follows.

- Oracle BAM Adapter
- Oracle Web Services
- Java Message Service
- Oracle® Data Integrator
- Enterprise Java Beans

## Tuning Production Roll-out

Oracle recommends that you use the suggested configuration and load testing checklist before production.

The checklist outlines some configurations and load testing.

- Ensure that your Java Messaging Service and Automatic Service Management configuration is in accordance with the Oracle Enterprise Deployment Guide.

Summary of JMS Servers

JMS servers act as management containers for the queues and topics in JMS modules that are targeted to them.  
This page summarizes the JMS servers that have been created in the current WebLogic Server domain.

Customize this table

JMS Servers (Filtered - More Columns Exist)

Name	Persistent Store	Target	Current Target	Health	Scope	Domain Partitions
BamAlertEngineJmsServer_auto_1	BamAlertEngineJmsFileStore_auto_1	bam_server1 (migratable)	bam_server1		Global	
BamAlertEngineJmsServer_auto_2	BamAlertEngineJmsFileStore_auto_2	bam_server2 (migratable)	bam_server2		Global	
BamCQServiceJmsServer_bam_server1	BamCQServiceJmsFileStore_bam_server1	bam_server1	bam_server1		Global	
BamCQServiceJmsServer_bam_server2	BamCQServiceJmsFileStore_bam_server2	bam_server2	bam_server2		Global	
BAMJMServer_auto_1	BAMJMSFileStore_auto_1	bam_server1 (migratable)	bam_server1		Global	
BAMJMServer_auto_2	BAMJMSFileStore_auto_2	bam_server2 (migratable)	bam_server2		Global	
BamPersistenceJmsServer_auto_1	BamPersistenceJmsFileStore_auto_1	bam_server1 (migratable)	bam_server1		Global	
BamPersistenceJmsServer_auto_2	BamPersistenceJmsFileStore_auto_2	bam_server2 (migratable)	bam_server2		Global	
BamReportCacheJmsServer_auto_1	BamReportCacheJmsFileStore_auto_1	bam_server1 (migratable)	bam_server1		Global	
BamReportCacheJmsServer_auto_2	BamReportCacheJmsFileStore_auto_2	bam_server2 (migratable)	bam_server2		Global	

- Ensure that the load balancer configuration is correct. Set the correct values for **WebLogic Plug-In Enabled**. Select **Yes** if the cluster receives requests from a proxy plug-in or `HttpClusterServlet`. A call to `getRemoteAddr` returns the address of the browser client from the proprietary `WL-Proxy-Client-IP` header, instead of the Web server. Select **No** to disable the `weblogic-plugin-enabled` parameter (`weblogic-plugin-enabled=false`) in the `config.xml` file. Select **Inherit** for this cluster to inherit the value selected for **WebLogic Plug-In Enabled** for the domain.

- Check the persistence configuration on Enterprise Messaging Service (EMS). Too many threads may cause contention. Too few may reduce throughput. Oracle recommends that you start with 20 threads and tune it based on the throughput.
- Check that your Application Roles are defined correctly.
- Check that the JVM Parameters have been tuned properly. For more information, see the Oracle Fusion Middleware Tuning Performance of Oracle WebLogic Server guide.
- Ensure that your dashboards display correctly with high production loads on different devices.

### Load Testing

Use the following pointers for load testing.

1. Use Oracle Enterprise Single Sign-on Logon Manager to simulate users coming on board. To do this, use the following test as a ballpark:
  - Start 20 virtual users in the frequency of 2 users every 30 seconds and run this for 20 minutes.
  - Next, stop all virtual users in the frequency of two users every five seconds.
  - Monitor this activity using the Viewset Monitoring capability.
2. Use EMS clients to pump data into the system. Start multiple EMS Servers. Ensure that each EMS Server is deployed to different managed servers.
3. Monitor throughput. Use the BAM Business Query by creating a time series on the x-axis based on the Data Object creation date column.
4. If you use the Java VisualVM tool, you can monitor the Heap, Memory, and CPU Usage.
5. Using the Automatic Workload Repository reports, check to see if there are dead locks. Also check the SQL execution time and determine if you need to create or modify indexes.
6. Run a Longevity Test. Run your system for a minimum of five days with EMS. Subject the system to continuous load, purge every day, and ensure that virtual users come and go.
7. Upon failover, bring down one server, monitor the performance, and once all performance issues are detected and fixed, bring the server back up. Manually, migrate services back to the server upon restart.

# Part II

## Using Oracle Business Activity Monitoring as a Designer

This part describes the Oracle BAM entities that comprise a project and are used to build dashboards and alerts.

This part includes the following chapters:

- [Planning and Creating Projects](#)
- [Working with Data Objects](#)
- [Creating Business Queries](#)
- [Creating KPIs](#)
- [Creating and Using Business Views](#)
- [Creating Dashboards](#)
- [Creating Alerts](#)
- [Creating Parameters](#)

# 4

## Planning and Creating Projects

This chapter describes how to plan and create a project in Oracle Business Activity Monitoring (Oracle BAM).

This chapter includes the following sections:

- [Planning a Project](#)
- [Choosing Bottom-Up or Top-Down Project Building Steps](#)
- [Creating a Project](#)
- [Opening a Project](#)
- [Renaming a Project](#)
- [Deleting a Project](#)
- [Securing a Project](#)
- [Managing Changes in a Project](#)

### Planning a Project

A *project* is a BAM entity that groups other related BAM entities.

Other BAM *entities* that you can add to or create in a project are:

- *Data Objects*, which contain the data to be monitored, such as external databases or streams of data from ongoing business processes. See [Working with Data Objects](#) for more information.
- *Business Queries*, which fetch and analyze the data to be monitored, such as sales figures for different countries. See [Creating Business Queries](#) for more information.
- *Key Performance Indicators (KPIs)*, which fetch and analyze especially important measures, such as current production rates. See [Creating KPIs](#) for more information.
- *Business Views*, which display data fetched by queries and KPIs in various formats, such as charts, tables, and gauges. See [Creating and Using Business Views](#) for more information.
- *Parameters*, which filter fetched data according to user input, such as the country for which to view sales figures. See [Creating Parameters](#) for more information.
- *Alerts*, which notify users of out-of-range fetched data, such as a spike in the average wait time at a call center. See [Creating Alerts](#) for more information.
- *Dashboards*, which display collections of related views, such as a pie chart of sales by country and a bar graph of sales by salesperson. See [Creating Dashboards](#) for more information.

These entities operate together to achieve a business monitoring goal. Before you create a project, you should plan what the entities in the project will do.



**Note:**

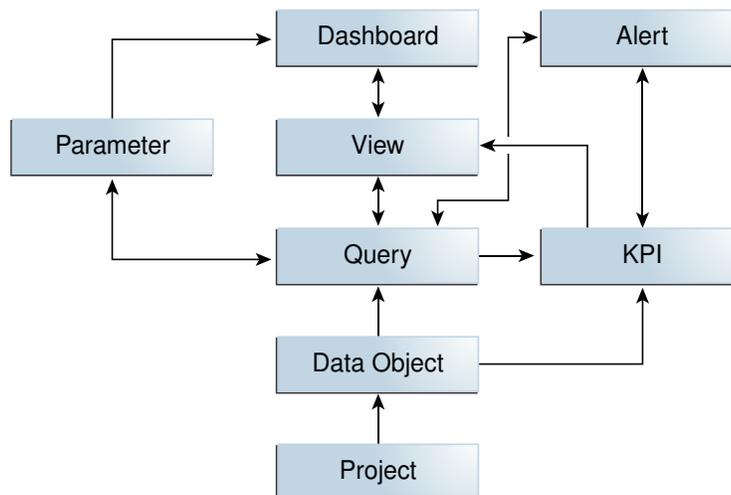
You can add any data object to which you have access to any project. You cannot move or copy other entities from one project to another. However, you can export a project to a ZIP file and import it into another BAM server. See [Using Commands with Oracle BAM](#) for more information.

## Choosing Bottom-Up or Top-Down Project Building Steps

You can create a project from the bottom up or from the top down, as long as certain aspects of the project are bottom up.

Figure 1 illustrates the steps for creating a project:

**Figure 4-1 Project Creation Steps**



Some aspects of building projects are always bottom-up. You must create your project and add data objects to it first. You cannot configure a query or a realtime KPI before you add a data object. Likewise, you cannot configure a scheduled KPI before you create the query on which it is based. You cannot create a KPI from a view. You cannot create a parameter from a dashboard.

However, other aspects of project building can be either bottom-up or top-down. You can create a dashboard, then a view, then a query, or you can create a query, then a view, then a dashboard. You can create a parameter before or after you create a query that uses it. You can create a query or KPI that triggers an alert action. Or you can create a stand-alone alert, then create an alert action in a query or KPI to trigger the stand-alone alert.

## Creating a Project

This section outlines the detailed procedure to create a Project.

Use the following procedure to create a project.

**To create a project:**

1. Go to the **Designer** page.

The project name appears in the top left corner of the BAM window.

Process Analytics is the initial and default project in BAM if BPM is configured in the same domain.

2. Click the down arrow to the right of the project name.
3. Select **Create** from the pop-up menu.

The Create Project dialog appears.

4. Type a **Name**, a **Display Name**, and optionally a **Description**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the project is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

5. Click one of the following buttons:
  - **Create BPM Example** — Creates a project with data objects for Oracle Business Process Management (BPM) processes.
  - **Create** — Creates a generic project with no data objects.

## Opening a Project

This section outlines the steps to open a Project.

Use the following procedure to open a project.

### To open a project:

1. Go to the **Designer** page.

The project name appears in the top left corner of the BAM window.

Process Analytics is the initial and default project in BAM if BPM is configured in the same domain.

2. Click the down arrow to the right of the project name.
3. Select **Open Project** from the pop-up menu.

The Open Project dialog appears.

4. Click the name of the project you want to open.

The project opens, and the project name appears in the top left corner of the BAM window.

## Renaming a Project

This section outlines the procedure to rename a project.

Use the following procedure to change the **Display Name** of a project.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

**To rename a project:**

1. Go to the **Designer** page.  
The project name appears in the top left corner of the BAM window.  
Process Analytics is the initial and default project in BAM if BPM is configured in the same domain.
2. If the project you wish to rename is not the current project, open the project you wish to rename. See [Opening a Project](#).
3. Click the down arrow to the right of the project name.
4. Select **Rename** from the pop-up menu.  
The Rename dialog appears.
5. Edit the **Display Name** of the project.  
You cannot change the internal **Name**. You can use this name to export a project using BamCommand.
6. Click **Save**.  
The project name in the top left corner of the BAM window is changed to the new name.

## Deleting a Project

This section outlines the procedure to delete a Project.

Use the following procedure to delete a project. All entities in the project except data objects are also deleted.

**To delete a project:**

1. Open the project you want to delete. See [Opening a Project](#).  
The project name appears in the top left corner of the BAM window.
2. Click the down arrow to the right of the project name.
3. Select **Delete** from the pop-up menu.  
A dialog asks you to confirm the project deletion.
4. Click **Yes**.  
The project is deleted. Process Analytics, the initial and default project in BAM, opens.

## Securing a Project

This section outlines the procedure to secure a project.

By default, BAM entities inherit security settings from the project in which they are created.

**To change security settings for a project:**

1. Go to the **Designer** page.  
The project name appears in the top left corner of the BAM window.  
Process Analytics is the initial and default project in BAM if BPM is configured in the same domain.
2. Click the down arrow to the right of the project name.

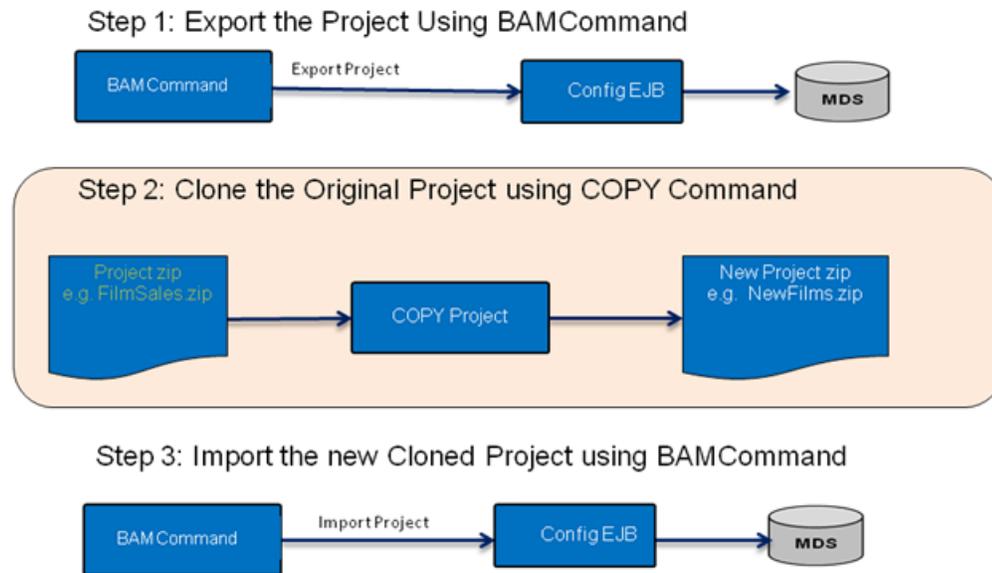
3. Select **Security** from the pop-up menu.  
The security tab for the project opens.
4. To add a role or group to whom you can explicitly grant or deny permissions, follow these steps:
  - a. Click the **Add** icon in the Grant Permissions or Deny Permissions table.  
The Add Application Roles, Groups, and Users dialog opens.  
See [Managing Oracle BAM Users](#) for information about how to add users to roles and groups.
  - b. Type a **Name** for the role or group you are adding.
  - c. Select from the drop-down **List: Application Role or Group**.
  - d. Click **Search** to populate the Available Members list.
  - e. To add a member to the Selected Members list, select the member and click the single right arrow.
  - f. To add all members to the Selected Members list, select the member and click the double right arrow.
  - g. To remove members from the Selected Members list, use the single and double left arrows.
  - h. When the Selected Members list is final, click **OK**.  
The Add Application Roles, Groups, and Users dialog closes, and the **Name** you specified appears in the table.
5. To remove a role or group, select the table row and click the **Remove** icon.
6. To grant permissions, select **Read**, **Write**, **Remove**, or **Security** for the users, roles, and groups listed in the Grant Permissions table.
7. To deny permissions, select **Read**, **Write**, **Remove**, or **Security** for the users, roles, and groups listed in the Deny Permissions table.
8. Click **Save**.

## Copying a Project

1. In order to Clone a project, download the `bam-impexp.copyproject.jar` and `copyproject.sh` files from Oracle Tech Network and copy them to the `<FMW_HOME>/soa/bam/lib` and `<FMW_HOME>/soa/bam/bin` locations respectively.
2. If your project has KPIs, you must apply Patch 19153993 as a fix for bug 18897246. Previously, the KPI Watchlist Data Control entry did not have the project name in its ID which meant that if you had two or more projects with the same KPI watchlist names, the KPI watchlists and business views would not function correctly for the new cloned project. The patch has produced a fix for this issue.

### Cloning a Project

Cloning a project is a three-step process as shown in the following image.



**To copy a project:**

1. Export the Project using BAMCommand: `./bamcommand -cmd export -name FilmSales -file FilmSales.zip -type project`
2. Clone the original Project, using the COPY script.

It allows you to clone the original project and copy it into a new project. Apart from the project name, all other artifact names are the same as the original project. The COPY script must be executed from the `<FMW_HOME>/soa/bam/bin` location. The syntax for the command is as follows:

```
./copyproject.sh <ORIGINAL project ZIP File> <New Project Name > <New Project Zip file path>
```

Here is an example COPY project command:

```
./copyproject.sh FilmSales.zip CLONEPROJECT /scratch/ CLONEPROJECT_1.zip
```

Command Line Argument	Corresponding Detail
1st	The original project ZIP file
2nd	New project name
3rd	New project ZIP file name and location

3. Import the project you wish to copy. See [Import](#).

## Managing Changes in a Project

Changing an entity in a project might affect other entities that are dependent on the changed entity. For example, deleting a query breaks views based on that query.

When you delete an entity, any dependent entities are listed, and you are asked to confirm the deletion.

If an entity is broken because an entity it depends on no longer exists or is also broken, its icon in the left navigation pane changes to a red circle with a white X:



You can open and edit the entity to fix it. For example, if a view is broken because the query it is based on no longer exists, you can select a different query or create a new query for the view.

Deleting a view removes it from any dashboards that reference it, and the dashboard icons do not change to a red circle with a white X.

# 5

## Working with Data Objects

This chapter discusses how to use data objects in Oracle Business Activity Monitoring (Oracle BAM) as a user in a Designer role.

This chapter includes the following sections:

- [Understanding Data Objects](#)
- [Data Object Prerequisites and Uses](#)
- [Adding a Data Object to a Project](#)
- [Removing a Data Object from a Project](#)
- [Viewing Data Object Information](#)

### Understanding Data Objects

A BAM *data object* models data for use within BAM. It contains the data to be monitored.

The data can be saved in raw format in a database or in an external table. The data can also be a stream from an ongoing activity.

Data objects are accessed directly by queries and key performance indicators (KPIs), and indirectly by parameters, views, dashboards, and alerts. Unlike these other BAM entities, a data object can be shared among different BAM projects.

This section contains the following topics:

- [Data Object Columns](#)
- [Designer Access to Data Objects](#)
- [Data Object Types](#)

### Data Object Columns

A data object is structured like a table, with columns and rows. A column can be one of these types:

- A *measure* column contains numeric data, such as sales amounts or call processing times.
- A *dimension* column contains data used to group or partition numeric data, such as product names or countries.
- An *attribute* column contains information that can be used for filtering or references, such as an ID or a timestamp.

In addition, a column of any type can be used for one of these purposes:

- A *primary* column contains IDs used to uniquely identify rows and join data objects.
- An *index* column contains the same information as another column, but sorted for efficient data retrieval.

A *hierarchy* is a set of dimension columns in a data object with a hierarchical relationship, such as country, state or province, and city.

Each column has a data type. Supported data types in BAM are VARCHAR (or String), DATETIME, INT, DECIMAL, and FLOAT.

 **Note:**

DATETIME column values use coordinated universal time (UTC). However, in dashboards and alerts, DATETIME values are converted to the user's preferred time zone. See [Setting Language, Time Zone, and Accessibility Preferences](#) for more information. You may also notice discrepancies in the value formatting of datetime values in business views. To ensure that your datetime values appear correctly in business views, change it to match the data object and business query formats. See [Formatting Business Views](#) for more information.

## Designer Access to Data Objects

An Oracle BAM user with a Designer role can:

- Add data objects to a project
- Use data objects to create queries and KPIs
- View data object details
- Edit row security, if granted update permission
- Edit data directly, if granted update permission
- Create alert actions that insert or delete rows, if granted update or delete permission

Only an Oracle BAM user with an Administrator role can create data objects, modify their structure and metadata, and grant security permissions to other roles. For more information, see [Creating and Managing Oracle BAM Data Objects](#).

## Data Object Types

There are four types of data objects:

- A *simple* data object is used as is. It can contain indexes and hierarchies, but no other column additions or modifications. It corresponds to a database view in the BAM database. Other data objects can use or refer to simple data objects.

Administrators create simple data objects for very basic data.

- A *derived* data object extends a simple data object or another derived data object. It inherits the columns and hierarchies of the parent data object and adds its own columns.

Administrators create derived data objects if additional columns are needed.

- An *external* data object references data that is persisted outside of Oracle BAM. When a connection to the external data object is configured in Oracle WebLogic Server, Oracle BAM can access it. The external data object name may be changed from the actual table name, but the column names are unchanged. Some columns may not be selected for visibility in BAM.

Administrators create external data objects to monitor external data using BAM alerts and dashboards.

External Data Objects can either be of the Fact or Dimension type. You can identify an external data object's type by the "Schema Type" label, in the **General** tab.

**Note:**

External data objects cannot be used for real time KPI, this is not supported. You can create KPI against simple data objects associated with dimension type external data objects.

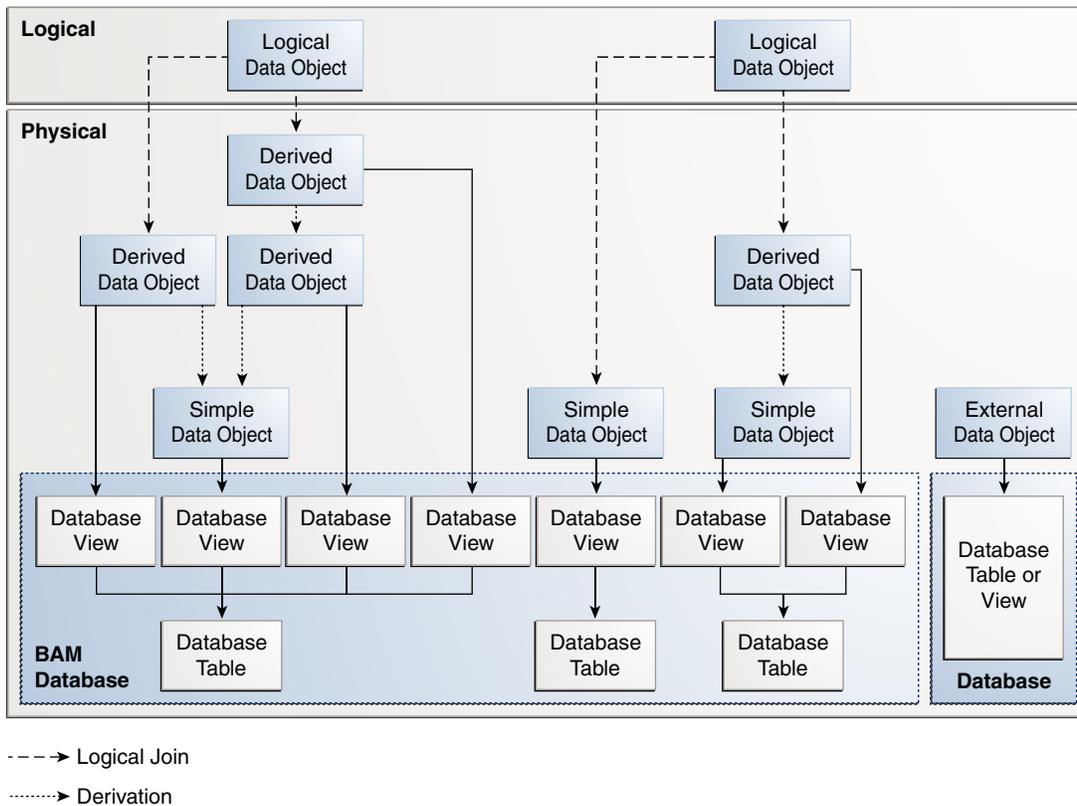
- A *logical* data object references data in other archived relation data objects. Unlike a derived data object, it is read-only and not a physical data object. However, it can do more than add columns. It can rename or hide columns, or add calculated columns. It can also join two or more simple or derived data objects. The default join type is an inner join. You can enable left outer join for logical data objects, for each join. See [Creating a Logical Data Object](#) for more information. Typically, in compliance with process star schema standards, one source data object is a fact table consisting of measures, while the others are dimension tables.

Administrators create logical data objects to modify columns or to enable queries of data from two or more data objects.

Simple, derived, and external data objects are *physical* data objects, which can be modified and are mapped to database tables. A *logical* data object is read-only, is not persisted, references data in other data objects, and is most commonly used to join other data objects.

Figure 1 shows the relationships between data object types.

**Figure 5-1 Relationships Between Data Object Types**



A simple data object can be a stream, archived stream, or archived relation. A derived data object must have the same settings as its parent. A logical data object can only be an archived relation.

- A *stream* allows only row inserts and is not persisted. It typically contains data in which only current or very recent values are important, such as stock prices.
- An *archived stream* also allows only row inserts. However, it is persisted, with non-unique rows included. It typically contains data in which current and recent values are most important, but historical values are required or useful.
- An *archived relation* allows inserts, updates, and deletes and is persisted, with unique rows. It typically contains data in which historical values are most important.

Although BAM cannot send inserts, updates, or deletes to an external data object, an external data object is considered an archived relation.

Typically continuous queries and realtime KPIs fetch data from streams, while other query types and scheduled KPIs fetch data from archived relations. See [Creating Business Queries](#) and [Creating KPIs](#) for more information.

[Table 5-1](#) summarizes the features of the different data object types in BAM.

**Table 5-1 Data Object Type Feature Comparison**

Feature	Simple, Stream	Simple, Relation	Derived, Stream	Derived, Relation	External, Relation	Logical, Relation
Physical data object	X	X	X	X	X	-
Insert rows	X	X	X	X	-	-
Update and delete rows	-	X	-	X	-	-
Add columns	X	X	X	X	X	X
Hide columns	X	X	X	X	X	X
Add calculated columns	X	X	X	X	X	X
Join other data objects	-	-	-	-	-	X

## Data Object Prerequisites and Uses

Before you can use a data object with other BAM entities such as queries and dashboards, you must create a project.

A project references one or more data objects and contains entities based on these data objects. For more information about projects, see [Planning and Creating Projects](#).

Only users in the Administrator role can create data objects. For more information about creating data objects, see [Creating and Managing Oracle BAM Data Objects](#).

After you add data objects to a project, you can use them in the following ways:

### In a query

A query fetches data from a data object and analyzes the data. See [Creating Business Queries](#) for more information.

### In a KPI

A realtime KPI fetches a single important measure from a data object and analyzes it. A scheduled KPI analyzes a single measure fetched by a query. See [Creating KPIs](#) for more information.

## Adding a Data Object to a Project

To be able to create queries, views, or dashboards to monitor data, you need to add data objects to a project first.

### To add a data object to a project:

1. Go to the **Designer** page.
2. Create or open a project.  
See [Creating a Project](#) or [Opening a Project](#).
3. Click **Data Objects**, or right-click **Data Objects** and select the **Create** menu item.  
The Data Objects dialog appears.
4. You can filter the list of data objects to more easily see the ones that you need. To filter the list:
  - a. Type a full or partial name in the text field above the **Name** column.
  - b. Type a full or partial path in the text field above the **Path** column.
  - c. Select a data object type from the drop-down list above the **Type** column: View All (the default), Simple Data Object, Logical Data Object, External Data Object, or Derived Data Object.
  - d. Click the **Query by Example** (filter) icon to hide the data objects not matching the filter criteria.
5. Check the box to the left of each data object to be added.
6. To filter the unselected data objects, check **View Only Selected**.
7. Click **Add**.

## Removing a Data Object from a Project

You cannot delete a data object as a user in the Designer role. However, you can remove a data object from a project.

### To remove a data object from a project:

1. Go to the **Designer** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Right-click the data object and select the **Delete** menu item.  
A dialog asks you to confirm the data object removal.
4. Click **OK**.  
The data object disappears from the list.

## Viewing Data Object Information

To view information about a data object, you must open it.

### To open a data object:

1. Go to the **Designer** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.

4. Select the subtab of interest. See [Table 2](#).

Most of the information is read-only, but even in the Designer role, you can edit Row Security or Data if granted permission. Administrators can only edit data objects from the Administrator page.

5. Make any desired changes and click **Save**.
6. Click the X on the data object tab to close it.

[Table 2](#) summarizes the information tabs that apply to the different data object types in BAM.

**Table 5-2 Data Object Type Tab Comparison**

Tab	Simple	Derived	External	Logical
<a href="#">The General Tab</a>	X	X	X	X
<a href="#">The Columns Tab</a>	X	X	-	X
<a href="#">The Tables Tab</a>	-	-	X	-
<a href="#">The Calculated Fields Tab</a>	X	X	X	X
<a href="#">The Indexes Tab</a>	X	X	-	-
<a href="#">The Hierarchies Tab</a>	X	X	X	X
<a href="#">The Retention Tab</a>	X	X	-	-
<a href="#">The Row Security Tab</a>	X	X	X	X
<a href="#">The Data Tab</a>	X	X	X	X
<a href="#">The Filter Tab</a>	-	-	-	X

The following sections briefly summarize the information the data object tabs display. For more detailed information, see [Creating and Managing Oracle BAM Data Objects](#).

### The General Tab

The General tab applies to all data object types, but displays different information for each type.

For all types, the following information is displayed:

- Type: Simple, Derived, External, or Logical
- Name: internal data object name

- Display Name: displayed data object name
- Description

For all except derived data objects, the following information is displayed:

- Category: this value can label and group similar data objects, such as BPM data objects.

For simple and derived data objects, the following information is displayed:

- View Name: internal name of the BAM database view
- Slow Changing Dimension (checked or unchecked)

This indicates a dimension table that changes infrequently. The data object creator can set this option to improve the performance of continuous queries.

- Archived (checked or unchecked)
- Continuous Query Type: Stream or Relation
- Number of String Columns
- Number of Long String Columns
- Number of Integer Columns
- Number of Float Columns
- Number of Decimal Columns
- Number of Date/Time Columns

The data object creator can set these options if more than the default number of columns of these types are required. In a simple data object, these cannot be changed.

For logical data objects, the following information is displayed:

- Alias: internal table alias for joined data objects

## The Columns Tab

The Columns tab applies to all types except external data objects.

For simple data objects, and for parent data objects of derived data objects, the following information is displayed for each column:

- Column Name: internal column name
- Column Type: Measure, Dimension, or Attribute
- Data Type: VARCHAR, DATETIME, INT, DECIMAL, or FLOAT
- Size: in bytes, or Precision and Scale for DECIMAL or FLOAT data types
- Nullable (checked or unchecked): whether a null value is allowed
- Hidden (checked or unchecked): whether the column is visible and selectable in queries, views, alerts, and so on
- Default Value
- Comment

For simple data objects, the following additional information is displayed for each column:

- Display Name: displayed column name
- Unique (checked or unchecked): whether unique values for each row are required

For derived data objects, the following information is displayed for each added column:

- Column Name: internal column name
- Display Name: displayed column name
- Column Type: Measure, Dimension, or Attribute
- Data Type: VARCHAR, DATETIME, INT, DECIMAL, or FLOAT
- Nullable (checked or unchecked): whether a null value is allowed
- Hidden (checked or unchecked)
- Default Value
- Comment

For logical data objects, the following information is displayed for each column:

- Logical Data Object Column Name: internal column name
- Logical Data Object Column Display Name: displayed column name
- Source Column Display Name: displayed column name in source data object
- Source Data Object: displayed source data object name
- Column Type: Measure, Dimension, or Attribute
- Data Type: VARCHAR, DATETIME, INT, DECIMAL, or FLOAT
- Size: in bytes, or Precision and Scale for DECIMAL or FLOAT data types

For logical data objects, the following additional information is displayed regarding joins:

- Primary Data Object: displayed primary data object name
- Primary Column: internal name of primary column in primary data object used in join
- Foreign Data Object: displayed foreign data object name
- Foreign Column: internal name of primary column in foreign data object used in join

## The Tables Tab

The Tables tab applies only to external data objects. It displays the following information for all columns in the data object:

- Column Name: internal column name
- Column Type: Measure, Dimension, or Attribute
- Data Type: VARCHAR, DATETIME, INT, DECIMAL, or FLOAT
- Size: in bytes, or Precision and Scale for DECIMAL or FLOAT data types
- Nullable (checked or unchecked): whether a null value is allowed
- Hidden (checked or unchecked)
- Unique (checked or unchecked): whether unique values for each row are required
- Selected Columns: checked columns are visible to BAM

## The Indexes Tab

The Indexes tab applies to simple and derived data objects. It displays the following information for index columns in the data object:

- Name: internal index column name
- Index Columns: internal indexed column name

## The Hierarchies Tab

The Hierarchies tab applies to all data object types. It displays the following information for any hierarchies defined:

- Name: hierarchy name
- Structure: display names of columns in the hierarchy, from highest to lowest

An example hierarchy is country, state or province, and city.

## The Retention Tab

The Retention tab applies to derived data objects and to simple data objects that are archived streams or archived relations. It displays the number of days of data retained in the data object. A value of zero specifies unlimited retention. For non-archived streams, no data is persisted, and any retention value is ignored. Data Retention schedules can be changed through Oracle Enterprise Manager Fusion Middleware Control. See [Setting Data Retention in a Data Object](#) for more information.

## The Row Security Tab

The Row Security tab applies to all data object types. It displays the following information:

- Name: a name of a security role
- Security Filter: a filter expression limiting the rows that can be accessed

This is useful for working with data objects that contain sensitive or confidential data not intended for all BAM viewers, designers, or even administrators.

A user must have at least read permission for row security to apply. A user must have at least write permission to be able to determine row security for other users. See [Creating and Managing Oracle BAM Data Objects](#).

## The Data Tab

The Data tab applies to all data object types. It displays the actual data stored in the data object in columns and rows. You can add and edit data rows if granted update permission. You can delete data rows if granted delete permission. You can also choose to display a specific number of data rows per page, or order the whole set on the database by editing your Preferences.

## Modifying Data Tab Preferences

This topic describes how you can modify data tab preferences.

To modify data tab preferences:

1. Login to Oracle BAM Composer.
2. Click on your account name.
3. Select **Preferences** from the drop-down list.
4. Click the **Data Object Display** tab.

5. Specify the `Rows/Page` value from the drop-down list.
6. Optionally, set a value for `Max Export All Size` to control the number of rows to be exported.
7. Optionally, click the **Order on Database** check box to order the results on the whole set on the database.

## The Filter Tab

The Filter tab applies only to logical data objects. It displays a filter expression limiting the rows that the logical data object includes. You can only access data that both this filter and row security allow you to access. See [The Row Security Tab](#).

## The Calculated Fields Tab

The Calculated Fields tab applies to all data object types. It displays the calculated columns in the data object. Operators and expression functions, combined with column names, specify new calculated columns.

Administrators can create calculated columns only in logical data objects. They can define parameters at the data object level inside calculated fields. This is different from project-level parameters that Designers can specify.

# 6

## Creating Business Queries

This chapter explains how to create queries in Oracle Business Activity Monitoring (Oracle BAM).

This chapter includes the following sections:

- [Understanding Business Queries](#)
- [Business Query Prerequisites and Uses](#)
- [Creating a Flat SQL Query](#)
- [Creating a Group SQL Query](#)
- [Creating a Tree Model Query](#)
- [Creating a Continuous Query](#)
- [Creating a Pre-defined SQL Query](#)
- [Editing a Business Query](#)
- [Renaming a Business Query](#)
- [Deleting a Business Query](#)
- [Securing a Business Query](#)
- [Filtering Data](#)
- [Using Top N Data](#)
- [Applying Aggregate Functions to Groups](#)
- [Using a Time Group or a Time Series](#)

### Understanding Business Queries

A *business query* or *query* is a request for data that matches specified conditions.

A query can fetch from a data object once, on a schedule, or continuously.

A *Flat SQL Query* is a simple table of data fields and their values. You select the data object and then select the data fields.

A *Group SQL Query* query is an analysis of one or more numeric data fields, called measures, grouped by non-numeric data fields called dimensions. For example, a group SQL query might analyze sales grouped by country. You select the data object and then select the measures and dimensions.

A *Tree Model Query*, written in SQL, is an analysis of one or more measures grouped by a hierarchy of dimensions. For example, a tree model query might analyze sales grouped by country, state or province, county or district, and city. You select the data object and then select the measures and a hierarchy.

A *Continuous Query* is an analysis of data collected in real time. For example, a continuous query might fetch the call volume for a call center every ten minutes. There are multiple continuous query templates, which perform these analyses:

- Detecting trends
- Detecting duplicates
- Monitoring counts
- Monitoring aggregate values such as averages
- Detecting missing events
- Monitoring highest aggregate values
- Monitoring key performance indicators (KPIs)

You cannot use a continuous query in any view. However, you can add the **Insert values into a data object** alert action to a continuous query. You can then use the data object that receives the values, called a *write-back data object*, as the basis for another query and view. See [Insert values into a data object](#) for more information.

Continuous queries are written in Continuous Query Language, or CQL. Unlike SQL, which operates on data in a database or cache and responds to client requests, CQL operates on data in-memory and pushes the results to clients.

In any query that fetches measure data, a measure can be any calculated field defined in the data object. For more information about calculated fields in data objects, see [Working with Data Objects](#).

[Table 1](#) summarizes the data object types and business view types with which each business query type can be used.

**Table 6-1 Query Type Compatibility with Data Object Types and View Types**

This Business Query Type	Fetches Data From	Sends Data To
Flat SQL Query	Any non-stream data object	List, Action List, and Geo Map views only
Group SQL Query	Any non-stream data object	All views <i>except</i> List, Action List, Geo Map, Treemap, or KPI Watchlist
Tree Model Query	A non-stream data object with at least one hierarchy	Treemap views only
Continuous Query: Duplicate Detection Template	A stream data object only	Alerts only
Continuous Query: all other templates	Any non-external data object	Alerts only

A KPI Watchlist view is only for KPIs; see [Creating KPIs](#) for more information.

For more information about data objects, see [Working with Data Objects](#).

For more information about views, see [Creating and Using Business Views](#). For further details about the query requirements for each view type, see [Table 8-1](#).

For more information about alerts, see [Creating Alerts](#).

## Business Query Prerequisites and Uses

Before you can create a query, you must add to your project the data object containing the data fields the query will retrieve and analyze.

For more information about projects, see [Planning and Creating Projects](#). For more information about data objects, see [Working with Data Objects](#).

After you create queries, you can use them in the following ways:

**In a KPI**

A realtime KPI can use a continuous query to monitor a risk indicator. A scheduled KPI must be based on a group SQL query with no dimensions. See [Creating KPIs](#) for more information.

**In a view**

A view displays the data that a non-continuous query retrieves. You cannot use a continuous query in any view. See [Creating and Using Business Views](#) for more information.

**In a dashboard**

A dashboard contains a group of related views. See [Creating Dashboards](#) for more information.

**With a parameter**

You can filter query data using a user-specified parameter value. See [Creating Parameters](#) for more information.

**With an alert**

A continuous query can initiate an alert action. For example, you can define an action that notifies you when the value of an Internal Costs field exceeds a certain dollar amount. This alert action is specific to the query and does not appear in the left panel navigator under Alerts. As an alternative, you can configure an alert to watch a continuous query as its triggering event. For more information, see [Creating Alerts](#).

## Creating a Flat SQL Query

This section outlines the procedure to create a Flat SQL Query.

**To create a flat SQL query:**

1. Go to the **Designer** page.
2. Click **Business Queries** in the left panel navigator, or right-click **Business Queries** and select the **Create** menu item.

The Business Queries dialog opens.

3. Type a **Name** and optionally edit the **Display Name**, select **Flat SQL Query**, and click **Create**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the query is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

A tab opens for the new query.

4. Select a **Data Object**.

Data fields in the data object are listed.

5. Check the boxes to the left of the data fields to include in the query. To include all fields in the data object, check **Select All**.

To be used in a Geo Map view, a flat SQL query must include Address and Country Code fields or Latitude and Longitude fields, and one measure. See [Creating Geo Map Business Views](#) for more information.

6. Optionally use the **Move Up** and **Move Down** icons to move the selected data field up or down in the list.  
  
The first-to-last order of the list determines the left-to-right order in which the fields are displayed in a view.
7. Optionally click the **Sort Order** icon to toggle between ascending order, descending order, and unsorted for a data field. The default is unsorted.  
  
This determines the order in which data values are displayed in the column of the resulting table. If multiple fields are sorted, the fields listed first take precedence.
8. To display only a specific number of the highest values for each numeric data field, check **Top N** at the bottom of the tab and type a number for N. The default is 3. See [Using Top N Data](#) for more information.
9. Optionally add filters. See [Filtering Data](#) for more information.
10. Optionally edit the **Display Name**. Mouse over the name in the top left corner of the tab and the name becomes editable.
11. Click **Save**.  
  
Preview data appears at the bottom of the tab.
12. If you configured an initial or default value for a parameter in a filter, the Input Values dialog appears. Optionally edit the value. Click **OK** to accept the parameter input value.
13. Optionally display a preview, XML code, or SQL code for the saved query.
  - To display a preview of how the data might look in a table view, click the **Preview** icon.  
  
If you configured an initial or default value for a parameter in a filter, the Input Values dialog appears. Optionally edit the value. Click **OK** to accept the parameter input value.
  - To display the query in XML format, click the **Modifier** icon.
  - To display the query as an SQL statement, click the **SQL** icon.  
If the preview area is not displayed, click the **Restore Pane** icon in the lower right corner of the tab.

## Creating a Group SQL Query

This section outlines the procedure to create a Group SQL Query.

### To create a group SQL query:

1. Go to the **Designer** page.
2. Click **Business Queries** in the left panel navigator, or right-click **Business Queries** and select the **Create** menu item.  
  
The Business Queries dialog opens.
3. Type a **Name** and optionally edit the **Display Name**, select **Group SQL Query**, and click **Create**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the query is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

A tab opens for the new query.

4. Select a **Data Object**.

Data fields in the data object are listed.

5. Check the boxes to the left of the Measures (Y-axis) data fields to include in the query.

6. Optionally use the **Move Up** and **Move Down** icons to move the selected data field up or down in the Measures list.

The first-to-last order of the list determines the left-to-right order in which the fields are displayed in a view.

7. For each selected data field, display the drop-down list and check one or more aggregation functions. To specify all aggregation functions, check **All**. See [Applying Aggregate Functions to Groups](#) for more information.

8. To display only a specific number of the highest values for each numeric data field, check **Top N** at the bottom of the tab and type a number for N. The default is 3. See [Using Top N Data](#) for more information.

9. Check the boxes to the left of the Dimensions (X-axis) data fields to include in the query. These are the categories by which the Measure data field values are grouped for aggregation.

If you are using the query in a Scheduled KPI or a Gauge type view, do not select any dimensions. For more information, see [Creating a Scheduled KPI](#) or [Creating Gauge Business Views](#).

10. If you check a datetime field as a dimension, the Time Grouping dialog appears. See [Using a Time Group or a Time Series](#) for information about how to define a time series or time group.

 **Note:**

If a Group SQL query uses a datetime dimension with Quarter, Week, DayOfYear, or DayOfMonth as time units in graph views, drill to details and hierarchical drilling does not work.

11. Optionally use the **Move Up** and **Move Down** icons to move the selected data field up or down in the Dimensions list.

The first-to-last order of the list determines the left-to-right order in which the fields are displayed in a view.

12. Optionally click the **Sort Order** icon to toggle between ascending order, descending order, and unsorted for a Dimension data field. The default is unsorted.

This determines the order in which data values are displayed in views. If multiple fields are sorted, the fields listed first take precedence.

13. Optionally select a **Legend**.

If specified, this is the most important Dimension data field used for grouping. The legend matches values in this field to the colors used in a view. For example, a bar chart view with a Department legend might show green for Sales, blue for Development, and red for Production.

14. Optionally add filters. See [Filtering Data](#) for more information.
15. Optionally edit the **Display Name**. Mouse over the name in the top left corner of the tab and the name becomes editable.
16. Click **Save**.  
Preview data appears at the bottom of the tab.
17. If you configured an initial or default value for a parameter in a filter, the Input Values dialog appears. Optionally edit the value. Click **OK** to accept the parameter input value.
18. Optionally display a preview, XML code, or SQL code for the saved query.
  - To display a preview of how the data might look in a table view, click the **Preview** icon.
  - To display a preview of how the data might look in a bar graph view, click the **Bar Graph** icon.
  - To display a preview of how the data might look in a line graph view, click the **Line Graph** icon.
  - To display a preview of how the data might look in a pivot table view, click the **Pivot Table** icon.
  - To display the query in XML format, click the **Modifier** icon.
  - To display the query as an SQL statement, click the **SQL** icon.

If the preview area is not displayed, click the **Restore Pane** arrow icon in the lower right corner of the tab.

If you configured an initial or default value for a parameter in a filter, the Input Values dialog appears when you select any of the preview icons. Optionally edit the value. Click **OK** to accept the parameter input value.

## Creating a Tree Model Query

This section outlines the procedure to create a Tree Model Query.

A tree model query is applicable only to a data object with at least one hierarchy of dimension data fields defined. For example, a hierarchy might consist of the country, state or province, and city. See [Working with Data Objects](#) for more information.

You can use a tree model query only in a Treemap view, which is designed to display hierarchical data effectively. See [Creating Treemap Business Views](#) for more information.

### To create a tree model query:

1. Go to the **Designer** page.
2. Click **Business Queries** in the left panel navigator, or right-click **Business Queries** and select the **Create** menu item.

The Business Queries dialog opens.

3. Type a **Name** and optionally edit the **Display Name**, select **Tree Model Query**, and click **Create**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the query is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

A tab opens for the new query.

4. Select a **Data Object**.

Data fields in the data object are listed.

5. Check the boxes to the left of the Measures (Y-axis) data fields to include in the query.

6. Optionally use the **Move Up** and **Move Down** icons to move the selected data field up or down in the Measures list.

7. For each selected data field, display the drop-down list and check one or more aggregation functions. To specify all aggregation functions, check **All**. See [Applying Aggregate Functions to Groups](#) for more information.

8. To display only a specific number of the highest values for each numeric data field, check **Top N** at the bottom of the tab and type a number for N. The default is 3. See [Using Top N Data](#) for more information.

9. Select a **Hierarchy** from the drop-down list. The data fields in the hierarchy are the categories by which the Measure data field values are grouped for aggregation. Datetime hierarchies are not supported.

10. Optionally add filters. See [Filtering Data](#) for more information.

11. Optionally edit the **Display Name**. Mouse over the name in the top left corner of the tab and the name becomes editable.

12. Click **Save**.

Preview data appears at the bottom of the tab.

13. If you configured an initial or default value for a parameter in a filter, the Input Values dialog appears. Optionally edit the value. Click **OK** to accept the parameter input value.

14. Optionally display a preview, XML code, or SQL code for the saved query.

- To display a preview of how the data might look in a table view, click the **Preview** icon.
- To display the query in XML format, click the **Modifier** icon.
- To display the query as an SQL statement, click the **SQL** icon.

If the preview area is not displayed, click the **Restore Pane** arrow icon in the lower right corner of the tab.

If you configured an initial or default value for a parameter in a filter, the Input Values dialog appears when you select any of the preview icons. Optionally edit the value. Click **OK** to accept the parameter input value.

## Creating a Continuous Query

This section outlines the procedure to create a Continuous Query.

Continuous queries use templates. Each template has its own settings. Therefore, the steps to create a continuous query are different for each template.

**To create a continuous query:**

1. Go to the **Designer** page.
2. Click **Business Queries** in the left panel navigator, or right-click **Business Queries** and select the **Create** menu item.

The Business Queries dialog opens.

3. Type a **Name** and optionally edit the **Display Name**, select **Continuous Query**, and click **Create**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the query is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

A tab opens for the new query.

4. Optionally check **Activate Continuous Queries: Name** to activate the query. The query is activated after it is saved if this box is checked.

You can uncheck this box later to deactivate the query without deleting it.

5. Select a **Template**. See one of the following sections for detailed steps on using the template you select.

- [Using the Trending Detection Template](#)
- [Using the Duplicate Detection Template](#)
- [Using the Monitor Count Template](#)
- [Using the Moving Aggregation Template](#)
- [Using the Missing Event Template](#)
- [Using the Top N Template](#)
- [Using the KPI Alert Template](#)

 **Note:**

All templates now support calculated fields when you select a Measure Field or use the Group By option. However, an embedded calculated field, that is — one calculated field within another, is not supported.

6. Optionally edit the **Description**.

A default description is provided based on the template you selected. You can edit the description to include the specific options you selected in the template.

7. Click **Save**.

8. Optionally display a preview of the CQL statement for the saved query. For continuous queries, the **Preview** icon is a magnifying glass in the upper right corner of the tab.

9. The **<add action>** option appears after you save the query. You can optionally add an alert action that fires when the query detects out-of-range conditions. For more information about alert actions, see [Alert Actions](#).

As an alternative to creating an alert action in the query, you can configure an alert to watch a continuous query as its triggering event. See [Select a CQL Query](#) for more information.

10. Click **Save** again if you configured an alert action.

## Using the Trending Detection Template

This template detects when a numeric field shows a specified percentage change in value for a specified number of consecutive intervals. For example, such a query might detect when the call volume for a call center increases by 10% for two consecutive 30-minute intervals.

### To create a continuous query using the Trending Detection Template:

#### 1. Specify the following **Measure** settings:

- **Data Object** — The data object containing the measure the query will analyze.
- **Filter** — Click the funnel icon to optionally add filters. This icon is clickable after you select a data object. See [Filtering Data](#) for more information.
- **Measure Field** — The numeric field the query measures.
- **Aggregation Function** — Select None, Maximum, Minimum, Count, Sum, or Average. See [Applying Aggregate Functions to Groups](#) for more information.
- **Group By** — Select the dimensions by which the Measure Field values are grouped for aggregation. You can select **All**. This option is not displayed if the Aggregation Function is set to None.

Time Grouping is not supported for continuous queries.

- **Partition By** — Select the ordered fields by which the Measure Field values are partitioned. You can select **All**.

Partitioning data values groups them without performing an aggregation function on them. Therefore, this option is displayed only if the Aggregation Function is set to None.

- **Use rolling window** — Optionally check this box to define a period over which and a frequency at which the aggregation or partition is determined for incoming data. For example, you could update the number of calls completed in the last hour every ten minutes.

For archived relation data objects, this setting is checked automatically and cannot be unchecked.

- **Range Length** — The period over which the aggregation or partition is determined, from an amount of time in the past to the present. Examples of ranges are the last 15 or 30 minutes.

If you specify more than one hour, a message appears warning of possible performance degradation when you select **Save** or **Preview**.

- **Update Interval** — The aggregation or partition is determined every time new incoming data is received. The update interval determines how often the result within the rolling window is updated.
- **Units** — Select Day, Hour, Minute, or Second for the Range Length and Update Interval.
- **Based on** — A date or time column upon which the rolling window calculation is based. For example, choosing Call Start Time enables the system to determine whether a given call is within the rolling window period.

This setting is displayed only if the selected data object is a relation. It is not supported for stream data objects.

#### 2. Specify the following **Trending** settings:

- **Change: Operation** — Select greater than, greater than or equal to, less than, or less than or equal to.
  - **Change: Percent** — Type a value between 1 and 100 inclusive for the percent of change to the measure.
  - **Consecutive Interval** — Type an integer value representing the number of Update Interval occurrences needed to trigger an alert.
3. Specify **Output** options to be included in the alert notification. The data field value options presented depend on the Measure Field. You can select **All**.

## Using the Duplicate Detection Template

The Duplicate Detection Template can only query stream type data objects.

This template detects when a data field has duplicate values within a specified period of time. For example, such a query might detect when the same order is placed twice within a day.

### To create a continuous query using the Duplicate Detection Template:

1. Specify the following **Measure** settings:
  - **Data Object** — The data object containing the measure the query will analyze.
  - **Measure Field** — The field the query analyzes for duplicates.
  - **Use rolling window** — Use of a rolling window is mandatory for this template. This defines a period over which and a frequency at which the aggregation or partition is determined for incoming data. For example, you could update the number of calls completed in the last hour every ten minutes.
  - **Range Length** — The period over which data values are analyzed for duplicates, from an amount of time in the past to the present. Examples of ranges are the last 15 or 30 minutes.  
  
If you specify more than one hour, a message appears warning of possible performance degradation when you select **Save** or **Preview**.
  - **Units** — Select Day, Hour, Minute, or Second for the Range Length.
2. Specify **Output** options to be included in the alert notification. The data field value options presented depend on the Measure Field. You can select **All**.

#### Note:

The Duplicate Detection Template does not support calculated fields. The Measure Field and Output options will not contain any calculated fields from the data object.

## Using the Monitor Count Template

This template monitors the count of one or more data fields. For example, such a query might monitor the number of currently active calls in a call center.

### To create a continuous query using the Monitor Count Template:

1. Specify the following **Measure** settings:
  - **Data Object** — The data object containing the measure the query will analyze.

- **Filter** — Click the funnel icon to optionally add filters. This icon is clickable after you select a data object. See [Filtering Data](#) for more information.
- **Measure Field** — The field the query counts. You can select **All** to specify all fields in the data object that can be counted.
- **Group By** — Select the dimensions by which the Measure Field values are grouped for counting. You can select **All**.

If a Measure Field cannot be grouped by the selected Group By field, values in that Measure Field are concatenated in the Output.

- **Count: Operation** — Select greater than, greater than or equal to, less than, less than or equal to, or equal to.
- **Count: Threshold** — Type an integer value for the threshold to be compared to the count.
- **Use rolling window** — Optionally check this box to define a period over which and a frequency at which the aggregation or partition is determined for incoming data. For example, you could update the number of calls completed in the last hour every ten minutes.

For archived relation data objects, this setting is checked automatically and cannot be unchecked.

- **Range Length** — The period over which the aggregation or partition is determined, from an amount of time in the past to the present. Examples of ranges are the last 15 or 30 minutes.  
If you specify more than one hour, a message appears warning of possible performance degradation when you select **Save** or **Preview**.
- **Update Interval** — The count is determined every time new incoming data is received. The update interval determines how often the result within the rolling window is updated.
- **Units** — Select Day, Hour, Minute, or Second for the Range Length and Update Interval.
- **Based on** — A date or time column upon which the rolling window calculation is based. For example, choosing Call Start Time enables the system to determine whether a given call is within the rolling window period.

This setting is displayed only if the selected data object is a relation. It is not supported for stream data objects.

2. Specify **Output** options to be included in the alert notification. The data field value options presented depend on the Measure Field. You can select **All**.

## Using the Moving Aggregation Template

This template monitors an aggregation (average, sum, and so on) of one or more data fields. For example, such a query might monitor the average wait time for calls in a call center.

### To create a continuous query using the Moving Aggregation Template:

1. Specify the following **Measure** settings:
  - **Data Object** — The data object containing the measure the query will analyze.
  - **Filter** — Click the funnel icon to optionally add filters. This icon is clickable after you select a data object. See [Filtering Data](#) for more information.
  - **Measure Field** — The numeric fields the query measures.

- **Aggregation Function** — Select All, Maximum, Minimum, Count, Sum, or Average for each selected Measure Field. See [Applying Aggregate Functions to Groups](#) for more information.
- **Group By** — Select the dimensions by which the Measure Field values are grouped for aggregation. You can select **All**.

Time Grouping is not supported for continuous queries.

- **Use rolling window** — Optionally check this box to define a period over which and a frequency at which the aggregation or partition is determined for incoming data. For example, you could update the number of calls completed in the last hour every ten minutes.

For archived relation data objects, this setting is checked automatically and cannot be unchecked.

- **Range Length** — The period over which the aggregation or partition is determined, from an amount of time in the past to the present. Examples of ranges are the last 15 or 30 minutes.

If you specify more than one hour, a message appears warning of possible performance degradation when you select **Save** or **Preview**.

- **Update Interval** — The aggregation is calculated every time new incoming data is received. The update interval determines how often the result within the rolling window is updated.
- **Units** — Select Day, Hour, Minute, or Second for the Range Length and Update Interval.
- **Based on** — A date or time column upon which the rolling window calculation is based. For example, choosing Call Start Time enables the system to determine whether a given call is within the rolling window period.

This setting is displayed only if the selected data object is a relation. It is not supported for stream data objects.

2. Specify **Output** options to be included in the alert notification. The data field value options presented depend on the Measure Field. You can select **All**.

## Using the Missing Event Template

This template detects when an expected event does not occur. For example, such a query might detect when order approval is missing between order placement and order shipment.

### To create a continuous query using the Missing Event Template:

1. Specify the following **Measure** settings:
  - **Data Object** — The data object containing the measure the query will analyze.
  - **Filter** — Click the funnel icon to optionally add filters. This icon is clickable after you select a data object. See [Filtering Data](#) for more information.
  - **Measure Field** — The ordered fields the query measures. You can select **All**.
  - **Partition By** — Select the ordered fields by which the Measure Field values are partitioned. You can select **All**.  
  
Partitioning data values groups them without performing an aggregation function on them.
  - **Use rolling window** — Optionally check this box to define a period over which and a frequency at which the aggregation or partition is determined for incoming data. For

example, you could update the number of calls completed in the last hour every ten minutes.

For archived relation data objects, this setting is checked automatically and cannot be unchecked.

- **Range Length** — The period over which the aggregation or partition is determined, from an amount of time in the past to the present. Examples of ranges are the last 15 or 30 minutes.

If you specify more than one hour, a message appears warning of possible performance degradation when you select **Save** or **Preview**.

- **Update Interval** — The partition is determined every time new incoming data is received. The update interval determines how often the result within the rolling window is updated.
- **Units** — Select Day, Hour, Minute, or Second for the Range Length and Update Interval.
- **Based on** — A date or time column upon which the rolling window calculation is based. For example, choosing Call Start Time enables the system to determine whether a given call is within the rolling window period.

This setting is displayed only if the selected data object is a relation. It is not supported for stream data objects.

2. Specify the following **Event** settings:

- **Pattern** — Select one of the following patterns:
  - **Event A is not followed directly by Event B** — Event B does not occur immediately after Event A.
  - **Event C is missing between Event A and Event B** — Event C does not occur between Event A and Event B.
  - **Event A is not followed by Event B in specific duration** — Event B does not occur within a specified period of time after Event A.
- **Event A, Event B, and Event C** — Select the following settings for each event.
  - **Field** — Select the data field that represents the event.
  - **Operation** — Select greater than, less than, or equal to.
  - **Value** — Type a value to which the event field is compared.

Settings for Event C appear only if **Event C is missing between Event A and Event B** is selected.

- **Duration** — Type an integer and select Day, Hour, Minute, or Second.

This setting appears only if **Event A is not followed by Event B in specific duration** is selected.

3. Specify **Output** options to be included in the alert notification. The data field value options presented depend on the Measure and Partition By fields. You can select **All**.

## Using the Top N Template

This template monitors the top values of a grouped aggregation (average, sum, and so on) of a numeric field. For example, such a query might monitor which three countries have the top sales totals.

**To create a continuous query using the Top N Template:**

1. Specify the following **Measure** settings:

- **Data Object** — The data object containing the measure the query will analyze.
- **Filter** — Click the funnel icon to optionally add filters. This icon is clickable after you select a data object. See [Filtering Data](#) for more information.
- **Measure Field** — The numeric field the query measures.
- **Aggregation Function** — Select Maximum, Minimum, Count, Sum, or Average. See [Applying Aggregate Functions to Groups](#) for more information.
- **Group By** — Select the dimensions by which the Measure Field values are grouped for aggregation. You can select **All**.

Time Grouping is not supported for continuous queries.

- **Top N** — Specify the number of highest values to be fetched for the Measure Field aggregation groups. For example, you could specify the countries with the top three sales figures.
- **Use rolling window** — Optionally check this box to define a period over which and a frequency at which the aggregation or partition is determined for incoming data. For example, you could update the number of calls completed in the last hour every ten minutes.

For archived relation data objects, this setting is checked automatically and cannot be unchecked.

- **Range Length** — The period over which the aggregation or partition is determined, from an amount of time in the past to the present. Examples of ranges are the last 15 or 30 minutes.

If you specify more than one hour, a message appears warning of possible performance degradation when you select **Save** or **Preview**.

- **Update Interval** — The aggregation is calculated every time new incoming data is received. The update interval determines how often the result within the rolling window is updated.
- **Units** — Select Day, Hour, Minute, or Second for the Range Length and Update Interval.
- **Based on** — A date or time column upon which the rolling window calculation is based. For example, choosing Call Start Time enables the system to determine whether a given call is within the rolling window period.

This setting is displayed only if the selected data object is a relation. It is not supported for stream data objects.

2. Specify **Output** options to be included in the alert notification. The data field value options presented depend on the Measure Field. You can select **All**.

## Using the KPI Alert Template

A continuous query that uses the KPI Alert Template is similar to a realtime KPI. However, you can use a realtime KPI in a KPI watchlist view. You cannot use a continuous query in any view. See [Creating KPIs](#) for more information about KPIs.

This template detects when a grouped aggregation (average, sum, and so on) of a numeric field shows a specified change from a defined norm. For example, such a query might detect when the call volume for a call center is more than one standard deviation from the average of the last 24 hours.

**To create a continuous query using the KPI Alert Template:****1.** Specify the following **Measure** settings:

- **Data Object** — The data object containing the measure the query will analyze.
- **Filter** — Click the funnel icon to optionally add filters. This icon is clickable after you select a data object. See [Filtering Data](#) for more information.
- **Measure Field** — The numeric field the query measures.
- **Aggregation Function** — Select Moving Average, Moving Sum, Moving Min, or Moving Max. See [Applying Aggregate Functions to Groups](#) for more information.
- **Group By** — Select the dimensions by which the Measure Field values are grouped for aggregation. You can select **All**.

Time Grouping is not supported for continuous queries.

- **Use rolling window** — Optionally check this box to define a period over which and a frequency at which the aggregation or partition is determined for incoming data. For example, you could update the number of calls completed in the last hour every ten minutes.

For archived relation data objects, this setting is checked automatically and cannot be unchecked.

- **Range Length** — The period over which the aggregation or partition is determined, from an amount of time in the past to the present. Examples of ranges are the last 15 or 30 minutes.

If you specify more than one hour, a message appears warning of possible performance degradation when you select **Save** or **Preview**.

- **Update Interval** — The aggregation is recalculated every time new incoming data is received. The update interval determines how often the result within the rolling window is updated.
- **Units** — Select Day, Hour, Minute, or Second for the Range Length and Update Interval.
- **Based on** — A date or time column upon which the rolling window calculation is based. For example, choosing Call Start Time enables the system to determine whether a given call is within the rolling window period.

This setting is displayed only if the selected data object is a relation. It is not supported for stream data objects.

**2.** Specify the following **Threshold** settings:

- **Criteria** — Specify one of the following threshold criteria:
  - **Measure is greater than a (Norm + Deviation)** — Measure values are higher than a threshold.
  - **Measure is less than a (Norm – Deviation)** — Measure values are lower than a threshold.
  - **Measure is not within a (Norm +/- Deviation)** — Measure values are either higher or lower than thresholds that define a normal range.
- **Norm Definition: Constant** — Type an integer value in the **Simple Value** text box to define the norm.
- **Norm Definition: Historical** — Specify the following settings to define a historical norm:

- **Aggregation Function** — Select Average, Sum, Min, or Max.
  - **Window Size** — The period over which the aggregation is calculated, from an amount of time in the past to the present. Examples of window sizes are the last 30 minutes or the last 24 hours.
  - **Based on** — A date or time column upon which the rolling window calculation is based. For example, choosing Call Start Time enables the system to determine whether a given call is within the rolling window period.  
  
This setting is displayed only if the selected data object is a relation. It is not supported for stream data objects.
  - **Refresh Time** — The first or next time the historical norm will be calculated.
  - **Refresh Frequency** — The frequency at which the norm is recalculated.
  - **Units** — Select Month, Week, Day, Hour, Minute, or Second for the Window Size and Refresh Frequency.
  - **Deviation Definition** — Specify the following settings to define a deviation from the norm:
    - **Scalar Value** — Type an integer value to define the deviation.
    - **Percent of Norm** — Type a percentage of the norm.
    - **Calculation** — Type a multiplier in the **Times** text box and select **Standard Deviation** or **Variance**.
3. Specify the following **Alert Event: Output** options to be included in the alert notification:
- **All** — The measure, norm, allowed deviation, and actual deviation values.
  - **Measure** — The measure value.
  - **Norm** — The norm value.
  - **Allowed Deviation** — The lower or upper threshold value, or both.
  - **Actual Deviation** — The difference between the measure value and the norm.

## Creating a Pre-defined SQL Query

Oracle BAM allows you to write your own SQL which can then be used for flat or group queries against that SQL. The arbitrary SQL that you enter is treated as a sub-query by the system. The returned result set is treated as set of columns like in a Data Object.

To create a pre-defined SQL:

1. Go to the **Designer** page.
2. Click **Business Queries** in the left panel navigator, or right-click **Business Queries** and select the **Create** menu item.

The Business Queries dialog opens.

3. Type a **Name** and optionally edit the **Display Name**, select **Flat/Group SQL Query**, and click **Create**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the query is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time. A tab opens for the new query.

4. Click the 'Yes' button next to the 'Create query based on defined SQL'.

A Data Source drop-down list appears.

5. Choose a data source from the drop-down list. The data source you choose must have all data server information – including a username and password - already pre-configured on the Weblogic Server Administration Console.
6. Enter the SQL statement you need. You can modify the pre-defined SQL statement as needed. Columns are generated according to your SQL statement and when the SQL changes, they're updated accordingly.

**Note:** Hierarchical queries are not supported within pre-defined SQL queries.

**Note:** Pre-defined SQL queries do not support duplicate column names. If your query has two or more identical column names, they must be renamed to ensure that the resulting SQL query columns are unique.

7. To view editable fields, measures, dimensions, and filters, click outside the SQL statement box, or save or preview the SQL statement.
8. If you have based your SQL on a flat query:

- a. Select the fields and filters you need for you want to apply. To include all fields in the data object, check Select All.
- b. To be used in a Geo Map view, a flat SQL query must include Address and Country Code fields or Latitude and Longitude fields, and one measure. See Section 7.3.5, "Creating Geo Map Business Views" for more information.
- c. Optionally use the Move Up and Move Down icons to move the selected data field up or down in the list. The first-to-last order of the list determines the left-to-right order in which the fields are displayed in a view.
- d. Optionally click the Sort Order icon to toggle between ascending order, descending order, and unsorted for a data field. The default is unsorted. This determines the order in which data values are displayed in the column of the resulting table. If multiple fields are sorted, the fields listed first take precedence.

9. If you have created your SQL based on a group query:

- a. Check the boxes to the left of the Measures (Y-axis) data fields to include in the query. These are the categories by which the Measure data field values are grouped for aggregation.
- b. Optionally use the Move Up and Move Down icons to move the selected data field up or down in the Measures list. The first-to-last order of the list determines the left-to-right order in which the fields are displayed in a view.
- c. For each selected measure, display the drop-down list and check one or more aggregation functions. To specify all aggregation functions, check All. See Section 5.13, "Applying Aggregate Functions to Groups" for more information.

Note that aggregations (SQL GROUP BY) are applied on top of the pre-defined SQL query which may already have aggregations performed. You can select the SUM aggregation to effectively pass through values from opaque query that already does aggregation.

- d. Check the boxes to the left of the Dimensions (X-axis) data fields to include in the query. These are the categories by which the Measure data field values are grouped for aggregation.
- e. If you are using the query in a Scheduled KPI or a Gauge type view, do not select any dimensions. For more information, see Section 6.4, "Creating a Scheduled KPI" or Section 7.3.4, "Creating Gauge Business Views."

- f. If you check a datetime field as a dimension, the Time Grouping dialog appears. See Section 5.14, "Using a Time Group or a Time Series" for information about how to define a time series or time group.
  - g. Optionally use the Move Up and Move Down icons to move the selected data field up or down in the Dimensions list. The first-to-last order of the list determines the left-to-right order in which the fields are displayed in a view.
  - h. The first-to-last order of the list determines the left-to-right order in which the fields are displayed in a view. This determines the order in which data values are displayed in views. If multiple fields are sorted, the fields listed first take precedence.
10. Optionally add filters. See Section 5.11, "Filtering Data" for more information.
  11. Click Save.  
Preview data appears at the bottom of the tab.
  12. At the bottom of the page, you can see a "Top N" checkbox. If you want to get the highest values for each field, you can enable this check box.
  13. Select a number for N. The default value is 0.
  14. Optionally select a Legend. If specified, this is the most important Dimension data field used for grouping. The legend matches values in this field to the colors used in a view. For example, a bar chart view with a Department legend might show green for Sales, blue for Development, and red for Production.
  15. Optionally edit the Display Name. Mouse over the name in the top left corner of the tab and the name becomes editable.
  16. Click Save.  
Preview data appears at the bottom of the tab.
  17. Optionally display a preview, bar graph, line graph, pivot table, XML code, or SQL code for the saved query.
    - a. To display a preview of how the data might look in a table view, click the Preview icon.
    - b. To display a preview of how the data might look in a bar graph view, click the Bar Graph icon.
    - c. To display a preview of how the data might look in a line graph view, click the Line Graph icon.
    - d. To display a preview of how the data might look in a pivot table view, click the Pivot Table icon.
    - e. To display the query in XML format, click the Modifier icon.
    - f. To display the query as an SQL statement, click the SQL icon.If the preview area is not displayed, click the Restore Pane arrow icon in the lower right corner of the tab.
  18. If you configured an initial or default value for a parameter in a filter, the Input Values dialog appears when you select any of the preview icons. Optionally edit the value. Click OK to accept the parameter input value.

## Editing a Business Query

When you edit a query, the changes propagate to all views that include the query and all dashboards that include these views.

Use the following procedure to open, edit, and save a query.

**To edit a query:**

1. In the left navigation pane, click the arrow to the left of **Business Queries**.  
All saved queries in the current project are displayed in a list.
2. Click the query name, or right-click the query name and select the **Edit** menu item.
3. Make the desired changes.
4. Click **Save**.

 **Note:**

Before editing a query, close any business views, editors, or dashboards that are using it. This ensures that there is no data transmission loss to any dependent artifacts of the business query in use.

## Renaming a Business Query

This section outlines the procedure to rename a Business Query.

Use the following procedure to change the **Display Name** of a query.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

**To rename a query:**

1. In the left navigation pane, click the arrow to the left of **Business Queries**.  
All saved queries in the current project are displayed in a list.
2. To rename the query, right-click the query name and select the **Rename** menu item.  
The name becomes text in an editable field.
3. Type the new name and press **Enter**.  
The new name appears in the list.

When you edit the query, the new **Display Name** is displayed on the tab. However, the internal **Name** remains unchanged. When you edit the query, this name is displayed on the left in the header.

## Deleting a Business Query

This section outlines the procedure to delete a Business Query. The query is removed from any views or dashboards that reference it.

Use the following procedure to delete a query.

**To delete a query:**

1. In the left navigation pane, click the arrow to the left of **Business Queries**.  
All saved queries in the current project are displayed in a list.
2. Click the query icon and click the **Delete** icon, or right-click the query name and select the **Delete** menu item.

A dialog asks you to confirm the query deletion.

3. Click **OK**.

The query disappears from the list.

## Securing a Business Query

A query inherits security settings from the project in which it is created.

For more information about projects, see [Planning and Creating Projects](#).

### To change security settings for a query:

1. In the left navigation pane, click the arrow to the left of **Business Queries**.  
All saved queries in the current project are displayed in a list.
2. Right-click the query and select **Security Settings** from the pop-up menu.  
The security tab for the query opens.
3. To add a role or group to whom you can explicitly grant or deny permissions, follow these steps:
  - a. Click the **Add** icon in the Grant Permissions or Deny Permissions table.  
The Add Application Roles, Groups, and Users dialog opens.  
See [Managing Oracle BAM Users](#) for information about how to add users to roles and groups.
  - b. Type a **Name** for the role or group you are adding.
  - c. Select from the drop-down **List**: Application Role or Group.
  - d. Click **Search** to populate the Available Members list.
  - e. To add a member to the Selected Members list, select the member and click the single right arrow.
  - f. To add all members to the Selected Members list, select the member and click the double right arrow.
  - g. To remove members from the Selected Members list, use the single and double left arrows.
  - h. When the Selected Members list is final, click **OK**.  
The Add Application Roles, Groups, and Users dialog closes, and the **Name** you specified appears in the table.
4. To remove a role or group, select the table row and click the **Remove** icon.
5. To grant permissions, select **Read**, **Write**, **Remove**, or **Security** for the users, roles, and groups listed in the Grant Permissions table.
6. To deny permissions, select **Read**, **Write**, **Remove**, or **Security** for the users, roles, and groups listed in the Deny Permissions table.
7. Click **Save**.

## Filtering Data

You can add data filters to most types of queries. You can also filter data in some alert operations.

A simple example filter is "Sales greater than 12000." When this filter is applied, only rows with values larger than 12000 in the Sales data field are fetched. See [When a data field in a data object meets specified conditions](#) and [Delete rows from a data object](#) for more information.

By adding logical sub-branches, you can create a hierarchy of filters. (In previous releases, branches were called headers.)

To filter data completely, fetching no rows from the data object, check **Filter All Data**.

### To add a row data filter:

1. Click the **New Filter** icon.  
The Add/Edit Filter dialog appears.
2. Select the data field to be filtered from the drop-down list on the left.
3. Select the operation from the drop-down list in the center. The default operation is **is equal to**. See [Filter Comparison Operations](#) for more information.
4. Select the item with which to compare the value in the filtered data field:
  - *Type a Value* — A value you specify.

 **Note:**

The auto-suggest feature doesn't work for calculated fields in filters. If you type a valid value and refresh, you can still see the results.

- **Select a Column** — The value in another data field.
  - **Select a Parameter** — User input through a parameter. See [Creating Parameters](#) for more information.
  - **Select a Token** — A token. Select **User Application Roles** to specify the security roles with access to the query. Select **Current User** to specify the name of the currently logged-in user. See [Securing a Business Query](#) for information about granting access to the query.
5. Click **OK** to save the filter.
  6. Optionally edit the root branch, which is **All are true** by default. Select the branch and click the **Edit** icon. Then choose one of the following branch types and click **OK**:
    - **All are true** — A row is fetched if all filters under the branch are true.
    - **Any is true** — A row is fetched if at least one filter under the branch is true.
    - **Not all are true** — A row is fetched if at least one filter under the branch is false.
    - **None is true** — A row is fetched if all filters under the branch are false.
  7. Optionally select the root branch and click the **New Branch** icon to add a sub-branch. Then choose one of the branch types listed in the previous step and click **OK**.

- Optionally add any additional filters or sub-branches you need. A branch does nothing unless at least one filter is under it. To delete a branch or filter, select it and click the **Delete** icon.
- Click **Save**.

## Adding a Group Data Filter

You can add group data filters to business queries, for example, sum of sales or average of sales, or count of region. You can apply the group filters to queries at run-time as well.

To add a group filter:

- Select the **Group Filters** tab.
- Click the **New Filter** icon. The Add/Edit Filter dialog appears.
- Select the data field to be filtered from the drop-down list on the left.
- Select the operation from the drop-down list in the center. The default operation is **is equal to**.
- Select the item with which to compare the value in the filtered data field.
- Type a Value — A value you specify. Alternatively, select a Parameter — User input through a parameter. See [Creating Parameters](#) for more information.
- Click **OK** to save the filter.

## Filter Comparison Operations

You can use the following operations for filter comparisons:

**is equal to** returns the exact value. For numeric data types, nulls are not fetched for filters using a typed-in value of zero. See [Exceptions for Datetime Filter Comparisons](#) for timestamp exceptions.

For numeric data types, nulls are not fetched for filters using an **is equal to** or **is not equal to** operation and a typed-in value of zero.

**is less than** returns values less than specified value.

**is less than or equal to** returns values less than or equal to specified value.

**is greater than** returns values greater than specified value.

**is greater than or equal to** returns values greater than or equal to specified value.

**is not equal to** returns all values except specified value. For numeric data types, nulls are not fetched for filters using a typed-in value of zero.

**is like** returns values that match a string pattern. Include an underscore ( `_` ) as a wildcard for a single character in a string, and a percent symbol ( `%` ) as a wildcard for one character or more. Wildcard characters can be combined, for example, `%mm_00` would return all columns (35mm 200, 35mm 400, 35mm 800). Do not enter any spaces in the expression because spaces are treated as characters in the data match.

**is not like** returns values that do not match a string pattern. You can use the wildcard characters permitted in **is like** comparison expression.

**is null** returns values where the column is null.

**is not null** returns values where the column is not null.

**is in list** returns values included in a list. See [Exceptions for Datetime Filter Comparisons](#) for timestamp exceptions. When you select this option, the text box in which you can type a value expands to allow multiple values. Add as many values as needed, each on a separate line.

**is not in list** returns values not included in the list.

## Exceptions for Datetime Filter Comparisons

You can use datetime data types with the **is like** or **is not like** operation, but you must specify the full datetime format, including the time zone. For example:

```
02/20/2014 08:00:00 AM
```

The datetime value entered is assumed to be in the UTC time zone.

Note that **is equal to** and **is in list** filters do not work for timestamp values because Oracle databases use microsecond precision, while Oracle BAM uses second precision.

You can work around this limitation by creating two filters: one using **is greater than or equal to** and the desired value, and another using **less than** and the value plus one second.

For example, to create a filter for **is equal to** 10:05:30 11:14:35 AM:

1. Make sure the type of the branch under which both filters are created is **All are true**.
2. In the first filter, choose the operation **is greater than or equal to** and type the value 10:05:30 11:14:35.
3. In the second filter, choose the operation **is less than** and type the value 10:05:30 11:14:36.

See [Column Data Types](#) for more information about the BAM datetime format.

## Using Top N Data

You can choose to display only the top values of the first data field in Flat SQL Queries. In Group SQL and Tree Model Queries, you can select the data field to which to apply Top N.

In List and Action List views, you must apply sorting to a field before applying top N. The top N selection applies to the first sorted field.

If two or more values are the same, and not all of them can be included in the top N, which values are included is random. For example, if N is 3 and the top four are 100, 200, 300, and 300, one of the 300 values is omitted.

See [Creating and Using Business Views](#) for more information about views.

## Applying Aggregate Functions to Groups

Many query types support adding aggregate functions to groups of data.

The aggregate functions are defined in [Table 2](#).

 **Note:**

When you use an aggregation function such as an average in a query, the first data point the query fetches will be null, because the aggregation of no data is null.

The Sum function is only supported for the integer data type. For decimal or float data types, exact values are not guaranteed.

**Table 6-2 Aggregate Functions**

Function	Description
Count	For a field of any type, this function returns a count of the values, including duplicates.
Count Distinct	For a field of any type, this function returns a count of the unique values, not including duplicates.
Minimum	For a field of any type, this function returns the earliest, lowest, or alphanumeric first value in the group.
Maximum	For a field of any type, this function returns the latest, highest, or alphanumeric last value in the group.
Sum	For a numeric field, this function returns the sum.
Average	For a numeric field, this function returns the statistical average.
Median	For a numeric field, this function returns the statistical median. For a datetime field, it returns the middle chronological value. For a string field, it returns the middle alphanumeric value.
Std Dev	For a numeric field, this function returns the statistical standard deviation.
Variance	For a numeric field grouped by a dimension, this function returns the statistical variance accounted for by a particular group.
Percent of Total	For a numeric field grouped by a dimension, this function returns the group sum percentage of the sum of all groups. This function is not appropriate for fields that may sum to zero.

## Using a Time Group or a Time Series

You can create a query and a corresponding view in which the grouping (X-axis) is based on a datetime field.

### To configure a time group or series:

1. Select a datetime field as a **Dimension** in a Group SQL Query or as a **Group By** field in a Continuous Query.

To change a time group or series you have already configured, select the datetime field and click the **Edit** icon at the top of the Dimensions list.

The Time Grouping dialog appears.

2. Select either **Use Time Series** or **Use Time Groups**.
  - **Use Time Series** — Displays the data from the first datetime data point available in the data object to the last in the configured time interval.

- **Use Time Groups** — Displays data grouped into a set number of time intervals. For example, if you select Month from the time unit list, all data from January from all years where data is available are grouped in one data point on the chart.

 **Note:**

When using time-based queries, drilling hierarchy is only supported for queries with one dimension and single time units.

3. Select **Continuous Time Series** to display empty groups for time intervals with no data. This setting appears only if you select **Use Time Series**.

There may be time gaps in which the data object does not have entries. The Continuous Time Series feature adds groups to the result with values of zero, so that the X-axis appears as a smooth time series on a graph. The zero-value result may not appear at design time but shows up in your business view at run time.

Continuous Time Series is valid only if you have chosen a single datetime field to group by. Continuous Time Series is not supported if any additional group fields are selected.

4. Select a **Time Unit** from the list.

If you selected **Use Time Series**, you can select one time unit. If you selected **Use Time Groups**, you can select more than one.

- **Year** displays groups for all of the years where data is available. You can specify a **Quantity** of 1, 2, 5, or 10 years to include in each group.
- **Quarter** displays four groups representing the quarters of a year (January-March, April-June, July-September, and October-December). You can specify a **Quantity** of 1 quarter.

 **Note:**

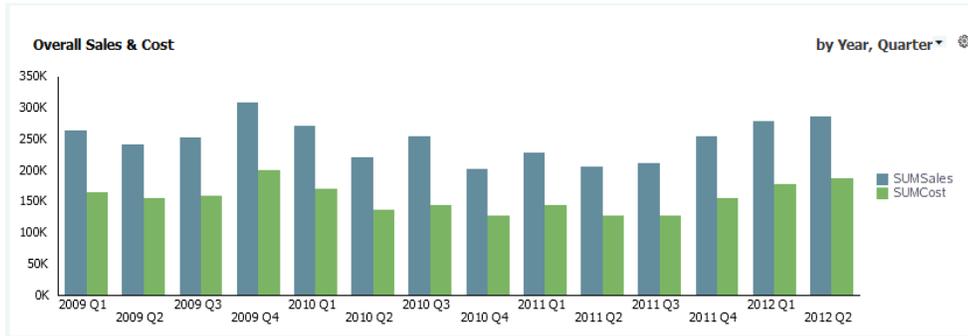
If you use a Group SQL Query with the Continuous Time Series option, the **Quarter** time unit does not work when applied to graph views.

- **Month** displays twelve groups representing the months of the year. You can specify a **Quantity** of 1, 2, 3, 4, 6, or 12 months.
- **Week** displays 52 groups representing the weeks in a year. You can specify a **Quantity** of 1, 2, 4, 13, 26, or 52 weeks.
- **Day** displays 365 groups representing the days in a year when it is the only time unit selected. Any additional time unit larger than Day in a time group determines the number of groups. For example, if you select Week and Day, seven groups representing the days of the week are displayed. You can specify a **Quantity** of 1 day.
- **Hour** displays 24 groups representing the hours of a day. You can specify a **Quantity** of 1, 2, 3, 4, 6, 8, 12, or 24 hours.
- **Minute** displays 60 groups representing the minutes in an hour. You can specify a **Quantity** of 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, or 60 minutes.
- **Second** displays 60 groups representing the seconds in a minute. You can specify a **Quantity** of 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, or 60 seconds.

5. Enter a **Quantity** of the time unit to group by. For example, entering a 2 next to the Month time unit displays the groups in two month increments (January and February are grouped as one data point on the chart).
6. Click **OK**.

Figure 6-1 shows a bar chart view with time groups for years and quarters.

**Figure 6-1 Bar Chart with Year and Quarter Time Groups**



**Note:**

Chart and Graph business views display non-continuous time series differently, in that graph business views space time unit data points along the x-axis evenly.

# 7

## Creating KPIs

This chapter explains how to create Key Performance Indicators (KPIs) in Oracle Business Activity Monitoring (BAM).

This chapter includes the following sections:

- [Understanding KPIs](#)
- [KPI Prerequisites and Uses](#)
- [Creating a Realtime KPI](#)
- [Creating a Scheduled KPI](#)
- [Editing a KPI](#)
- [Renaming a KPI](#)
- [Deleting a KPI](#)
- [Securing a KPI](#)

## Understanding KPIs

There are two major types of performance indicators: outcome metrics and driver metrics. KPIs are outcome metrics that compare performance measures to specific business goals that roll up into larger organizational strategies requiring monitoring, improvement, and evaluation.

Outcome metrics, sometimes called lagging indicators, measure business activity that a strategy is designed to achieve, such as return rate per month or average call processing time for the last moving hour. Driver metrics, sometimes called leading indicators, measure business activity that influences the outcome metrics.

KPI values vary with time, have targets to determine performance status, include dimensions to allow for more specific analysis, and can be compared over time for trending purposes and to identify performance patterns.

You can define two types of KPIs in BAM: Realtime KPIs and Scheduled KPIs. A Realtime KPI collects data and performs incremental calculations of a measure continuously. For example, a call center might update the number of waiting calls every time a new call arrives. A Scheduled KPI collects data and recalculates the aggregation from a specific time at a regular interval. For example, a call center might update the number of unresolved customer complaints every week.

A KPI's status is determined by comparing its value against high and low thresholds that you define. For example, for a KPI that measures product sales in thousands of units, you can define the ranges described in [Table 1](#).

**Table 7-1 Sample Thresholds and Statuses**

Evaluation Range	Threshold	Status
Values exceeding 120 indicate superior performance	121 and above	High
Values between 80 and 120 indicate good performance	80 - 120	Medium

**Table 7-1 (Cont.) Sample Thresholds and Statuses**

Evaluation Range	Threshold	Status
Values less than 80 indicate poor performance	79 and below	Low

Risk indicators are driver metrics that provide early warning signs to identify events likely to jeopardize KPIs. For example, a risk indicator for sales might be the number of units in stock, which can block completion of sales if it is too low. You can associate a risk indicator with a Realtime KPI.

## KPI Prerequisites and Uses

KPI prerequisites are a list of tasks you must complete to ensure that all KPIs you create function smoothly.

Before you can create a KPI, you must do the following:

- Add to your project the data object containing the measure the KPI will analyze. For more information about projects, see [Planning and Creating Projects](#). For more information about data objects, see [Working with Data Objects](#).
- If you are creating a Scheduled KPI, create a Group SQL Query with no dimensions on which to base the KPI. For more information about group queries, see [Creating a Group SQL Query](#).
- If you need to associate a risk indicator with a Realtime KPI, create a Continuous Query for the risk indicator. For more information about continuous queries, see [Creating a Continuous Query](#).

After you create KPIs, you can use them in the following ways:

### In a KPI Watchlist

You can create a special view called a KPI Watchlist that uses a group of related KPIs. After you save it, you can add it to dashboards. For more information about KPI Watchlists, see [Creating KPI Watchlist Business Views](#). For more information about dashboards, see [Creating Dashboards](#).

Although you cannot use a KPI directly in any other view, you can use the Group SQL Query on which a Scheduled KPI is based in a Gauge type view. [Creating Gauge Business Views](#).

### To trigger alert actions

You can use the values of a KPI measure (high, medium, low) to initiate different alert actions. For example, you can define an action that notifies you when the value of an Internal Costs KPI exceeds a certain dollar amount. These alert actions are specific to the KPI and do not appear in the left panel navigator under Alerts. For more information about alert actions, see [Alert Actions](#).

## Creating a Realtime KPI

This section outlines the procedure to create a Realtime KPI.

### To create a Realtime KPI:

1. Go to the **Designer** page.
2. Click **KPIs** in the left panel navigator, or right-click **KPIs** and select the **Create** menu item. The KPIs dialog opens.

3. Type a **KPI Name** and optionally edit the **KPI Display Name**, select **Realtime KPI**, and click **Create**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the KPI is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

A tab opens for the new KPI.

4. On the KPI tab, specify the following **Measure** settings:
  - **Description** — An optional description of the measure used in the overall description of the KPI query.
  - **Data Object** — The data object containing the measure the KPI will analyze.
  - **Measure aggregation** — Select AVG, COUNT, COUNTDISTINCT, SUM, MAX, or MIN.  
  
COUNTDISTINCT counts duplicates only once. The options displayed depend on the **Measure** selected. For example, if the **Measure** is a VARCHAR type, only COUNT and COUNTDISTINCT are displayed.
  - **Measure** — The numeric attribute the KPI measures.
  - **Use rolling window** — Optionally check this box to define a period over which and a frequency at which the aggregation is calculated for incoming data. For example, you could update the number of calls completed in the last hour every ten minutes.
  - **Update Interval** — The aggregation is recalculated every time new incoming data is received. The update interval determines how often the result within the rolling window is updated.
  - **Units** — Select hours, minutes, seconds, or milliseconds. This setting applies to both the Update Interval and the Window Size.
  - **Based on** — A date or time column upon which the rolling window calculation is based. For example, choosing Call Start Time enables the system to determine whether a given call is within the rolling window period.
  - **Window Size** — The period over which the aggregation is calculated, from an amount of time in the past to the present. Examples of window sizes are the last 30 minutes or the last 24 hours.
5. Specify the following optional **Threshold** settings:
  - **Description** — An optional description of the threshold used in the overall description of the KPI query.
  - **Threshold** — Click the down arrow button and select one of the following options. More settings are displayed depending on the option you select.
    - **Constant** — Type an integer value in the text box to define the upper end of the high range. The lower end of the low range is zero. This is the default unless the **Measure aggregation** is AVG.
    - **Historical** — Type an integer value in the text box. Select SECOND, MINUTE, HOUR, DAY, WEEK, or MONTH as the units for the integer. Then select a historical reference measure, which can be the same as the KPI measure. The KPI and historical reference measures use the same aggregation.

- **Standard Deviation** — This option is displayed only if the **Measure aggregation** is AVG. In that case, it is the default. No additional options appear.
  - **Deviation Ranges** — Click the down arrow button. Select **constant** (the default) or **percentage** if the Threshold is Constant or Historical. If the Threshold is Standard Deviation, this setting is **standard deviation**.
  - **Deviation Range display** — Click the gear icon to change the **Min**, **Max**, and **Increment** of the range.
  - **High and Low range sliders** — Move the range sliders until they show the desired high and low threshold values. The sliders are set to 1 and –1 by default.
  - **<add action>** — You can optionally add different actions for the High, Medium, and Low ranges of the KPI. For more information about actions, see [Alert Actions](#).
6. Specify the following optional **Risk Indicators** settings:
    - **Description** — An optional description of the risk indicator used in the overall description of the KPI query.
    - **Risk Indicators** — A continuous query of a measure defined in the data object as a driver metric for the KPI measure. For more information about continuous queries, see [Creating a Continuous Query](#).
  7. Click **Save**.

## Creating a Scheduled KPI

This section outlines the procedure to create a Scheduled KPI.

### To create a Scheduled KPI:

1. Go to the **Designer** page.
2. Click **KPIs** in the left panel navigator, or right-click **KPIs** and select the **Create** icon.  
The KPIs dialog opens.
3. Type a **KPI Name** and optionally edit the **KPI Display Name**, select **Scheduled KPI**, and click **Create**.  
The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the KPI is created.  
The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.  
A tab opens for the new KPI.
4. On the KPI tab, specify the following **Measure** settings:
  - **Description** — An optional description of the measure used in the overall description of the KPI query.
  - **Business Query** — The query of the measure the KPI will analyze. Only group queries with no dimensions are listed. For more information about queries, see [Creating Business Queries](#).
  - **Measure aggregation** — Select AVG, COUNT, COUNTDISTINCT, SUM, MAX, or MIN. COUNTDISTINCT counts duplicates only once. The options you can select for this setting depend on the query you select.

- **Every** — Select SECOND, MINUTE, HOUR, DAY, or MONTH.
  - **Starting on** — Type a date, or click the Select Date icon and select a date from the pop-up calendar. The default is today.
  - **At** — Type integers for the hour and minute, or use the Increment and Decrement buttons. The hour box uses 24-hour time, so 2 pm is 14, for example.
5. Specify the following optional **Threshold** settings:
- **Description** — An optional description of the threshold used in the overall description of the KPI query.
  - **Threshold** — Click the down arrow button and select one of the following options. The options you can select for this setting depend on the query you select. More settings are displayed depending on the option you select.
    - **Constant** — Type an integer value in the text box to define the upper end of the high range. The lower end of the low range is zero. This is the default unless the **Measure aggregation** is AVG.
    - **Historical** — Type an integer value in the text box. Select SECOND, MINUTE, HOUR, DAY, WEEK, or MONTH as the units for the integer. Then select a historical reference measure, which can be the same as the KPI measure. The KPI and historical reference measures use the same aggregation.
    - **Standard Deviation** — This option is displayed only if the **Measure aggregation** is AVG. In that case, it is the default. No additional options appear.
  - **Deviation Ranges** — Click the down arrow button. Select **constant** (the default) or **percentage** if the Threshold is Constant or Historical. If the Threshold is Standard Deviation, this setting is **standard deviation**.
  - **Deviation Range display** — Click the gear icon to change the **Min**, **Max**, and **Increment** of the range.
  - **High and Low range sliders** — Move the range sliders until they show the desired high and low threshold values. The sliders are set to 1 and –1 by default.
  - **<add action>** — You can optionally add different actions for the High, Medium, and Low ranges of the KPI. For more information about actions, see [Alert Actions](#).
6. Click **Save**.

## Editing a KPI

When you edit a KPI, the changes propagate to all KPI watchlists that include the KPI and all dashboards that include these KPI watchlists.

Use the following procedure to open, edit, and save a KPI.



### Note:

If you edit the group query upon which a Scheduled KPI is based, do not add a dimension, or the KPI will not work.

### To edit a KPI:

1. In the left navigation pane, click the arrow to the left of **KPIs**.

All saved KPIs in the current project are displayed in a list.

2. Click the KPI name, or right-click the KPI and select the **Edit** menu item.
3. Make the desired changes.
4. Click **Save**.

## Renaming a KPI

This section outlines the procedure to rename a KPI.

Use the following procedure to change the **Display Name** of a KPI.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

### To rename a KPI:

1. In the left navigation pane, click the arrow to the left of **KPIs**.  
All saved KPIs in the current project are displayed in a list.
2. To rename the KPI, right-click the KPI name and select the **Rename** menu item.  
The name becomes text in an editable field.
3. Type the new name and press **Enter**.  
The new name appears in the list.

When you edit the KPI, the new **Display Name** is displayed on the tab. However, the internal **Name** remains unchanged. When you edit the KPI, this name is displayed on the left in the header.

## Deleting a KPI

This section outlines the procedure to delete a KPI.

Use the following procedure to delete a KPI. The KPI is removed from any views or dashboards that reference it.

### To delete a KPI:

1. In the left navigation pane, click the arrow to the left of **KPIs**.  
All saved KPIs in the current project are displayed in a list.
2. Click the KPI icon and click the **Delete** icon, or right-click the KPI name and select the **Delete** menu item.  
A dialog asks you to confirm the KPI deletion.
3. Click **OK**.  
The KPI disappears from the list.

## Securing a KPI

A KPI inherits security settings from the project in which it is created.

For more information about projects, see [Planning and Creating Projects](#).

**To change security settings for a KPI:**

1. In the left navigation pane, click the arrow to the left of **KPIs**.  
All saved KPIs in the current project are displayed in a list.
2. Right-click the KPI and select **Security Settings** from the pop-up menu.  
The security tab for the KPI opens.
3. To add a role or group to whom you can explicitly grant or deny permissions, follow these steps:
  - a. Click the **Add** icon in the Grant Permissions or Deny Permissions table.  
The Add Application Roles, Groups, and Users dialog opens.  
See [Managing Oracle BAM Users](#) for information about how to add users to roles and groups.
  - b. Type a **Name** for the role or group you are adding.
  - c. Select from the drop-down **List**: Application Role or Group.
  - d. Click **Search** to populate the Available Members list.
  - e. To add a member to the Selected Members list, select the member and click the single right arrow.
  - f. To add all members to the Selected Members list, select the member and click the double right arrow.
  - g. To remove members from the Selected Members list, use the single and double left arrows.
  - h. When the Selected Members list is final, click **OK**.  
The Add Application Roles, Groups, and Users dialog closes, and the **Name** you specified appears in the table.
4. To remove a role or group, select the table row and click the **Remove** icon.
5. To grant permissions, select **Read**, **Write**, **Remove**, or **Security** for the users, roles, and groups listed in the Grant Permissions table.
6. To deny permissions, select **Read**, **Write**, **Remove**, or **Security** for the users, roles, and groups listed in the Deny Permissions table.
7. Click **Save**.

# 8

## Creating and Using Business Views

This chapter describes how to create and format different types of Oracle Business Activity Monitoring (Oracle BAM) views, which display data fetched by business queries and key performance indicators (KPIs).

This chapter includes the following sections:

- [Understanding Business Views](#)
- [Business View Prerequisites and Uses](#)
- [Creating Business Views](#)
- [Formatting Business Views](#)
- [Runtime Interactions with Business Views](#)
- [Editing, Renaming, Deleting, and Securing Business Views](#)

### Understanding Business Views

A *business view* or *view* is a visual representation of data fetched by a business query or one or more KPIs.

In most views, numeric data fields, called *measures*, are grouped by non-numeric data fields, called *dimensions*. For example, sales, a measure, can be grouped by country, a dimension. An *aggregation* is a computation based on multiple values in a data field, such as a sum or average.



#### Note:

The underlying query determines the order in which a view displays multiple measures or dimensions. See [Creating Business Queries](#) for more information.

[Table 1](#) lists view categories and view types within each category.

Oracle BAM uses various view types based on Oracle Application Development Framework's Data Visualization Tools (DVT). They are:

- Graph Views
- Chart Views
- Table Views
- Gauge Views
- Geo Map View
- Treemap View
- KPI Watchlist View

Area Chart, Bar Chart, Horizontal Bar Chart, Line Chart, Pie Chart, Combo Chart, Scatter Chart, and Bubble Chart views are collectively called *chart* views. There are corresponding *graph* view variations of each of these views.

Some graph views are *switchable*. They can be changed to a compatible type. Provided that queries are compatible, some views are switchable with each other. These are Area, Bar, Horizontal Bar, Line, Pie, and Combo views. Gauge views are switchable within the category.

You can format views in many ways, changing colors, fonts, labels, thresholds, and other properties. See [Formatting Business Views](#) for more information.

You can configure view characteristics that can be changed dynamically at runtime. You can configure how users can display more detailed data, which is called *drilling*. You can configure *actions* that the end user can perform on the view. You can also configure *active data*, which changes the query on which the view is based to a continuous query. See [Runtime Interactions with Business Views](#) for more information.

**Table 8-1 Business View Categories and Types**

Icon	Category	Business View Types	Business Query Requirements	Recommended If
	Area	Area Stacked Area Area Chart	Group SQL Query	The X-axis is a single consecutive dimension, such as a datetime field, and the data lines are unlikely to cross.
	Bar	Bar Dual-Y Bar Stacked Bar Dual-Y Stacked Bar Bar Chart	Group SQL Query	The horizontal axis displays multiple or non-consecutive dimensions.
	Horizontal Bar	Horizontal Bar Dual-Y Bar Stacked Bar Dual-Y Stacked Bar Horizontal Bar Chart	Group SQL Query that does not use a time series	The vertical axis displays multiple or non-consecutive dimensions.
	Line	Line Dual-Y Line Stacked Line Dual-Y Stacked Line Line Chart	Group SQL Query	The X-axis is a single consecutive dimension, such as a datetime field, and the data lines are likely to cross.
	Pie	Pie Multiple Pie Ring Multiple Ring Pie Chart	Group SQL Query	You are analyzing a single measure by a single dimension, or for a Multiple variation, several measures by the same dimension, and your data has no negative values.
	Combo	Combo Graph Combo Chart	Group SQL Query	You are comparing very different measures.

**Table 8-1 (Cont.) Business View Categories and Types**

Icon	Category	Business View Types	Business Query Requirements	Recommended If
	Table	List Action List Collapsed List Pivot Table	List and Action List: Flat SQL Query  Collapsed List: Group SQL Query with one or more aggregation types, one or more measures, and one or more dimensions  Pivot Table: Group SQL Query with one or more aggregation types, one or more measures, and two or more dimensions	You want to display exact numbers instead of a graph, or to show data in tabular format.
	KPI Watchlist	KPI Watchlist	One or more Realtime or Scheduled KPIs	You are interested in a set of related critical measures.
	Gauge	Dial Dial with Thresholds Arrow Statusmeter Vertical Statusmeter Circular Status Meter Horizontal Status Meter	Group SQL Query with no dimensions	You are interested in the aggregated value of a single critical measure.
	Geo Map	Point Theme Color Theme GPS Theme	Point Theme: Flat SQL Query with address and country code fields or latitude and longitude fields, and one measure  GPS Theme: Flat SQL Query with latitude and longitude fields and one measure  Color Theme: Flat SQL Query, one measure, see <a href="#">Table 7-7, Geo Map Color Theme General Properties</a> for location fields	Geography is especially important in analyzing a measure.
	Scatter	Scatter Graph Scatter Chart	Group SQL Query with at least two measures	You are interested in the relationship between two measures.
	Bubble	Bubble Graph Bubble Chart	Group SQL Query with at least three measures	You are interested in the relationship between three measures.
	Treemap	Treemap	Tree Model Query	Dimensions are hierarchical, for example: country, state or province, and city.

## Business View Prerequisites and Uses

Business View prerequisites are tasks you must complete before you create a business view, to ensure its smooth functioning.

Before you can create a business view, you must do the following:

- Add to your project the data object containing the data the view will display. For more information about projects, see [Planning and Creating Projects](#). For more information about data objects, see [Working with Data Objects](#).
- If you are creating a KPI Watchlist type view, create one or more KPIs on which to base the view. For more information about KPIs, see [Creating KPIs](#).
- If you are creating any other view, create a business query on which to base the view. See the Business Query Requirements column in [Table 8-1](#) for the type of query to create. See [Creating Business Queries](#) for more information about queries.

You cannot use a Continuous Query in any view. However, you can make data active in most views. See [Using Active Data](#) for more information.

After you create business views, you can use them in the following ways:

### In a dashboard

A dashboard contains a group of related business views. To display data to users, a view must be included in a dashboard. See [Creating Dashboards](#) for more information.

You can use a bottom-up approach and create views before creating a dashboard. Or you can use a top-down approach and create a dashboard before creating the views within it. See [Creating a Dashboard and Adding Views](#) and [Choosing Bottom-Up or Top-Down Project Building Steps](#) for more information.

### With a parameter

Business views do not interact with parameters directly. However, you can filter query data using a user-specified parameter value, which changes the data displayed in views that are included in a dashboard. See [Creating Parameters](#) for more information.

Views do not interact with Alerts. However, you can configure view-specific actions. See [Using Actions in Business Views](#) for more information.

If you are using Oracle JET visualizations to view business views, ensure that you use value formatting on Table business views for the values to reflect correctly as described in a data object. This is because the precision for decimal figures extends to only three decimal points. If you have more, you must use value formatting to render them as defined.

## Creating Business Views

This section describes how to create and use each of the views in Oracle BAM.

This section contains the following topics:

- [Creating Chart and Graph Business Views](#)
- [Creating Table Business Views](#)
- [Creating KPI Watchlist Business Views](#)
- [Creating Gauge Business Views](#)
- [Creating Geo Map Business Views](#)

- [Creating Scatter and Bubble Business Views](#)
- [Creating Treemap Business Views](#)

## Creating Chart and Graph Business Views

Business views are broadly classified as Graph and Chart views. Chart views are based on Data Visualization Tools (DVT) that provide enhanced capabilities in addition to what Graph views offer. Here are some properties specific to Chart views:

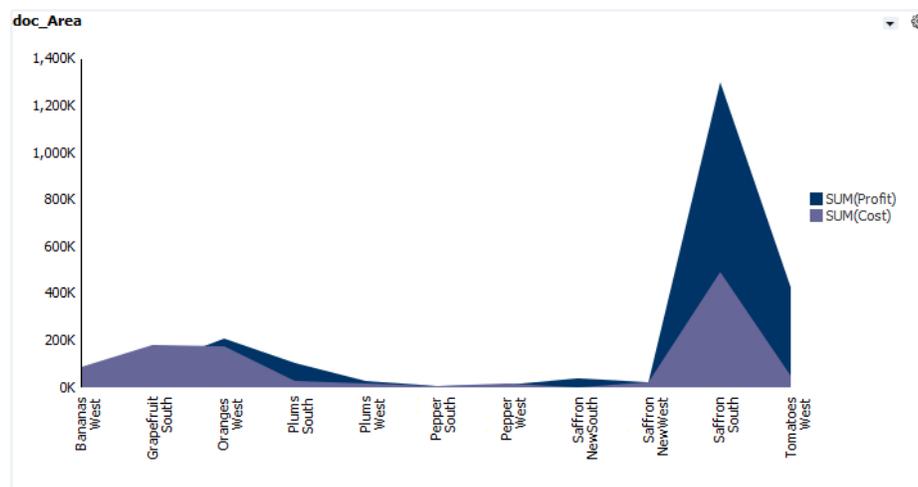
- Sliding zoom wheel at the bottom of a Chart view. For bar, line, area, and combo charts, you can use the preview control located below the chart to narrow down the X-axis range and display data for the selected range.
- Hide and show behavior: You can click legend buttons to hide or show selected groups or series from the chart.
- “Intelligent” time axis layout: When the X-axis of a bar, line, area, or combo chart is based on time value, the DVT chart will make decisions on the spacing between neighboring groups.

For example, consider the time groups: 1st-January, 1st February, and 1st November. Because the time interval between 1st-January and 1st-February is much lesser than the time interval between 1st February and 1st November, the 1st and 2nd groups will be located much closer to each other than the 2nd and 3rd groups. Also, the number of group labels for time groups shown on the X-axis will vary based on the selected data range. For example, if you view chart data for entire range, the X-axis labels will show only the year. If you zoom in on a subset of the data, you will start to see months and days.

Business views include all those in the following categories:

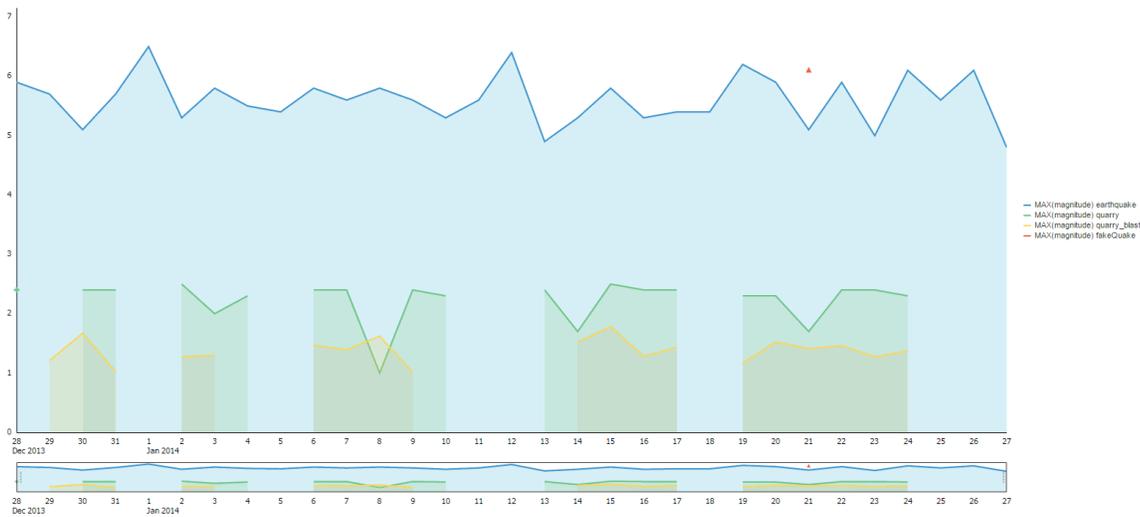
An **Area** view shows a graphed line of values with the area filled in for visual comparison.

**Figure 8-1 Area Graph Example**



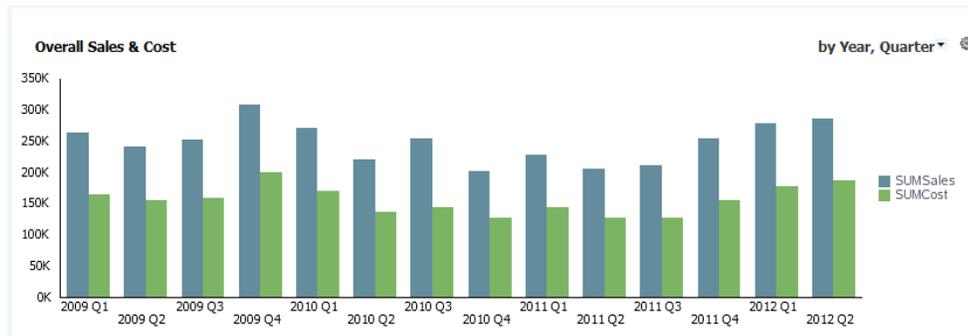
In comparison with the graph view, here's what the Chart view looks like:

**Figure 8-2 Area Chart Example**



A **Bar** view shows vertical columns to represent summarized values.

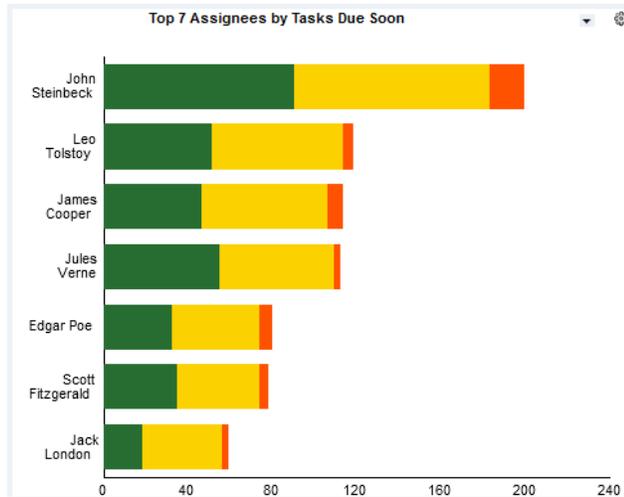
**Figure 8-3 Bar Graph Example**



A **Horizontal Bar** view shows horizontal columns to represent summarized values.

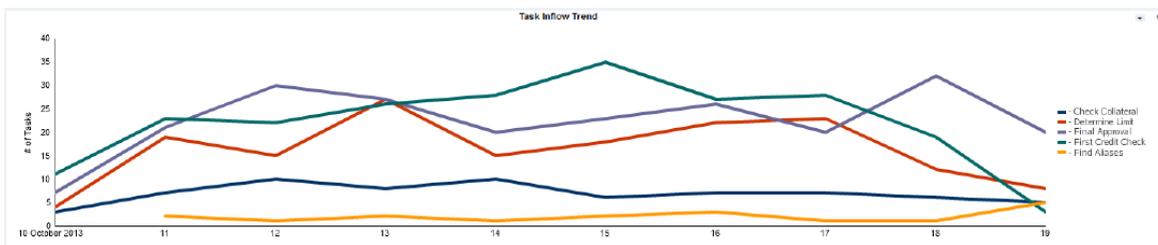
In a Horizontal Bar chart, the X-axis is vertical and the Y-axis is horizontal, unlike the standard convention. This is necessary because you can switch from one graph view to another and retain the axis properties.

**Figure 8-4 Horizontal Bar Graph Example**



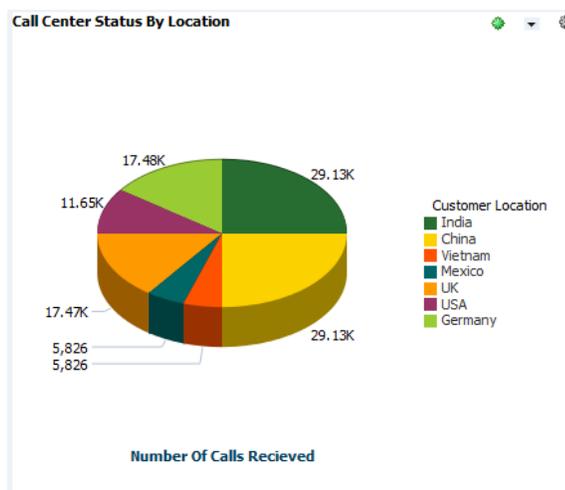
A **Line** view shows a graphed line of values or compared values.

**Figure 8-5 Line Graph Example**



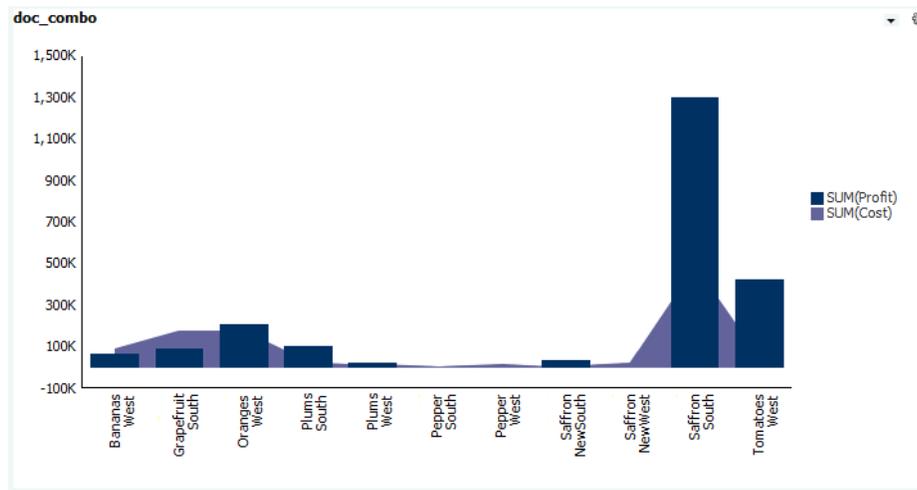
A **Pie** view displays values in segments of a circle.  
If your data contains negative values, no pies are displayed.

**Figure 8-6 Pie Graph Example**



A **Combo** view shows a combination of values as areas, bars, and lines in the same view.

**Figure 8-7 Combo Graph Example**

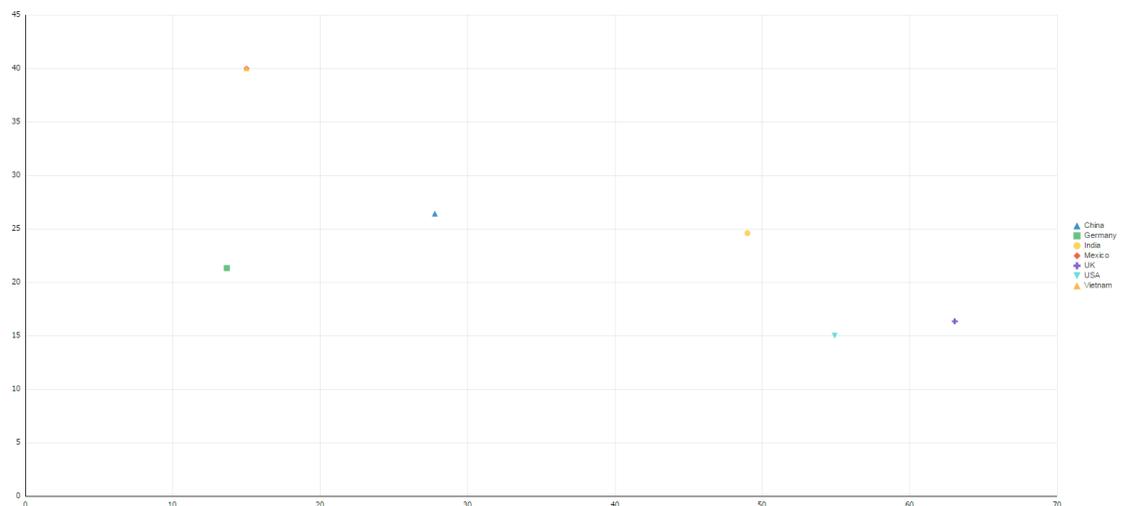


**Note:**

If the data is not continuous, or if the underlying query uses a Legend, using a Bar or Horizontal Bar graph is recommended. Null values in Area, Line, and Combo graphs can result in missing lines.

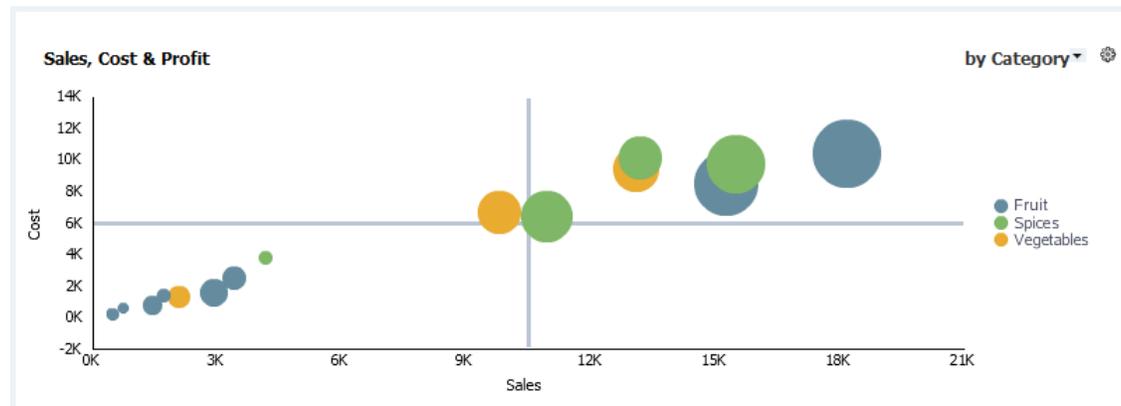
**Scatter views** are like Bubble views, except that X,Y coordinates appear as points that vary in shape and color, instead of bubbles. Except for the selection of a **Bubble Size** data field, the steps to create a Scatter view are the same as those to create a Bubble view.

**Figure 8-8 Scatter Graph Example**



Here is an example of a **Bubble Graph** view.

**Figure 8-9 Bubble Graph Example**



If you use a Bubble Chart and the Bubble Size attribute is a negative value, the corresponding bubble is not displayed.

There are common variations among the view types:

- A *Dual-Y* view has two Y-axes, one on the left and another on the right.
- A *Stacked* view displays sets of values stacked in a single segmented column instead of side-by-side in separate columns. [Figure 8-4](#) shows a stacked horizontal bar chart with segmented bars.
- A *Dual-Y Stacked* view is a combination of *Dual-Y* and *Stacked*.

 **Note:**

If a measure in an Area chart is hidden behind another measure, changing the order of measures in the underlying query or switching to a Stacked Area chart is recommended.

 **Note:**

When using multiple measures, you can sort which columns should appear first. The order will change how business view legends, bar graph and chart grouping-orders, and stacked charts display.

A view displays data from a Group SQL Query. See [Creating a Group SQL Query](#) for more information.

Although the views have differences when displayed, the steps to create any view are the same. In general, one graph view can be switched to any other. Note that Chart Switching does not apply to DVT-based chart views. Chart switching has some limitations:

- Charts that use Active Data can only be switched to basic chart types, not variations such as *Dual-Y* or *Stacked*. See [Using Active Data](#) for more information.

- Charts based on queries that use time series cannot be switched to any Horizontal Bar type. See [Using a Time Group or a Time Series](#) for more information.
- If you switch to a Pie chart and your data contains negative values, no pies are displayed.

 **Note:**

When using graph business views and using one view to drive results in another, attempting to drill to details in the target/driven view sometimes shows a `no data to display` message. This only occurs on the first drill-to-details attempt after driving. Attempt to drill to details again.

## Creating a Business View: Steps

You can create views from the Designer page.

### To create a view:

1. Go to the **Designer** page.
2. Click **Business Views** in the left panel navigator, or right-click **Business Views** and select the **Create** menu item.

The Business Views dialog opens.

3. Type a **Name** and optionally edit the **Display Name**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the view is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

4. Select a **Category**.
  5. Select a thumbnail representing the desired view type.
  6. Click **Create**.
- A tab opens for the new view, with a visual preview using generic data.
7. Select a query from the **Query** drop-down list. Only Group SQL Queries are listed.
- The preview changes to reflect the data fields fetched by the query.
8. To change the formatting of the view, click **Properties**. See these sections for details about formatting properties that apply to chart views:

- [Common General Properties](#)
- [Chart, Scatter, and Bubble General Properties](#)
- [Label, Text, Title, and Value Properties](#)
- [X-Axis Properties](#)
- [Y-Axis Properties](#)
- [Legend Properties](#)
- [Target Properties](#)

- [Slice Properties](#)

To save the properties you have edited, click **Apply**.

You can use the JET-based properties editor to modify the business view properties. The sample URL for the console is `http://host:port/bam/composer/streamconsole.html`. Using the JET-based editor, you can view the visualization and modify the properties of the selected business view. The changes you make in the JET-based editor will persist in MDS and the ADF-based visualizations will show these changes.

9. To add runtime interactions to the view, click **Runtime-Interaction**. See these sections for details about runtime interactions that apply to chart views:
  - [Using Drilling](#)
  - [Using Actions in Business Views](#)
  - [Using Active Data](#)

10. Click **Save**.

The preview changes to reflect changed property and runtime settings.

**Note:**

Views do not re-save properly after query is changed if view tab stays open while you make the query changes.

## Creating Table Business Views

This section provides information about creating Table views. It contains the following topics:

- [Understanding Table Business Views](#)
- [Creating Table Business Views: Steps](#)
- [Adjusting Column Width in Table Business Views](#)
- [Reordering Columns in Table Business Views](#)
- [Sorting Data in Table Business Views](#)
- [Selecting Editable Fields in an Action List](#)
- [To Enable or Disable Pivot Table Features](#)

## Understanding Table Business Views

A Table view displays data values in table format. The following Table view types are available in Oracle BAM:

A **List** displays a list you can sort. It displays data from a Flat SQL Query.

An **Action List** has values in one or more columns that are editable. It also displays insert, update, and delete operations on the data object without requiring a refresh. It displays data from a Flat SQL Query.

A **Collapsed List** displays rows of aggregated data values grouped by one or more dimensions. You can also apply color-coded thresholds for data values. It displays data from a Group SQL query with one or more aggregation types, one or more measures, and one or more dimensions.

**Figure 8-10 Collapsed List Example**

Application Name	Process Name	Process Summary			Opened Today	Closed Today	Closure % Today	Faulted
		Not Due Soon	Due Soon	Overdue				
Business Account Opening	Open Business	953	106	15	12	18	0.19%	▲ 30
Debt Collection	Collect Debt	20	33	4	2	1	0.04%	3
Employee Expense Approval	Approve Expense	21	22	8	4	6	0.05%	1
Employee Onboarding	Employee Onboarding	146	111	17	8	16	0.15%	17
Incident Reporting	Report Incident	104	92	13	14	11	0.12%	13
Internal Service Request	Request Service	19	25	1	3	2	0.04%	6
Loan Processing	Process Loan	142	100	9	10	10	0.14%	16
Sales Quote	Request Quote	163	86	28	14	12	0.16%	▲ 22
Travel Request Management	Request Travel	111	69	19	13	8	0.12%	15

A **Pivot Table** combines rows and columns to display a multi-dimensional view of aggregated data values. A Pivot Table is summarized vertically and horizontally for columns and rows. It displays data from a Group SQL query with one or more aggregation types, one or more measures, and two or more dimensions.

**Figure 8-11 Pivot Table Example**

pivot							
Category	Product	Region	Measure	SUM(Cost)	MAX(Cost)	SUM(Profit)	AVG(Profit)
Fruit	Bananas	West		68,124.00	84.00	90,832.00	112.00
	Grapefruit	South		92,454.00	114.00	180,853.00	223.00
	Oranges	West		207,616.00	181.00	175,987.00	108.50
	Plums	South		105,624.00	163.00	27,864.00	43.00
Spices	Pepper	West		27,058.00	163.00	16,600.00	100.00
		South		8,130.00	10.00	4,878.00	6.00
	Saffron	NewSouth		12,165.00	15.00	17,031.00	21.00
		NewWest		36,855.00	1,053.00	3,500.00	100.00
Vegetables	Tomatoes	South		20,876.00	614.00	21,760.00	640.00
		West		1,299,471.00	1,053.00	494,130.00	317.16
				426,012.00	524.00	49,593.00	61.00

\*List views include all Table views except Pivot Tables, which are List, Action List, and Collapsed List. \*List views share some features not common to Pivot Tables.

You cannot switch from one Table view type to another.



**Note:**

Table business views imported from previous Oracle BAM releases may have improper alignment, value and color formatting. To fix this, either manually update all previous business view properties, or simply click **Save** to allow them to retain the default formatting.

See [Creating a Flat SQL Query](#) and [Creating a Group SQL Query](#) for more information.

## Creating Table Business Views: Steps

The steps to create any Table view are similar. Differences between the view types are noted.

**To create a Table view:**

1. Go to the **Designer** page.
2. Click **Business Views** in the left panel navigator, or right-click **Business Views** and select the **Create** menu item.

The Business Views dialog opens.

3. Type a **Name** and optionally edit the **Display Name**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the view is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

4. Select **Table** as the Category.
5. Select a thumbnail representing the desired view type.
6. Click **Create**.

A tab opens for the new view.

7. Select a query from the **Query** drop-down list.

Only queries meeting the view requirements are listed: Flat SQL Queries for List and Action List view types, or Group SQL Queries for Collapsed List and Pivot Table types.

A visual preview reflects the data fields fetched by the query.

8. To change the formatting of the view, click **Properties**. See these sections for details about formatting properties that apply to Table views:

- [Common General Properties](#)
- [List General Properties](#)
- [General Properties Specific to Pivot Table Business Views](#)
- [Label, Text, Title, and Value Properties](#)
- [Text Properties](#)
- [List Layout Properties](#)
- [Editable Fields Properties](#)
- [Pivot Table Data Properties](#)
- [Collapsed List Thresholds Properties](#)

To save the properties you have edited, click **Apply**.

9. To add runtime interactions to the view, click **Runtime-Interaction**. See these sections for details about runtime interactions that apply to Table views:

- [Using Drilling](#)
- [Using Actions in Business Views](#)
- [Using Active Data](#)

For List and Action List views, only Drill Across is supported. For Collapsed List views, Drill to Detail, and Drill Across are supported. Drilling is not supported for Pivot Table views.

10. Click **Save**.

The preview changes to reflect changed property and runtime settings.

## Adjusting Column Width in Table Business Views

If the values in a column are wider than the column width, you can change the width manually in the view by dragging the separator between the column headers.

The **Resizing** Layout property is enabled by default. This allows you to resize columns in a List, Action List, or Collapsed List while editing it.

## Reordering Columns in Table Business Views

The **Reordering** Layout property is enabled by default. This allows you to rearrange the order of columns in a List, Action List, or Collapsed List while viewing it in the BAM Viewer (the Home page).

## Sorting Data in Table Business Views

For a List, Action List, or Collapsed List, click on the column header of the field you want to sort. An up or down arrow appears next to the column header to indicate whether this field is sorted in ascending or descending order.

### To configure Pivot Table sorting:

1. From the view tab, click **Properties**.
2. Click **Data**.
3. For each data field in the Sorting table, select Ascending or Descending from the drop-down list.
4. Click **Apply**.

## Selecting Editable Fields in an Action List

Editable fields are what distinguish an Action List from a regular List.

### To enable editing of data fields:

1. From the view tab, click **Properties**.
2. Click **Editable Fields**.
3. Check the box for each desired field.
4. Click **Apply**.

When you view the Action List in the BAM Viewer (the Home page), values in editable field columns appear in text boxes. You can edit each table cell as needed. To enable these edits to be saved, you must configure a **Save** action. See [Using Actions in Business Views](#) for more information.

### **Note:**

Custom Actions are now supported in Action Lists and you may experience errors if they have not placed the jar files in a location which can be accessed by Oracle BAM.

## To Enable or Disable Pivot Table Features

A Pivot Table view has features not present in the other Table views.

1. From the view tab, click **Properties**.
2. Click **General**.
3. Check or uncheck **Pivot Enable**.  
The default is checked.
4. Click **Apply**.

## To Pivot the View of Data

1. In the view tab, click and drag a heading from a column location to a row location.  
The current row headings are highlighted to indicate to where the heading will be moved.
2. Release the mouse button to drop the heading in the new location.  
The data displayed reflects the updated combination of row and column headings.

Figure 8-12 shows a pivot table before pivoting.

**Figure 8-12 Pivot Table Before Pivoting**

pivot							
Category	Product	Region	Measure	SUM(Cost)	MAX(Cost)	SUM(Profit)	AVG(Profit)
Fruit	Bananas	West		68,124.00	84.00	90,832.00	112.00
	Grapefruit	South		92,454.00	114.00	180,853.00	223.00
	Oranges	West		207,616.00	181.00	175,987.00	108.50
	Plums	South		105,624.00	163.00	27,864.00	43.00
		West		27,058.00	163.00	16,600.00	100.00
Spices	Pepper	South		8,130.00	10.00	4,878.00	6.00
		West		12,165.00	15.00	17,031.00	21.00
	Saffron	NewSouth		36,855.00	1,053.00	3,500.00	100.00
		NewWest		20,876.00	614.00	21,760.00	640.00
		South		1,299,471.00	1,053.00	494,130.00	317.16
Vegetables	Tomatoes	West		426,012.00	524.00	49,593.00	61.00

Figure 8-13 shows a pivot table after pivoting. The Region column has been moved between the Category and Product columns.

**Figure 8-13 Pivot Table After Pivoting**

pivot							
Category	Region	Product	Measure	SUM(Cost)	MAX(Cost)	SUM(Profit)	AVG(Profit)
Fruit	South	Grapefruit		92,454.00	114.00	180,853.00	223.00
		Plums		105,624.00	163.00	27,864.00	43.00
	West	Bananas		68,124.00	84.00	90,832.00	112.00
		Oranges		207,616.00	181.00	175,987.00	108.50
		Plums		27,058.00	163.00	16,600.00	100.00
Spices	NewSouth	Saffron		36,855.00	1,053.00	3,500.00	100.00
		NewWest		20,876.00	614.00	21,760.00	640.00
	South	Pepper		8,130.00	10.00	4,878.00	6.00
		Saffron		1,299,471.00	1,053.00	494,130.00	317.16
		West	Pepper		12,165.00	15.00	17,031.00
Vegetables	West	Tomatoes		426,012.00	524.00	49,593.00	61.00

## To Add A Category Aggregate Function to a Pivot Table

1. From the view tab, click **Properties**.
2. Click **Data**.
3. Select the **Aggregate Position**: None, After, or Before.
4. Select the **Aggregate Function**: Sum, Average, Count, Minimum, or Maximum.
5. Click **Apply**.

## Creating KPI Watchlist Business Views

A KPI Watchlist displays the following information for one or more key performance indicators (KPIs):

- **KPI** lists KPI names.
- **Risks** lists the color-coded status of risk indicators associated with each KPI.
- **Actual** shows the most recent KPI value.
- **Today** shows a line graph of the last ten KPI values. Older KPI values are on the left and newer values are on the right.
- **Target** shows a horizontal bar that compares the current KPI value to a target value.

**Figure 8-14 KPI Watchlist Example**

Call Center Watchlist				
KPI	Risks	Actual	Today	Target
Call Volume		116,510		
Processing Time		37.7		
Wait Time		23.9		
Escalated Calls		17,481		
Dropped Calls		104,856		

See [Creating KPIs](#) for more information about how to create KPIs.

### To create a KPI Watchlist view:

1. Go to the **Designer** page.
2. Click **Business Views** in the left panel navigator, or right-click **Business Views** and select the **Create** menu item.

The Business Views dialog opens.

3. Type a **Name** and optionally edit the **Display Name**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the view is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

4. Select **KPI Watchlist** as the Category.

5. Select the **KPI Watchlist** thumbnail.
6. Click **Create**.  
A tab opens for the new view, with a visual preview of the column headings.
7. Select one or more KPIs from the drop-down list. To specify all KPIs, check **All**.  
The preview changes to reflect the selected KPIs.
8. Click **Save**.

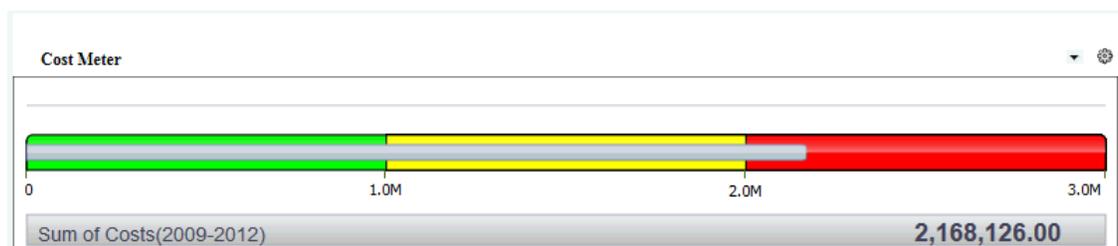
## Creating Gauge Business Views

A gauge view displays a single value as a position on a dial or meter, or a number next to a color-coded shape.

The following gauge view types are available in Oracle BAM:

- A **Dial** indicates the current value in a curved numeric marked gauge.
- A **Dial with Thresholds** indicates the current value in a curved numeric marked gauge with color-coded low, medium, and high ranges.
- An **Arrow** is an arrow shape that is color-coded according to whether the value is low, medium, or high. The direction of the arrow indicates whether the current value is higher (up), the same (right), or lower (down) than the previous value.
- A **Statusmeter** indicates the current value in a horizontal numeric marked gauge with color-coded low, medium, and high ranges.
- A **Vertical Statusmeter** indicates the current value in a vertical numeric marked gauge with color-coded low, medium, and high ranges.
- A **Circular Statusmeter Gauge** indicates the current value in a circular gauge representing the numerical sum along a circular axis.
- A **Horizontal Statusmeter Gauge** indicates the current value in a horizontal gauge representing the numerical sum along a horizontal axis.

**Figure 8-15 Statusmeter Gauge Example**



By default, color codes for value ranges are green for low, yellow for medium, and red for high. See [Gauge Thresholds Properties](#) for more information.

A gauge view displays data from a Group SQL Query with no dimensions. See [Creating a Group SQL Query](#) for more information.

Although you can only use a KPI directly in a KPI Watchlist view, you can use the Group SQL Query on which a Scheduled KPI is based in a Gauge type view. [Creating a Scheduled KPI](#).

Although Gauge views have differences when displayed, the steps to create them are the same, and any one can be switched to any other.

**To create a Gauge view:**

1. Go to the **Designer** page.
2. Click **Business Views** in the left panel navigator, or right-click **Business Views** and select the **Create** menu item.

The Business Views dialog opens.

3. Type a **Name** and optionally edit the **Display Name**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the view is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

4. Select **Gauge** as the Category.
5. Select a thumbnail representing the desired view type.
6. Click **Create**.

A tab opens for the new view, with a visual preview using generic data.

7. Select a query from the **Query** drop-down list. Only Group SQL Queries with no dimensions are listed.

The preview changes to reflect the data fields fetched by the query.

8. Select from the drop-down list the measure data field for which the Gauge will display the value.

The preview changes to reflect the selection.

9. To change the formatting of the view, click **Properties**. See these sections for details about formatting properties that apply to Gauge views:

- [Common General Properties](#)
- [Gauge General Properties](#)
- [Label, Text, Title, and Value Properties](#)
- [Axis Properties Specific to Dial and Statusmeter Business Views](#)
- [Legend Properties](#)
- [Gauge Thresholds Properties](#)

To save the properties you have edited, click **Apply**.

10. To add runtime interactions to the view, click **Runtime-Interaction**. See these sections for details about runtime interactions that apply to Gauge views:

- [Using Drilling](#)
- [Using Actions in Business Views](#)
- [Using Active Data](#)

11. To change to a different view type, click **Chart Switcher**, select **Gauge**, then select a view type thumbnail.

The preview changes to reflect the new view type.

12. Click **Save**.

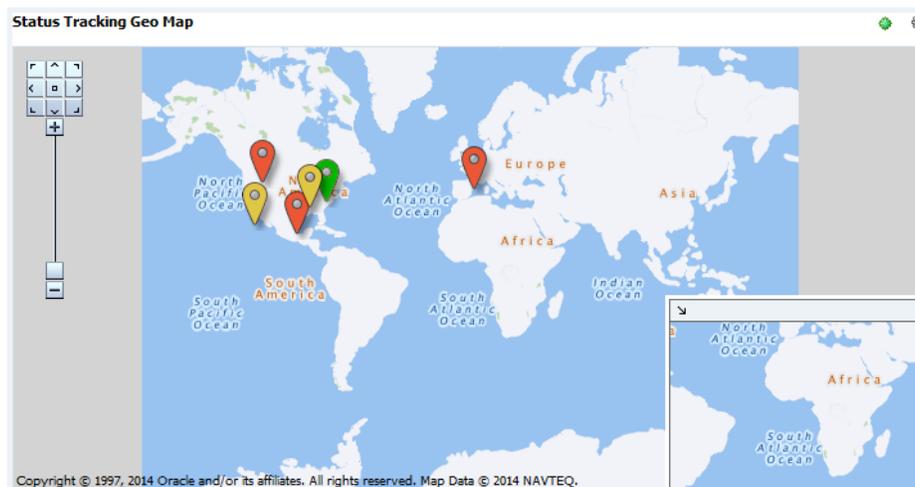
The preview changes to reflect changed property and runtime settings.

## Creating Geo Map Business Views

A Geo Map view displays values of a measure by location. The following Geo Map view types are available in Oracle BAM:

- A **Point Theme** represents static locations as points on a map. It displays data from a Flat SQL Query that must include address and country code fields or latitude and longitude fields, and one measure.
- A **Color Theme** represents geographic regions as colors on a map. It displays data from a Flat SQL Query that must include a location field and one measure.
- A **GPS Theme** represents moving locations tracked by GPS as points on a map. It displays data from a Flat SQL Query that must include latitude and longitude fields and one measure.

**Figure 8-16 Point Theme Geo Map Example**



Latitude and longitude values must be valid. Invalid values result in data points plotted at random locations.

You cannot switch from one Geo Map view type to another.

See [Creating a Flat SQL Query](#) for more information about flat queries.

**To create a Geo Map view:**

1. Go to the **Designer** page.
2. Click **Business Views** in the left panel navigator, or right-click **Business Views** and select the **Create** menu item.

The Business Views dialog opens.

3. Type a **Name** and optionally edit the **Display Name**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the view is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

4. Select **Geo Map** as the Category.
5. Select a thumbnail representing the desired view type.
6. Click **Create**.

A tab opens for the new view, with a visual preview of the map.

7. Select a query from the **Query** drop-down list. Only Flat SQL Queries are listed.
8. To use address and country code fields to define locations for a Point Theme Geo Map, check **Use Address**.

The country code field values must be ISO 3166 two- or three-character country names. The eLocation geo coder supports the following country names without extra data licenses: USA, AUT, DNK, ESP, FRA, GBR, SWE, ITA, NLD, BEL, DEU, PRI, CAN, LUX, CHE, NOR, FIN, LIE, PRT, and IRL.

The address field values can be in any format that is on one line. If there is no separate country code field, the address field must include the previously listed country codes.

Be sure to include both the city and the state. Omitting the state might result in the city being plotted in an incorrect location.

If you do not check **Use Address**, or if you are creating a GPS Theme Geo Map, latitude and longitude fields are used.

Latitude values are +90 through -90. Longitude values are +180 through -180. Use of N, S, E, and W is not supported.

See [Table 8-7](#) for more information about allowed location field names for a Color Theme Geo Map.

9. The next step depends on the previous one. Do one of the following:
  - Select data fields from the **Address** and **Country Code** drop-down lists.
  - Select data fields from the **Longitude** and **Latitude** drop-down lists.
10. Select from the **Map Data** drop-down list the measure data field for which the Geo Map will display the values.

The preview changes to reflect the selection.

11. To change the formatting of the view, click **Properties**. See these sections for details about formatting properties that apply to Geo Map views:
  - [General Properties Specific to Geo Map Business Views](#)
  - [Label, Text, Title, and Value Properties](#)
  - [Map Points Properties](#)
  - [Zoom Properties](#)

To save the properties you have edited, click **Apply**.

12. To add runtime interactions to the view, click **Runtime-Interaction**. See these sections for details about runtime interactions that apply to Geo Map views:
  - [Using Actions in Business Views](#)
  - [Using Active Data](#)

Active data is not supported for Color Theme Geo Maps.

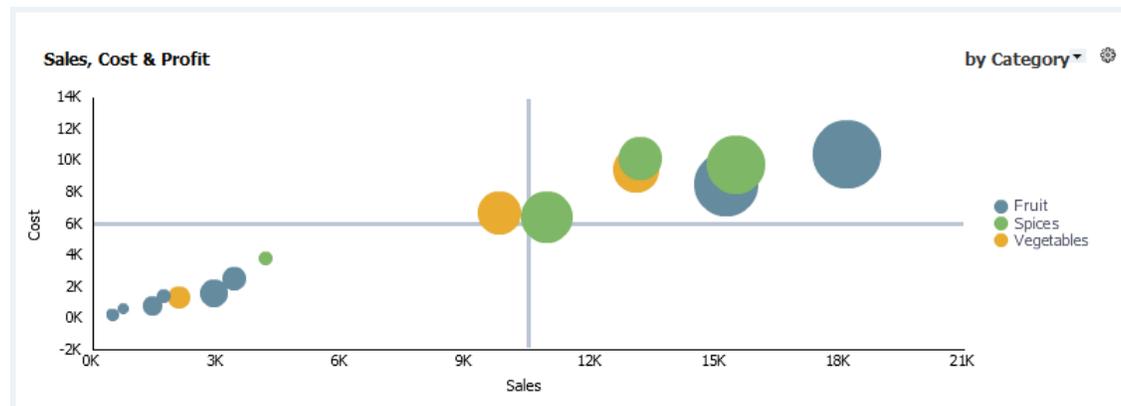
**13. Click Save.**

The preview changes to reflect changed property and runtime settings.

## Creating Scatter and Bubble Business Views

Bubble views are like Scatter views, except that X,Y coordinates appear as bubbles that vary in size. Except for the selection of a **Bubble Size** data field, the steps to create a Bubble view are the same as those to create a Scatter view.

**Figure 8-17 Bubble View Example**



A Scatter or Bubble view displays data from a Group SQL Query. See [Creating a Group SQL Query](#) for more information.

### To create a Scatter or Bubble view:

1. Go to the **Designer** page.
2. Click **Business Views** in the left panel navigator, or right-click **Business Views** and select the **Create** menu item.

The Business Views dialog opens.

3. Type a **Name** and optionally edit the **Display Name**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the view is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

4. Select a Category: **Scatter** or **Bubble**.
5. Select a thumbnail representing the desired view type.
6. Click **Create**.

A tab opens for the new view, with a visual preview using generic data.

7. Select a query from the **Query** drop-down list. Only Group SQL Queries are listed.

The preview changes to reflect the data fields fetched by the query.

8. Select a data field from the **X Axis** drop-down list.  
The preview changes to reflect the X-Axis selection.
9. Select a data field from the **Y Axis** drop-down list.  
The preview changes to reflect the Y-Axis selection.
10. For a Bubble view only, select a data field from the **Bubble Size** drop-down list.  
The preview changes to reflect the Bubble Size selection.
11. To change the formatting of the view, click **Properties**. See these sections for details about formatting properties that apply to Scatter and Bubble views:
  - [Common General Properties](#)
  - [Chart, Scatter, and Bubble General Properties](#)
  - [Label, Text, Title, and Value Properties](#)
  - [X-Axis Properties](#)
  - [Y-Axis Properties](#)
  - [Legend Properties](#)
  - [Target Properties](#)To save the properties you have edited, click **Apply**.

 **Note:**

For Scatter and Bubble graph views, legends may not display time units if the view is based on a time series query that has date-time values. This only applies to graph views and NOT chart views.

12. To add runtime interactions to the view, click **Runtime-Interaction**. See these sections for details about runtime interactions that apply to Scatter and Bubble views:
  - [Using Drilling](#)
  - [Using Actions in Business Views](#)
  - [Using Active Data](#)
13. Click **Save**.  
The preview changes to reflect changed property and runtime settings.

## Creating Treemap Business Views

In a Treemap view, hierarchical data groupings are arranged in rectangles within rectangles. For example, a hierarchy might consist of the country, state or province, and city. The count of sales might determine the size of the rectangles, and the sum of sales might determine their colors.

[Figure 8-18](#) shows a Treemap view from one of the preassembled dashboards in the Process Analytics project. See [Bottleneck Analysis of Open Processes Dashboard](#) for more information.

Figure 8-18 Treemap Example



A Treemap view displays data from a Tree Model Query. See [Creating a Tree Model Query](#) for more information.

#### To create a Treemap view:

1. Go to the **Designer** page.
2. Click **Business Views** in the left panel navigator, or right-click **Business Views** and select the **Create** menu item.

The Business Views dialog opens.

3. Type a **Name** and optionally edit the **Display Name**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the view is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

4. Select **Treemap** as the Category.
5. Select the Treemap thumbnail.
6. Click **Create**.

A tab opens for the new view, with a visual preview using generic data.

7. Select a query from the **Query** drop-down list. Only Tree Model Queries are listed.
8. Select a measure data field from the **Values** drop-down list. This field determines the size of each rectangle.

A measure that contains negative values cannot be used for **Values** in a Treemap.

9. Select a measure or dimension data field from the **Colors** drop-down list. This field determines the color of each rectangle.
10. To change the formatting of the view, click **Properties**. See these sections for details about formatting properties that apply to Treemap views:
  - [Label, Text, Title, and Value Properties](#)

- [Treemap Layout Properties](#)
- [Treemap Thresholds Properties](#)
- [Node Depth Properties](#)
- [Labels Properties](#)
- [Tooltip Labels Properties](#)

To save the properties you have edited, click **Apply**.

**11.** Click **Save**.

The preview changes to reflect the data fields fetched by the query, the **Values** and **Colors** selections, and changed property settings.

## Map Points Properties

Map Points properties are specific to Point Theme and GPS Theme Geo Maps. This tab enables you to set category columns and icons for various points of interest that are displayed on the Geo Map. You can also set custom icons and images for categories.

The **Default Map Point Icon** drop-down list enables you to select an icon or upload a custom image. The default value is **Ball\_Orange**.

The **Category Column** drop-down list enables you to select a category

The Map Points table has customizable columns that you can use to add the **Category**, **Min Value**, **Max Value**, and **Image**. The Category column allows you to specify a string value as the name for a category. Min and Max Value columns allow you to specify a numerical value. If you are specifying a location, you can either enter the latitudes and longitudes or enter an address. When selecting an image, you can either upload one from a local source or specify a remote URL.

## Zoom Properties

Zoom properties are specific to all Geo Map views. It provides you with the ability to set zoom levels and initial Geo Map positions using longitude and latitude values.

The **Latitude** and **Longitude** fields can contain any numeric value including negative values and decimals. The default value for both is 0.0. Once you set specific values and save the business view, the maps will load the specified coordinates on all subsequent startups. Any value with a special character, which is not a decimal is not allowed in both fields.

The **Zoom Level** slider allows you to set zoom levels from 0 to 20, with 20 being the maximum level till which you can zoom in.

The changes you make for zoom properties will only take effect once you **Save** the business view. Clicking **Apply** will not reflect your changes immediately.

## Formatting Business Views

Formatting includes tasks such as adding display titles, formatting text and colors, and changing value formats. Many formatting features apply to specific view types only. Because each of the view types has different characteristics and structures, formatting options change depending on the view type. KPI Watchlist views have no formatting properties.

To edit the formatting properties of a view, click **Properties** in the view tab.

Some properties require saving for the changes to reflect. To save the properties you have edited, do one of the following:

- For properties without the Save Required icon, click **Apply**.
- For properties with the Save Required icon, click **Apply**, click **Close**, and click **Save**.

This section contains the following topics:

- [General Properties](#)
- [Label, Text, Title, and Value Properties](#)
- [Axis Properties](#)
- [Legend Properties](#)
- [Target Properties](#)
- [Slice Properties](#)
- [Text Properties](#)
- [Layout Properties](#)
- [Editable Fields Properties](#)
- [Pivot Table Data Properties](#)
- [Pivot Table Label Properties](#)
- [Thresholds Properties](#)
- [Node Depth Properties](#)
- [Labels Properties](#)
- [Zoom Properties](#)
- [Map Points Properties](#)

## General Properties

General formatting properties are shared by all views of a particular type. Common general properties are shared by most view types.

To edit the general formatting properties of a view, click **Properties** in the view tab, then click **General**.

This section contains the following topics:

- [General Properties for Business Views](#)
- [Scatter and Bubble General Properties](#)
- [List General Properties](#)
- [Gauge General Properties](#)
- [General Properties Specific to Pivot Table Business Views](#)
- [General Properties Specific to Geo Map Business Views](#)

## General Properties for Business Views

The following table describes the general formatting properties.

Table 8-2 General Properties

Property	Description	Applies To
Predefined Style	Specifies the overall color scheme: Default, Blues, Greens, (Green, Yellow, Red), or Alta. Also lists custom color themes set using the ColorThemes data object.	All charts and graphs.
Hover Behavior	Specifies whether a mouse-over on a dataset will dim other datasets. The default value is <b>none</b> .	Area Chart, Bar Chart, Horizontal Bar Chart, Line Chart, Combo Chart, Bubble Chart.
Stack	Specifies whether stacking should be enabled. The default value is <b>off</b> .	Area Chart, Bar Chart, Horizontal Bar Chart, Line Chart.
Description	Specifies a text description of the view. Applies to all view types that have formatting properties.	Area, Bar, Horizontal Bar, Line, Bubble, Combo, Scatter Graphs and Charts. List Views.
Empty Text	Specifies the text displayed when a chart element or table cell is empty because no data exists. The default text is No Values. Applies to chart, Scatter, Bubble, and Table views.	Area, Bar, Horizontal Bar, Line, Pie, Combo, Scatter, and Bubble Graphs and Charts. List Views.
Display Border	Specifies whether a view has a border. The default is No. Applies to Area, Scatter, Bubble, List, and Gauge views, in both graph and chart views.	Area, Bar, Horizontal Bar, Line, Pie, Combo, Scatter, and Bubble Graphs and Charts. List Views.
Border Color	Specifies a border color if <b>Display Border</b> is set to Yes. Click the down arrow to display a palette. The default is white. Applies to chart, Scatter, Bubble, List, and Gauge views.	Area, Bar, Horizontal Bar, Line, Pie, Combo, Scatter, and Bubble Graphs and Charts. List Views.
Display Fill	Specifies whether the graph area of a view has a fill color. The default is No. Applies to chart, Scatter, Bubble, List, and Gauge views.	Area, Bar, Horizontal Bar, Line, Pie, Combo, Scatter, and Bubble Graphs and Charts. List Views.
Fill Color	Specifies a fill color if <b>Display Fill</b> is set to Yes. Click the down arrow to display a palette. The default is white. Applies to chart, Scatter, Bubble, List, and Gauge views.	Area, Bar, Horizontal Bar, Line, Pie, Combo, Scatter, and Bubble Graphs and Charts. List Views.
3D Effect	Converts the business view into a 3D visualization.	Area, Bar, Horizontal Bar, Line, Pie, Combo, Scatter, and Bubble Graphs. Pie Chart.

Other formatting properties are elaborated in the subsequent sections.

## Scatter and Bubble General Properties

Table 7–3 describes the Scatter and Bubble general formatting properties. Area Chart, Bar Chart, Horizontal Bar Chart, Line Chart, Pie Chart, Combo Chart, Scatter Chart, and Bubble Chart category views are collectively called *chart* views.

**Table 8-3 Scatter and Bubble General Properties**

Property	Description
Hover Behavior	Specifies whether a mouse-over on a dataset will dim other datasets. The default value is <b>none</b> .

## List General Properties

Table 8-4 describes the general formatting properties common to all Table views except Pivot Tables, which are List, Action List, and Collapsed List.

**Table 8-4 List General Properties**

Property	Description
Apply Same Fill Color	Specifies that all columns have the same fill color if checked. The default is unchecked. If checked, click the down arrow to display a palette. The default color is white.
Column Fill Color	Specifies the fill color of the named column. Displayed only if <b>Apply Same Fill Color</b> is unchecked. Click the down arrow to display a palette. The default color is white.

## Gauge General Properties

Gauge business views have some or all of the properties described in the following table depending on the gauge type.

Property	Description
Gauge Value Font	Specifies a font for the values used in the gauge. The default is Helvetica.
Font size	Specifies a font size with the option for increments and decrements. The default value is 50.
BIU	Specifies if the text should be in Bold, Italics, or Underline.
Font Color	Specifies a font color. The default is black.
Indicator Color	Specifies a color for the indicator. Used only in Circular and Horizontal Status Meter Gauges.
Indicator Border Color	Specifies a color for the border of the indicator. Used only in Circular and Horizontal Status Meter Gauges.
Threshold Display	Specifies where and how the threshold values are indicated. You can choose from <b>On Indicator</b> , <b>Current on Plot Area</b> , and <b>All on Plot Area</b> . <b>On Indicator</b> displays the current threshold along the indicator. <b>Current on Plot Area</b> displays only the current threshold along the indicator. <b>All on Plot Area</b> displays all thresholds along the indicator. The default is <b>All on Plot Area</b> .
Plot Area	Specifies if the plot area is On or Off. The default is On.
Scale	Specifies what scale to use for a value. You can choose from <b>Auto</b> , <b>None</b> , <b>Thousand</b> , <b>Million</b> , <b>Billion</b> , <b>Trillion</b> , and <b>Quadrillion</b> . The default is <b>Thousand</b> .

Property	Description
Format Field as (or Format Metric Value Field as for Gauge business views)	Specifies the data value formatting: None, HTML, Number, Currency, or Percent. HTML is only for text fields in *List views. Number, Currency, and Percent are only for numeric fields. The default is None, which removes all formatting information from the view. This property does not affect views in currently open dashboards in the BAM Viewer (Home page) and is activated after the view is saved.
No of Digits After Decimal	Specifies the number of decimal places if <b>Format Field as</b> is <b>Number</b> , <b>Currency</b> , or <b>Percent</b> . The default is 2.
Negative Number Format	Specifies the format for negative numbers if <b>Format Field as</b> is <b>Number</b> , <b>Currency</b> , or <b>Percent</b> : -1234.00, 1234.00, 1234.00-, or (1234.00). The default is -1234.00.
Use Digit Grouping Symbol	Specifies whether digits are grouped by thousands if <b>Format Field as</b> is <b>Number</b> . The default is Yes.
Currency	Specifies a currency text string if <b>Format Field as</b> is Currency. The default is blank.

 **Note:**

When the **On Indicator** option is selected from the **Threshold Display** drop-down, the colors defined on the Thresholds tab will be used and the Indicator Color on the General tab will be ignored.

## General Properties Specific to Pivot Table Business Views

[Table 8-5](#) describes the general formatting properties specific to Pivot Table views.

**Table 8-5 Pivot Table General Properties**

Property	Description
Pivot Enable	Enables pivoting of columns and rows if checked (the default).
Header Background Color	Specifies a background color for headers. Specify a hexadecimal or RGB color code or click the down arrow to display a palette. The default is white.
Data Background Color	Specifies a background color for data. Specify a hexadecimal or RGB color code or click the down arrow to display a palette. The default is white.

## General Properties Specific to Geo Map Business Views

All Geo Map views have a **Base Map** general property set to `WORLD_MAP`. All other Geo Map general properties are theme-specific.

There is only one additional general property for Geo Map views with a GPS Theme. **Point Graphics** specifies the image file used for points on the map. The default is `Ball_Orange`.

[Table 7-7](#) describes the general formatting properties specific to Geo Map views with a Point Theme.

**Table 8-6 Geo Map Point Theme General Properties**

Property	Description
Red Min	Specifies the low end of the value range for red points on the map. The default is 0.0.
Red Max	Specifies the high end of the value range for red points on the map. The default is 100.0.
Red Image	Specifies the image file used for red points on the map. The default is 3DCube_Red.
Orange Min	Specifies the low end of the value range for orange points on the map. The default is 100.0.
Orange Max	Specifies the high end of the value range for orange points on the map. The default is 200.0.
Orange Image	Specifies the image file used for orange points on the map. The default is 3DCube_Orange.
Green Min	Specifies the low end of the value range for green points on the map. The default is 200.0.
Green Max	Specifies the high end of the value range for green points on the map. The default is 300.0.
Green Image	Specifies the image file used for green points on the map. The default is 3DCube_Green.

[Table 8-7](#) describes the general formatting properties specific to Geo Map views with a Color Theme.

**Table 8-7 Geo Map Color Theme General Properties**

Property	Description
Theme Name	Specifies the theme for the colored regions: MAP_COUNTRIES_NAME (the default), MAP_CONTINENTS_NAME, MAP_COUNTIES_NAME, MAP_STATES_ABBREV, or MAP_STATES_NAME.

Table 8-7 (Cont.) Geo Map Color Theme General Properties

Property	Description
Theme Location Column	<p>Specifies the name of the data field for the regions. The values in this column must exactly match the values in the Flat SQL Query location field. The values in this column must also be unique for a given location, or the same data might be plotted in multiple locations.</p> <p><b>If Theme Name is</b> MAP_COUNTRIES_NAME, options are COUNTRY (the default), CAPITAL, CONTINENT, GEOMETRY, RID, or AREA.</p> <p><b>If Theme Name is</b> MAP_CONTINENTS_NAME or MAP_COUNTIES_NAME, options are CONTINENT or GEOMETRY.</p> <p><b>If Theme Name is</b> MAP_STATES_ABBREV or MAP_STATES_NAME, options are CARTO_ID, FEATURE_TYPE, NAME, ISO_COUNTRY_CODE, LANG_CODE, SQKM, DATA_SOURCE, GEOMETRY, STATE_ABRV, or POLYGON_NAME.</p>
Bucket Count	Specifies the number of color-coded ranges for data values.
Min Color	Specifies a color for the lowest bucket. Specify a hexadecimal color code or click the down arrow to display a palette. The default is red. Intermediate buckets have intermediate colors.
Max Color	Specifies a color for the highest bucket. Specify a hexadecimal color code or click the down arrow to display a palette. The default is green. Intermediate buckets have intermediate colors.

## Label, Text, Title, and Value Properties

Most views have a title, legend titles, axis labels, data labels, and other text elements. Different views have different text elements and text formatting options, but each option functions in the same way in all views where it is present.

To edit the text properties of a view, click **Properties** in the view tab, then click **Text**.

\*List views have their own text properties. See [Text Properties](#).

Treemap views have their own label properties. See [Labels Properties](#).

[Table 8-8](#) describes the common text formatting properties.

**Table 8-8 Common Text Formatting Properties**

Property	Description
Text	Specifies the text to be displayed. For text elements that are not editable, such as pie slice labels, no text field is provided.
Font	Select a font from the drop-down list. The default is Tahoma.
Font Size	Specifies the point size of the font. The default is 11.
B I U	Specifies whether text is bold, italic, underlined, or a combination. The default is plain text.
Text Color	Specifies the text color. Click the down arrow to display a palette. The default is black.
Align	Specifies the text alignment. Click the down arrow and select Left, Center, or Right. The default is Left.
Automatic Rotation	Specifies whether text is automatically rotated to fit: No Auto Rotation (the default), Auto Rotate 90 Degrees, or Auto Rotate 270 Degrees.
Text Direction	Specifies whether text is always rotated: No Rotation (the default), Rotate 90 Degrees, or Rotate 270 Degrees.
Label Text	Specifies the data field label text in chart, Scatter, or Bubble views. The default is the data field name.
Footnote Text	Specifies text to be shown below the chart or graph. It can be aligned as needed.

[Table 8-9](#) describes the common value formatting properties.

**Table 8-9 Common Value Formatting Properties**

Property	Description
Format Field as (or Format Metric Value Field as for Gauge business views)	Specifies the data value formatting: None, HTML, Number, Currency, or Percent. HTML is only for text fields in *List views. Number, Currency, and Percent are only for numeric fields. The default is None, which removes all formatting information from the view. This property does not affect views in currently open dashboards in the BAM Viewer (Home page) and is activated after the view is saved.
No of Digits After Decimal	Specifies the number of decimal places if <b>Format Field as</b> is Number, Currency, or Percent. The default is 2.
Negative Number Format	Specifies the format for negative numbers if <b>Format Field as</b> is Number, Currency, or Percent: -1234.00, 1234.00, 1234.00-, or (1234.00). The default is -1234.00.

**Table 8-9 (Cont.) Common Value Formatting Properties**

Property	Description
Use Digit Grouping Symbol	Specifies whether digits are grouped by thousands if <b>Format Field as</b> is Number. The default is Yes.
Currency	Specifies a currency text string if <b>Format Field as</b> is Currency. The default is blank.

## Axis Properties

You can format any axis of any of the following view types:

- Area
- Bar
- Horizontal Bar
- Line
- Combo
- Dial
- Statusmeter
- Scatter
- Bubble

In a Horizontal Bar chart, the X-axis is vertical and the Y-axis is horizontal, unlike the standard convention. This is necessary because you can switch from one chart view to another and retain the axis properties.

As an alternative for a Y-axis, you can accept the default automatic axis formatting, which adjusts the axis based on active data.

To edit the axis properties of a view, click **Properties** in the view tab, then click **Axis**, **X-Axis**, **Y-Axis**, **Y1-Axis**, or **Y2-Axis**, depending on the view type.

This section contains the following topics:

- [Axis Properties Specific to Dial and Statusmeter Business Views](#)
- [X-Axis Properties](#)
- [Y-Axis Properties](#)

## Axis Properties Specific to Dial and Statusmeter Business Views

[Table 8-10](#) describes the axis formatting properties for Dial and Statusmeter view types.

**Table 8-10 Dial and Statusmeter Axis Formatting Properties**

Property	Description
Display Major Tick Marks	Specifies whether to display major ticks. The default is Yes.
Tick Interval	Specifies the interval between ticks. The default is 0.0.

**Table 8-10 (Cont.) Dial and Statusmeter Axis Formatting Properties**

Property	Description
Tick Color	Specifies the tick color. Click the down arrow to display a palette. The default is black.
Display Minor Tick Marks	Specifies whether to display minor ticks. The default is No.
Tick Label	Formats the tick label text. See <a href="#">Label_ Text_ Title_ and Value Properties</a> for more information.
Display Label	Specifies how the axis is labeled: (Minimum, maximum, and increments) (the default), Increments, Minimum and maximum values, Actual metric value, No tick labels, or Threshold values.
Angle Extent	Specifies the number of degrees that a Dial view spans. The default is 180.

## X-Axis Properties

[Table 8-11](#) describes the X-axis formatting properties. Not all views with an X-axis have all these properties.

**Table 8-11 X-Axis Formatting Properties**

Property	Description
Title	Specifies a title for the axis. See <a href="#">Label_ Text_ Title_ and Value Properties</a> for more information.
Axis Width	Specifies the width of the axis in pixels. The default is 1.
Axis Color	Specifies the axis color. Click the down arrow to display a palette. The default is black.
Major Tick Width	Specifies the width of the major ticks in pixels. The default is 1.
Major Tick Color	Specifies the major tick color. Click the down arrow to display a palette. The default is black.
Major Tick Line Style	Specifies the major tick line style (dash pattern) for major ticks. The default is solid (no dashes).
Major Tick Style	Specifies the major tick style: None (the default), Auto, Extended, Grid, Inside Axis, Span Across Axis, or Outside Axis.
Tick Label	Formats the tick label text. See <a href="#">Label_ Text_ Title_ and Value Properties</a> for more information.
Select Groups to format for horizontal fields	Specifies the dimensions to which the tick formatting applies.

## Y-Axis Properties

[Table 8-12](#) describes the Y-axis formatting properties. Not all views with a Y-axis have all these properties.

**Table 8-12 Y-Axis Formatting Properties**

Property	Description
Title	Specifies a title for the axis. See <a href="#">Label_ Text_ Title_ and Value Properties</a> for more information.
Axis Width	Specifies the width of the axis in pixels. The default is 1.
Axis Color	Specifies the axis color. Click the down arrow to display a palette. The default is black.
Auto Axis	Adjusts the axis based on the data if checked. The default is checked.
Axis Min	Specifies a minimum value for the axis if <b>Auto Axis</b> is not checked. The default is 0, for no minimum.
Axis Max	Specifies a maximum value for the axis if <b>Auto Axis</b> is not checked. The default is 0, for no maximum.
Axis Increment Minor	Specifies a minor increment for the axis if <b>Auto Axis</b> is not checked. The default is 0, for no set increment.
Axis Increment Major	Specifies a major increment for the axis if <b>Auto Axis</b> is not checked. The default is 0, for no set increment.
Major Tick Width	Specifies the width of the major ticks in pixels. The default is 1.
Major Tick Color	Specifies the major tick color. Click the down arrow to display a palette. The default is black.
Major Tick Line Style	Specifies the major tick line style (dash pattern) for major ticks. The default is solid (no dashes).
Major Tick Style	Specifies the major tick style: None (the default), Auto, Extended, Grid, Inside Axis, Span Across Axis, or Outside Axis.
Minor Tick Width	Specifies the width of the minor ticks in pixels. The default is 1.
Minor Tick Color	Specifies the minor tick color. Click the down arrow to display a palette. The default is black.
Minor Tick Line Style	Specifies the minor tick line style (dash pattern) for major ticks. The default is solid (no dashes).
Minor Tick Style	Specifies the minor tick style: None (the default), Auto, Extended, Grid, Inside Axis, Span Across Axis, or Outside Axis.
Tick Label	Formats the tick label text. See <a href="#">Label_ Text_ Title_ and Value Properties</a> for more information.
Format Field as	Specifies the Y-axis data field to which to apply value formatting. See <a href="#">Table 8-9</a> for more information.

## Legend Properties

You can format the legend of any of the following view types:

- Area
- Bar
- Horizontal Bar
- Line
- Pie
- Combo

- Gauge
- Scatter
- Bubble

To edit the legend properties of a view, click **Properties** in the view tab, then click **Legend**.

[Table 8-13](#) describes the legend formatting properties.

**Table 8-13 Legend Formatting Properties**

Property	Description
Display Legend	Specifies whether the legend is displayed or hidden.
Display Border	Specifies whether a legend has a border. The default is No.
Border Color	Specifies a border color if <b>Display Border</b> is set to Yes. Click the down arrow to display a palette. The default is white.
Display Fill	Specifies whether a legend has a fill color. The default is No.
Fill Color	Specifies a fill color if <b>Display Fill</b> is set to Yes. Click the down arrow to display a palette. The default is white.
Legend Position	Specifies whether the legend is on the left, right, bottom, or top side of the view. The default position is on the right side.
Graph Type	Specifies the graph type of each data field for a Combo view: Bar Graph, Line Graph, Area Graph, Curve Line, Centered Stepped Line, Stepped Line, or Default. By default, each data field is assigned a type in the above order.

## Target Properties

A target line makes it easy to compare a process you are tracking to a goal. You can add a target line to any of the following view types:

- Area
- Bar
- Horizontal Bar
- Line
- Combo
- Scatter
- Bubble

For a Horizontal Bar type view, the target line is vertical. For the other types, the target line is horizontal.

To edit the target properties of a view, click **Properties** in the view tab, then click **Target**.

[Table 8-14](#) describes the target line formatting properties.

**Table 8-14 Target Line Formatting Properties**

Property	Description
Display Line	Specifies whether the target line is displayed. The default is No.
Value	Specifies the value of the target line on the Y-axis. The default is zero.

**Table 8-14 (Cont.) Target Line Formatting Properties**

Property	Description
Text	Specifies a text label for the target line.
Line Width	Specifies the width of the target line in pixels. The default is 1.
Line Color	Specifies the target line color. Click the down arrow to display a palette. The default is red.
Line Style	Specifies the target line style (dash pattern). The default is solid (no dashes).
Show in Legend	Specifies whether the target line and its text label are included in the legend. The default is No.

## Animation Properties

You can add animation to any of the following view types:

- Area
- Bar
- Horizontal Bar
- Line
- Pie
- Combo
- Gauge
- Scatter
- Bubble

To edit the animation properties of a view, click **Properties** in the view tab, then click **Animation**.

This section contains the following topics:

- [Chart Animation Properties](#)
- [Gauge Animation Properties](#)

## Chart Animation Properties

[Table 8-15](#) describes the animation formatting properties for chart, Scatter, and Bubble views.

**Table 8-15 Chart Animation Formatting Properties**

Property	Description
Series Rollover Effect	Specifies whether rolling the mouse over a legend item or a series in the view highlights the series: None or Highlight. The default is Highlight.
Animation on Transition	Specifies whether animation occurs when a view is rendered: None or Auto. The default is None.

## Gauge Animation Properties

[Table 8-15](#) describes the Gauge animation formatting properties.

**Table 8-16 Gauge Animation Formatting Properties**

Property	Description
Animation on Data Change	Specifies whether animation occurs when data changes: Active (default), None, or Auto. Active animates only Active Data changes. Auto animates all data changes.
Animation Display Duration: Time	Specifies the animation display duration in milliseconds: 1000 to 1500. The default is 1000.
Animation on Transition	Specifies whether animation occurs when a view is rendered: None or Auto. The default is None.

## Slice Properties

Slice properties apply only to Pie views.

To edit the Pie slice properties of a view, click **Properties** in the view tab, then click **Slice**.

[Table 8-17](#) describes the Pie slice formatting properties.

**Table 8-17 Pie Slice Formatting Properties**

Property	Description
Slice Label Text Format	Formats the slice label text. See <a href="#">Label, Text, Title, and Value Properties</a> for more information.
Slice Behavior	Specifies whether you can explode slices when viewing the Pie view in the BAM Viewer (Home page). Explode moves one slice at a time, Explode All moves all slices out, and None disables exploding slices. The default is None.
Slice Label Position	Specifies where slice labels are positioned: Outside with Feeler, No Labels, Outside without Feeler, Inside, or Outside with Feeler (if needed). The default is Outside with Feeler.
Slice Label	Specifies the slice label text: Percentage, Slice Name, Value, or Slice Name and Value. The default is Percentage.

 **Note:**

The DVT—based pie charts do not support formatting of slice labels. As a result, you might see a large number of decimal points when the option **Slice Name and Value** is used.

## Text Properties

Text properties described here apply only to \*List views.

For descriptions of text formatting properties for other views, see [Label, Text, Title, and Value Properties](#)..

To edit the text properties of a List view, click **Properties** in the view tab, then click **Text**.

[Table 1](#) describes the text formatting properties.

**Table 8-18 Text Formatting Properties**

Property	Description
Table Title	Formats the table title. See <a href="#">Label, Text, Title, and Value Properties</a> for more information.
Table Text	Formats the table text. See <a href="#">Label, Text, Title, and Value Properties</a> for more information.
Column Header Visible	Specifies that column headers are visible if checked. The default is checked.
Column Header Display Text	Specifies the column header text. The default is the data field name.
Column Header Fill Color	Specifies the column header fill color. Click the down arrow to display a palette. The default is white.
Select table columns to format	Specifies the column to which to apply value formatting. See <a href="#">Table 8-9</a> for more information.

## Layout Properties

Layout properties apply to \*List and Treemap views.

To edit the layout properties of a view, click **Properties** in the view tab, then click **Layout**.

This section contains the following topics:

- [List Layout Properties](#)
- [Treemap Layout Properties](#)

## List Layout Properties

[Table 8-19](#) describes the \*List layout formatting properties. **Row Selection** does not apply to Collapsed List views.

**Table 8-19 List Layout Formatting Properties**

Property	Description
Horizontal Grid	Displays horizontal grid lines if checked.
Vertical Grid	Displays vertical grid lines if checked.
Row Banding Interval	Specifies the number of unbanded rows between banded rows. The default is 1.
Column Banding Interval	Specifies the number of unbanded columns between banded columns. The default is 0, for no column banding.
Resizing	Enables column resizing if checked.
Reordering	Enables reordering of columns if checked.
Row Selection	Specifies how many rows can be selected: none, single, or multiple. The default is single.
Text Wrapping	Enables text wrapping of data if checked. Enabling text wrapping shows the complete text in the cell wrapped in multiple lines. When disabled, the text is truncated with ellipsis. The complete text can be seen in the tool tips.

**Table 8-19 (Cont.) List Layout Formatting Properties**

Property	Description
Column Stretching	Specifies which columns can be stretched when the view is resized: none, multiple, or last. The default is none.
Column Visibility	Specifies which columns are displayed. Check a column name to display it. By default, all columns are displayed.
Apply	Click Apply to apply the changes done.
Close	Click Close to close the window.

## Treemap Layout Properties

**Layout Options** determines how the rectangle hierarchy in a treemap is arranged:

- **Squarified** makes all the rectangles at all levels as square as possible. This is the default.
- **Slice and Dice Horizontal** orders rectangles alphanumerically by node label from left to right.
- **Slice and Dice Vertical** orders rectangles alphanumerically by node label from top to bottom.

## Editable Fields Properties

Editable field properties apply only to Action List views.

To edit the editable field properties of a view, click **Properties** in the view tab, then click **Editable Fields**.

**Editable Fields** determines which columns have editable data values. Check a column name to make its data values editable. By default, no columns have editable data values.

Editable fields are only useful when combined with actions. See [Using Actions in Business Views](#) for more information.



### Note:

If the number of editable fields multiplied by the number of rows in your action list is over 10000, you will get an error when you use this view in a dashboard.

## Pivot Table Data Properties

Data properties apply only to Pivot Table views.

To edit the data properties of a view, click **Properties** in the view tab, then click **Data**.

[Table 8-20](#) describes the data formatting properties.

**Table 8-20 Data Formatting Properties**

Property	Description
Sorting	Specifies how label text is sorted: Ascending or Descending. The default is Ascending.
Aggregate Position	Specifies where additional aggregate values are positioned: None, After, or Before. The default is None.
Aggregate Function	Specifies a category aggregate function: Sum, Average, Count, Maximum, or Minimum. The default is Sum.

## Pivot Table Label Properties

Label properties apply to Pivot table views.

To edit the Label properties of a Pivot table view, click **Properties** in the view tab, then click **Labels**.

**Table 8-21 Pivot Table Label Properties**

Property	Description
Label	Specifies the data field label in a Pivot table view. The default is the data field name.
Label Text	Specifies the data field label text in a Pivot table view.
Column Width (pxl)	Specifies the column width in pixels.
Apply	Click this button to apply the changes.
Close	Click this button to close the dialog.

## Thresholds Properties

Threshold properties apply only to Collapsed List, Gauge, and Treemap views.

To edit the threshold properties of a view, click **Properties** in the view tab, then click **Thresholds**.

This section contains the following topics:

- [Collapsed List Thresholds Properties](#)
- [Gauge Thresholds Properties](#)
- [Treemap Thresholds Properties](#)

## Collapsed List Thresholds Properties

[Table 8-22](#) describes the Collapsed List threshold formatting properties.

**Table 8-22 Collapsed List Thresholds Formatting Properties**

Property	Description
Data Field	Specifies data field on which color-coded threshold values are based. This can be a measure field such as Sales or a dimension field such as Product.

**Table 8-22 (Cont.) Collapsed List Thresholds Formatting Properties**

Property	Description
Use and Format BG Color	Specifies thresholds for cell background colors if checked. The default is checked.
Use and Format Graphics	Specifies thresholds for cell status graphics if checked. The default is checked.
Hide Column Data	Lets you choose whether to render column data with the image, or just the image when you use images in thresholds. The default is unchecked.
Threshold Values	Specifies the colon-separated low and high thresholds of a data value range, for example 100:200. Click the <b>Add</b> icon to add a new range.
Threshold Color	Specifies the cell background color for a data value range. Click the down arrow to display a palette.
Threshold Image	Specifies the cell status graphic for a data value range. Click <b>Select Image</b> to display a menu of choices. Select <b>Upload Image</b> from this menu to add your own graphic. Type a path to a graphic file or click <b>Browse</b> to open a file chooser. Then click <b>OK</b> .
Overflow Color	Specifies the cell background color for values above the highest threshold. Click the down arrow to display a palette.
Underflow Color	Specifies the cell background color for values below the lowest threshold. Click the down arrow to display a palette.
Image Position	Specifies the position of cell status graphics: Left or Right. The default is Left.

## Gauge Thresholds Properties

[Table 8-23](#) describes the Gauge threshold formatting properties.

**Table 8-23 Gauge Thresholds Formatting Properties**

Property	Description
Minimum Value	Specifies the minimum value for the low value range and for display on a Dial or Statusmeter axis. The default is 0.0.
Maximum Value	Specifies the maximum value for the high value range and for display on a Dial or Statusmeter axis. The default is 100.0.
Threshold 1 Value	Specifies the maximum value for the low value range. The default is 33.33.
Threshold 1 Color	Specifies the low value color. Click the down arrow to display a palette. The default is green.
Threshold 2 Value	Specifies the maximum value for the medium value range. The default is 66.66.
Threshold 2 Color	Specifies the medium value color. Click the down arrow to display a palette. The default is yellow.
Threshold 3 Value	Specifies the maximum value for the high value range. This is the same as the <b>Maximum Value</b> and is read-only.
Threshold 3 Color	Specifies the high value color. Click the down arrow to display a palette. The default is red.

## Treemap Thresholds Properties

[Table 8-24](#) describes the Treemap threshold formatting properties.

**Table 8-24 Treemap Thresholds Formatting Properties**

Property	Description
Color	Specifies data field on which color-coded threshold values are based. This can be a measure field such as Sales or a dimension field such as Product.
Hierarchy Level	Specifies the grouping data field in the hierarchy having the rectangles to which threshold colors are applied. For example, if the hierarchy consists of country, state or province, and city, you can apply color coding to only one of those levels.
Linear Thresholds	Specifies that thresholds are linear and use light to dark shades of the same color.
Max Value	Enter the maximum value for the data label between 1 to 100 for a <b>Linear Threshold</b> .
Min Value	Enter the minimum value for the data label between 1 to 100 for a <b>Linear Threshold</b> .
Buckets	Specifies the number of data partitions between the <b>Max Value</b> and the <b>Min Value</b> . For example, if <b>Max Value</b> is 100, <b>Min Value</b> is 1, and <b>Buckets</b> is 5, then 5 data partitions are created as follows: 1-20, 21-40, 41-60, 61-80, 81-100.
Base Color	Specifies the color for values within a <b>Linear Threshold</b> . Click the down arrow to display a palette.
Custom Thresholds	Specifies that thresholds are not linear and can use unrelated colors.
Threshold Values	Specifies the comma-separated low and high thresholds of a data value range for <b>Custom Thresholds</b> . Click the <b>Add</b> icon to add a new range.
Threshold Color	Specifies the color of a data value range for <b>Custom Thresholds</b> . Click the down arrow to display a palette.
Overflow Color	Specifies the color for values above the highest threshold. Click the down arrow to display a palette.
Underflow Color	Specifies the color for values below the lowest threshold. Click the down arrow to display a palette.
No Value	Specifies the color for a rectangle corresponding to a group with no data values. Click the down arrow to display a palette.
Reversed	This check box reverses the order of your linear thresholds properties. For example, if you set a base color #C0EFFF with min=0, max=7, and buckets=7, you get threshold colors where #C0EFFF is near 0, and the darker colors are near 7. Using the <b>Reversed</b> check box would reverse that order, so #C0EFFF would appear when the value is near 7, and lightest when near 0.

**Note:**

Oracle recommends that you always set Treemaps thresholds before viewing dashboards to show color gradations and precise legend values.

## Node Depth Properties

Node depth properties apply only to Treemap views.

To edit the node depth properties of a view, click **Properties** in the view tab, then click **Node Depth**.

**Depth of nodes during initial rendering** determines how many levels of the hierarchy are initially displayed. The default is 2.

## Labels Properties

Label properties described here apply only to Treemap views.

For descriptions of label formatting properties for other views, see [Label, Text, Title, and Value Properties](#).

To edit the label properties of a Treemap view, click **Properties** in the view tab, then click **Labels**.

[Table 8-25](#) describes the Treemap label formatting properties.

**Table 8-25 Labels Formatting Properties**

Property	Description
Leaf Display	Specifies the type of label display for the lowest level hierarchical groupings: Node or Off. The default is Node, which places each label within its rectangle.
Group Display	Specifies the type of label display for all except the lowest level hierarchical groupings: Header, Node, or Off. The default is Header, which places each label above the top left corner of its rectangle.
Horizontal Alignment	Specifies the horizontal alignment for Node labels: Center, Start, or End. The default is Center.
Vertical Alignment	Specifies the vertical alignment for Node labels: Center, Top, or Bottom. The default is Center.
Data Label Position	Specifies the data label position on the DVT charts. The default value is None.

## Tooltip Labels Properties

Tooltip Labels properties are specific to Treemap views. You can use this property to configure tooltip text for treemaps.

Tooltip Label properties described here apply only to Treemap views.

For descriptions of label formatting properties for other views, see [Label, Text, Title, and Value Properties](#).

To edit the Tooltip label properties of a Treemap view, click **Properties** in the view tab, then click **Tooltip Labels**.

### Configuring Tooltip Labels

The two available tooltip labels and their respective default label texts are as follows.

Label	Label Text
COUNTX	# of Processes
AVGDAYX_UNTIL_DUE_DATE	Avg Days Until Due Date

You can edit label texts for tooltips to suit your project needs.



**Note:**

These changes work only when you click **Save** and not when you click **Apply**.

## Runtime Interactions with Business Views

This section describes how you can enable end users to change views while viewing them using the BAM Viewer (the Home page), by drilling, performing actions, or making data active.



**Note:**

Drilling and other runtime interactions are only supported in View mode, and not in the Designer mode.

This section contains the following topics:

- [Using Drilling](#)
- [Using Actions in Business Views](#)
- [Using Active Data](#)

### Using Drilling

*Drilling* enables end users using the BAM Viewer (the Home page) to select an item in a series and view data at a more detailed level. You, the designer, configure drilling in the Drilling properties of the view.

You cannot drill in views while you edit them. In the BAM Viewer, active data stops during drilling and then restarts.

This section contains the following topics:

- [Drilling Down](#)
- [Drilling Up](#)
- [Drilling Through](#)
- [Drilling Across](#)
- [Enabling Drilling Down](#)
- [Enabling Drilling Through to Details](#)
- [Configuring Drill-Across Targets](#)

### Drilling Down

Drilling down means displaying data one level down the drill path in the same view. Drilling down is an option when all of the following are true:

- Drill-down is enabled.  
It is disabled by default. See [Enabling Drilling Down](#).

- Data in the view is aggregated.
- The underlying data object has a hierarchy.
- The next level down is not the lowest level.

If the next level down is the lowest level, it is only possible to drill through to the actual data rows. In some cases, drill-through is not an option either because it is disabled or unavailable in the view type. See [Drilling Through](#).

The drilling areas include:

- Points in Area, Line, Combo, Scatter, or Bubble views
- Bars in Bar, Horizontal Bar, or Combo views
- Slices in Pie views
- Nodes in Treemap views
- Rows in Collapsed List views

To drill down, right-click the point, bar, slice, node, or row and select the drill level from the menu.

For views like area or line, two data points are needed to plot a graph. When drilling down, if two points are not available then an error message is displayed. For passive view, you can clear the filters applied using runtime filter but for active views you do not have such option. You need to go to MDS browser to manually clear the applied filters. On the views for which drilling is enabled, you can return to the original state by clicking the home icon in the toolbar.

## Drilling Up

*Drilling up* means displaying data one level up the drill path. You can only drill up if you have previously drilled down.

For views like area or line, two data points are needed to plot a graph. When drilling down, if two points are not available then an error message is displayed. For passive view, you can clear the filters applied using runtime filter but for active views you do not have such option. You need to go to MDS browser to manually clear the applied filters. On the views for which drilling is enabled, you can return to the original state by clicking the home icon in the toolbar.

### To drill up:

- Right-click the point, bar, slice, node, or row and select the drill level from the menu.
- Right-click the point, bar, slice, node, or row and select **Home** to go to the initial drill level of the view.

#### Note:

If you drill down in an Area, Line, or Combo view to a level at which only one data point is included, no line is drawn, and you cannot drill up. To restore the view, remove the filter that the drilling applies. See [Filtering Runtime Data in a View](#) for more information.

## Drilling Through

*Drilling through* means displaying the actual data rows at the lowest level of the drill path in a List view. Drilling through is an option when all of the following are true:

- Drill-through is enabled.  
It is enabled by default. See [Enabling Drilling Through to Details](#).
- Data in the view is aggregated.
- If the underlying data object has a hierarchy, the next level down is the lowest level.

For drill-through, the underlying data object need not have a hierarchy defined.

Because drilling down is disabled by default and drilling through is enabled by default, drilling through is the default drilling action.

Treemaps allow drilling down but not drilling through.

### To drill through:

- Right-click the point, bar, slice, or row and select **Show Details** from the menu.

## Drilling Across

Drilling across means opening a target dashboard or URL in a new browser window.

### To drill across:

- Right-click the view and select the target name.

See [Configuring Drill-Across Targets](#) for more information.

## Enabling Drilling Down

You can configure a Drill Hierarchy for drill-down in the following view types:

- Area
- Bar
- Horizontal Bar
- Line
- Pie
- Combo
- Collapsed List
- Scatter
- Bubble

The underlying data object must have a hierarchy defined. There are two types of hierarchies: normal and datetime-based.

Treemaps have no drill hierarchy configuration because the hierarchy in the underlying Tree Model Query is used for drilling down.

**To select a drilling hierarchy:**

1. From the view tab, click **Runtime-Interaction**.
2. Click **Drilling**.
3. Select the **Drill Hierarchy** tab.
4. Check **Enable Drilling**.
5. Click the **Browse** icon to the right of the **Hierarchy** setting.  
The Hierarchy Chooser dialog appears, listing normal and datetime-based hierarchies.
6. Select the **Hierarchy** to use during drilling.  
If no items appear in the list, the data object does not have any hierarchies defined. For information about configuring hierarchies in data objects, see [Adding Hierarchies](#).
7. Click **OK**.  
The Hierarchy Chooser dialog closes, and hierarchy information appears in the properties window.
8. Click **Apply**.

## Enabling Drilling Through to Details

You can drill through from aggregated to detailed data (shown in a List) in the following view types:

- Area
- Bar
- Horizontal Bar
- Line
- Pie
- Combo
- Collapsed List
- Gauge
- Scatter
- Bubble

The underlying data object need not have a hierarchy defined.

Not all views that allow drilling through to details allow hierarchical drilling. For example, Gauge views allow only drill-through.

Treemaps allow drilling down but not drilling through.

 **Note:**

Drilling through to details for gauge views is enabled through left-clicks, as opposed to right-click drill to details for other view types. Drilling through to details also does not apply to Circular and Horizontal Status Meter Gauges.

**To enable drilling through to details:**

1. From the view tab, click **Runtime-Interaction**.
2. Click **Drilling**.
3. Select the **Drill to Detail** tab.
4. Select **Enable drill through to detail**.
5. Select a **Destination Type**:
  - **Replace current view** replaces the space in the dashboard occupied by the current view with the target view.
6. Select **List** as the **View Type**.
7. Click the **Browse** icon to the right of the **Table Display Fields** setting.  
The Table Display Fields Chooser dialog appears.
8. Move the desired data fields from the **Available Fields** list to the **Selected Fields** list.
9. Click **OK**.  
The Table Display Fields Chooser dialog closes, and the selected fields are shown in a table.
10. Click **Apply**.

## Configuring Drill-Across Targets

You can configure drill-across targets in the following view types:

- Area
- Bar
- Horizontal Bar
- Line
- Pie
- Combo
- List
- Action List
- Collapsed List
- Scatter
- Bubble

You open a target dashboard or URL in a new browser window. You can create many drill-across targets for a view.

Drilling across cannot be done from Gauge, Pivot Table, KPI Watchlist, Treemap, or Geo Map views.

**To configure a drill-across target:**

1. From the view tab, click **Runtime-Interaction**.
2. Click **Drilling**.
3. Select the **Drill Across** tab.

4. Click the **Add** icon.  
The Create Drill Path dialog opens.
5. Type a **Drill Path Name**.  
This is the name that appears on the menu from which the end user selects a drill-across target.
6. Select a **Destination Type**:
  - **Launch a new browser window** opens a new browser window in which to display the target view.
  - **Replace the dashboard** replaces the dashboard to display the target view.
7. Select a **Destination**:
  - **Destination dashboard** displays the specified dashboard.
  - **Destination Web URL** displays the specified URL.
8. Select a **Dashboard** or type a **URL**.  
If you opt to type a URL, you can optionally select the **Set as relative URL** option that enables you to provide the relative URL instead of a complete URL. The hostname, port number, and context path will be same as that of the running Oracle BAM application.
9. Optionally, select **Set as default drill action** to enable left-clicks on graph elements.
10. Optionally, map fields in the current view to parameters in the target dashboard, by clicking the **+** sign.  
There are three types of parameter mapping:
  - **Value from Business View**: If the target dashboard uses parameters, you can pass values from the fields in the current view to the parameters in the target dashboard. The point, bar, slice, node, or row the end user right-clicks when choosing the target determines the specific value to be passed. For example, right-clicking a bar in a Bar view passes the group value represented in that bar to a parameter in the target dashboard.
  - **Constant Value**: You can pass a constant value to an external parameter.
  - **Parameter Value**: In the source dashboard, if parameters are used inside the source view's query, then that value can be passed to the external parameter.
11. Click **Save**.  
The Create Drill Path dialog closes, and the drill path is shown in the table.
12. Click **Apply**.

## Using Actions in Business Views

This section describes how to add Action Buttons to views that you create in Oracle BAM. Action Buttons allow users to refresh the view, add or change data, and ask for confirmation of these actions.

Action Buttons can appear in any view except a KPI Watchlist.

Alerts can also perform actions. See [Alert Actions](#) for more information.

**To create an Action Button:**

1. From the view tab, click **Runtime-Interaction**.

2. Click **Actions**.
3. Click the **Add** icon.

A new action button appears in the Action Button List table with a default name of `New Action#`, and an Actions list appears.
4. Optionally edit the **Name** and add a **Description** for the Action Button.

The text you enter in the Name field is displayed as the button text in the view.
5. Optionally select **Commit All** if you need to ensure that all data actions configured in this Action Button take place in a single transaction.
6. Click the arrow immediately to the left of **Actions** to display the Actions list.
7. Click the **Add** icon on the right side of the Actions list to define an action for the selected Action Button.

For all views, you have these options:

- **Open Dashboard** opens a specified dashboard in a new browser window. Click the **Search** icon, select a dashboard from the list, and click **OK**.

If the dashboard has parameters, [Table 8-26](#) describes the values you can pass.
- **Open URL** opens a specified URL in a new browser window. Click the **Add** icon for Open URL Action.

[Table 8-26](#) describes the values you can use to construct URLs. A URL is constructed by concatenating each of the values in the order in which you specify them.
- **Refresh View** redisplay the view with the latest data and properties.
- **Show Confirmation Message** displays a confirmation message dialog before performing any of the other actions configured for this Action Button. Type the text of the message.

For Action List views, you have these additional options:

- **Insert** inserts data. Click the magnifying glass and choose a data object. For each data field in the data object, choose one of the values described in [Table 8-26](#).
- **Update** updates data. For each data field in the Action List, choose one of the values described in [Table 8-26](#). You can use this action to update a row in another DO, using the data from the selected row, constants, or parameters. This will require DO selection and column mapping.
- **Save** saves all changes made to the Action List. Use this action when you have editable fields and want to save changes. These changes are applied to the same DO and columns, which means no mapping is required.
- **Delete** deletes selected rows from the Action List.
- **Custom** action allows you to add any custom action to the business view. To add custom action:
  - a. On the Business View, select the **Runtime-Interaction** option.
  - b. Click the **Actions** button. The Action Button List dialog opens.
  - c. Click **Add** icon in Action Button List.
  - d. Provide a **Name** and **Description** for the custom action.
  - e. Select the **Commit All** checklist, if you want to ensure that all data actions configured in this Action Button take place in a single transaction.

- f. Click **Actions** and **Action Add** drop button and select the custom action you added.
- g. Provide the following credentials:
  - Library path - for example, `/scratch/myfolder/BusinessViewJar.jar`
  - Class Name - for example, `oracle.beam.server.customAlert.BusinessAction`
  - Argument - provide any additional statement.
- h. Click Apply and Close the dialog.

Multiple actions for an Action Button occur in the order in which you create them.

 **Note:**

The Insert, Update, Save, and Delete actions do not work with external data objects.

- 8. Click **Apply**.

[Table 8-26](#) describes the values actions can pass in more detail.

**Table 8-26 Values for Actions**

Component	Description
Client Cookie	Type the value of a cookie that resides on the client where this report is viewed.
Constant Value	Type a constant value. The value can be any text that complies with the expected data type. For a datetime type, you can type the datetime or use the datetime picker.  For example, you can use this option to specify the beginning of a URL, such as <code>http://www.example.com/</code> .
Prompt/Parameter	Select a parameter from the drop-down list. This option retrieves the current value of a parameter even if it is changed by driving or prompt input. If the current value is empty, default value of the parameter is used.  For this option to be usable, the query on which the view is based must reference a parameter in a filter. See <a href="#">Creating Parameters</a> , for more information.
Unique Button Click ID	A unique ID number is generated and incremented with each button click. Applies to <b>Open Dashboard</b> , <b>Open URL</b> , <b>Insert</b> , and <b>Update</b> actions only.  This ID is remembered across dashboards and data objects. It is useful for inserting or updating rows in an action list when you need a unique identifier for each row.
Value from Action List	Select a column (data field name) from the Action List. Applies to <b>Insert</b> and <b>Update</b> actions only.

## Using Active Data

Active data continuously aggregates data or displays a segment of data within a defined time window. As time passes in a time window, older data is removed from the view and newer data is added.

Active data is supported only for basic Area, Bar, Line, Combo, Pie, Scatter, and Bubble views. None of the other variations of these view types support active data. However, active data is

supported for all variations of Gauge views and for Point Theme and GPS Theme Geo Map views.

 **Note:**

If you have configured value formats such as Number, Currency, Percentage, or Date with Active Data Service on Business Views like List, Action List, and Collapsed List, then, when auto refreshing the table data, the complete table data gets refreshed instead of row-level update. If the table data is more and refresh rate is 1 second, the refresh will impact readability and render slow.

Active data is not supported for Horizontal Bar, KPI Watchlist, Treemap, or Color Theme Geo Map views. Active data is also not supported for external data objects.

The time zone in your Personalization does not control the time slice for active data. Active data time is based on UTC.

 **Note:**

A time series or group arranges data in a view based on a datetime field, while active data focuses on a particular time slice in a data object. These options can coexist in the same view.

If the **Use Time Series** option groups view values, the view data appears to slide from right to left as time passes. If the **Use Time Groups** option is selected, there is no sliding effect, but the data displayed in the stationary time groups changes as time passes.

See [Using a Time Group or a Time Series](#) for more information about grouping data by datetime values in the query on which a view is based.

**To configure active data:**

1. From the view tab, click **Runtime-Interaction**.
2. Click **Active Data**.

Active data may not be used on a predefined SQL query. Business Views created using a predefined SQL query are used for tactical data display and not active data.

 **Note:**

Active data is not supported with opaque queries. Therefore, do not enable active data where opaque queries are used.

3. Check the **Turn this query into a continuous query** box.
4. Optionally, check the **Allow runtime modification** box. This enables you to change active data properties at run time in the view mode.
5. To make data aggregation (computation of averages, sums, and so on) active, specify the following settings:

- **Active Data Collapsing** — Optionally check this box to enable continuous data aggregation.
- **Interval** — Specify how often the aggregation result is updated in the view.
- **Units** — Select Day, Hour, Minute, or Second for the Interval.

If you change the **Interval** and **Units** in a dashboard open for viewing, the new data snapshot is rendered only when the interval ends. For example, if the interval is set to one minute, the new snapshot is shown after one minute and then active data collapsing occurs.

6. To define a time window, specify the following settings:

- **Use a time window** — Optionally check this box to define a period over which and a frequency at which data is updated. For example, you could update the number of calls completed in the last hour every ten minutes.
- **Sliding Range Based on** — A date or time column upon which the rolling window calculation is based. For example, choosing Call Start Time enables the system to determine whether a given call is within the rolling window period.

This setting is displayed only if the selected data object is a relation. It is not supported for stream data objects.

 **Note:**

In JET—based views, the sliding tool may shrink if you adjust it manually as and when active data keeps populating in the view. This might invalidate the time window you have selected resulting in the business view not displaying any data. To correct this, set the business view to refresh automatically. Alternatively, you can set the entire dashboard to auto refresh.

- **Range Length** — The period over which data is displayed, from an amount of time in the past to the present. Examples of ranges are the last 30 minutes or the last 24 hours.
- **Update Interval** — The frequency at which the result within the rolling window is updated.
- **Units** — Select Day, Hour, Minute, or Second for the Range Length and Update Interval.

7. Click **Apply**.

## Editing, Renaming, Deleting, and Securing Business Views

This section contains information on modifying business view properties or deleting them.

You can edit, rename, delete, and secure views just as you can any other Oracle BAM entity.

### Editing a Business View

Use the following procedure to open, edit, and save a view. When you edit a view, the changes propagate to all dashboards that include the view.



**Note:**

If a view no longer works because the underlying query or data object has changed, you must close and reopen the view before you can edit and save it.

**To edit a view:**

1. In the left navigation pane, click the arrow to the left of **Business Views**.  
All saved views in the current project are displayed in a list.
2. Click the view name, or right-click the view name and select the **Edit** menu item.
3. Make the desired changes.
4. Click **Save**. If you do not see the Save option or other tools like the Display Info (i) button, ensure that your browser is on the full screen mode. In addition, collapse the navigation pane. This is a known issue.

## Renaming a Business View

Use the following procedure to change the **Display Name** of a view.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

**To rename a view:**

1. In the left navigation pane, click the arrow to the left of **Business Views**.  
All saved views in the current project are displayed in a list.
2. To rename the view, right-click the view name and select the **Rename** menu item.  
The name becomes text in an editable field.
3. Type the new name and press **Enter**.  
The new name appears in the list.

When you edit the view, the new **Display Name** is displayed on the tab. This name is also displayed when you see the view in a dashboard.

However, the internal **Name** remains unchanged. When you edit the view, this name is displayed on the left in the header.

## Deleting a Business View

Use the following procedure to delete a view. The view is removed from any dashboards that reference it.

**To delete a view:**

1. In the left navigation pane, click the arrow to the left of **Business Views**.  
All saved views in the current project are displayed in a list.
2. Click the view icon and click the **Delete** icon, or right-click the view name and select the **Delete** menu item.

A dialog asks you to confirm the view deletion. If any dashboards contain the view, they are listed.

3. Click **OK**.

The view disappears from the list.

## Securing a Business View

A view inherits security settings from the project in which it is created. For more information about projects, see [Planning and Creating Projects](#).

### To change security settings for a view:

1. In the left navigation pane, click the arrow to the left of **Business Views**.

All saved views in the current project are displayed in a list.

2. Right-click the view and select **Security** from the pop-up menu.

The security tab for the view opens.

3. To add a role or group to whom you can explicitly grant or deny permissions, follow these steps:

- a. Click the **Add** icon in the Grant Permissions or Deny Permissions table.

The Add Application Roles, Groups, and Users dialog opens.

See [Managing Oracle BAM Users](#) for information about how to add users to roles and groups.

- b. Type a **Name** for the role or group you are adding.

- c. Select from the drop-down **List**: Application Role or Group.

- d. Click **Search** to populate the Available Members list.

- e. To add a member to the Selected Members list, select the member and click the single right arrow.

- f. To add all members to the Selected Members list, select the member and click the double right arrow.

- g. To remove members from the Selected Members list, use the single and double left arrows.

- h. When the Selected Members list is final, click **OK**.

The Add Application Roles, Groups, and Users dialog closes, and the **Name** you specified appears in the table.

4. To remove a role or group, select the table row and click the **Remove** icon.

5. To grant permissions, select **Read**, **Write**, **Remove**, or **Security** for the users, roles, and groups listed in the Grant Permissions table.

6. To deny permissions, select **Read**, **Write**, **Remove**, or **Security** for the users, roles, and groups listed in the Deny Permissions table.

7. Click **Save**.

# 9

## Creating Dashboards

This chapter describes how to create dashboards in Oracle Business Activity Monitoring (Oracle BAM).

This chapter includes the following sections:

- [Understanding Dashboards](#)
- [Dashboard Prerequisites and Uses](#)
- [Creating a Dashboard and Adding Views](#)
- [Adding Other Content to Dashboard Cells](#)
- [Formatting Dashboard Cells](#)
- [Formatting the Dashboard](#)
- [Creating a Tabbed Dashboard](#)
- [Editing a Dashboard](#)
- [Renaming a Dashboard](#)
- [Deleting a Dashboard](#)
- [Securing a Dashboard](#)
- [Displaying a Dashboard URL](#)
- [Viewing a Dashboard](#)

## Understanding Dashboards

A *business view* or *view* is a visual representation of data fetched by a business query or one or more key performance indicators (KPIs). A *dashboard* is a collection of related business views.

A dashboard can also have external content using a URL or can contain an image in a dashboard cell. To display data to users, a view must be included in a dashboard. Dashboards can display point-in-time or real-time information (active data).

Users can view dashboards, and alerts can send users links to dashboards based on conditions changing in the queries and KPIs on which the business views within them are based.

When designing dashboards, you can choose from a variety of style templates based on how many views you need and how much horizontal and vertical space each view requires. Each view occupies a rectangular space, called a *cell*, in the dashboard. You can add, resize, clear the contents of, and delete cells.

A *tabbed dashboard* contains other complete dashboards, which you view by clicking the tabs.

A *springboard* is a dashboard that uses images from various dashboard cells to launch other dashboards. See [Creating a Springboard](#) for more information.

# Dashboard Prerequisites and Uses

Dashboard prerequisites are a set of tasks you must complete before you create a dashboard, to ensure its smooth functioning.

Before you can create a dashboard, you must do the following:

- Add to your project the data objects containing the data the dashboard will display. For more information about projects, see [Planning and Creating Projects](#). For more information about data objects, see [Working with Data Objects](#).
- If the dashboard will include a KPI Watchlist type view, create one or more KPIs on which to base the view. For more information about KPIs, see [Creating KPIs](#).
- If the dashboard will include prompts for users to choose values, create parameters for query filters. For more information about parameters, see [Creating Parameters](#).

You can use a bottom-up approach and create business queries and views before creating a dashboard. Or you can use a top-down approach and create a dashboard before creating the views and queries within it. See [Choosing Bottom-Up or Top-Down Project Building Steps](#), [Creating Business Queries](#), and [Creating and Using Business Views](#) for more information.

 **Note:**

You cannot create KPI Watchlist, Tree Map, or Geo Map views from within a dashboard. First create these views using the navigator pane, then you can add them to dashboards.

After you create dashboards, you can use them in the following ways:

**In a view action**

One type of view action opens a dashboard. See [Using Actions in Business Views](#) for more information.

**As a view drill-across target**

One type of view drill-across target is a dashboard. See [Configuring Drill-Across Targets](#) for more information.

**In an alert action**

Some alert actions send users links to dashboards. See [Creating Alerts](#) for more information.

**In another dashboard**

You can link to one dashboard from another. See [Adding Other Content to Cells](#) for more information. A tabbed dashboard contains other complete dashboards. See [Creating a Tabbed Dashboard](#) for more information.

**In an email, browser, or web page**

You can copy the dashboard link and use it as you would any other URL to provide single-click access to the dashboard. See [Displaying a Dashboard URL](#) for more information.

## Creating a Dashboard and Adding Views

This section outlines the procedure to create a dashboard and add views in it.

### To create a dashboard and add views:

1. Go to the **Designer** page.
2. Click **Dashboards** in the left panel navigator, or right-click **Dashboards** and select the **Create** menu item.

The Dashboards dialog opens.

3. Type a **Name**, optionally edit the **Display Name**, and optionally add a **Description**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the dashboard is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

4. Select **Dashboard** as the dashboard type.

To create a Tabbed Dashboard, see [Creating a Tabbed Dashboard](#).

5. Select a thumbnail representing the desired style template.

You can add, resize, clear the contents of, or delete cells later if needed. See [Formatting Dashboard Cells](#) for more information.

6. Click **Create**.

A tab opens for the new dashboard, with a default arrangement of cells corresponding to the selected style template.

7. To create and add a new view:

### **Note:**

You cannot create KPI Watchlist, Treemap, or Geo Map views from within a dashboard. First create these views using the navigator pane, then you can add them to dashboards.

- a. Drag and drop a view category icon from the **Views** palette to one of the cells.

The Create View dialog opens.

For an alternative to dragging and dropping, see [Adding Other Content to Cells](#).

See [Creating and Using Business Views](#) for more information about view icons, categories, and types.

- b. Type a **Name** and a **Display Name**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the view is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

- c. Select a thumbnail representing the desired view type.
- d. Click **Create**.

The new view opens in the cell, with a visual preview using generic data.

- e. Click the **Open Query Editor** icon in the top right corner of the cell that contains the view.

The Business Query Editor opens.

- f. Select a query from the **Query** drop-down list or create a query by clicking the **Create New Query** icon.

See [Creating Business Queries](#) for more information about creating and editing queries.

 **Note:**

BAM supports date-only data filter using existing date-time parameters. During query filter configuration do not configure the same parameter for date-only and date-time for a **Dashboard**. As a designer, you should regard the following configurations in query filter area:

- Do not select an existing date-time parameter for data-only compare mode parameter in query filter popup.
- Always create two sets of BAM parameters, such as one for date-time, another one for date-only comparison.
- Do not select the following comparison operators, when the parameter default value is NULL:
  - less than
  - greater than
  - less than equal to
  - greater than equal to
- In the above cases, the Dashboard may display wrong data during runtime.

- g. Click **Apply**, then click **Close**.

The preview changes to reflect the data fields fetched by the query.

- h. To edit view properties or runtime interactions, click the **Business View Actions** icon in the top right corner of the cell that contains the view.

See [Creating and Using Business Views](#) for more information about view properties and runtime interactions.

- 8. To add an existing view from the current project, drag and drop a view from the **Project Views** palette or the left navigation pane to one of the cells.

The view opens in the cell, with a visual preview using data fields fetched by the query on which the view is based.

Click the **Refresh Views List** icon to update the **Project Views** palette.

For an alternative to dragging and dropping, see [Adding Other Content to Cells](#).

9. If the queries on which the views are based reference parameters, you can coordinate the parameters in the dashboard. Click the **Parameter Mapping Editor** icon in the top right corner of the dashboard.

See [Driving Parameters from Other Views](#) for more information.

10. To add items other than views to dashboard cells, see [Adding Other Content to Cells](#).
11. To format the cells, see [Formatting Dashboard Cells](#).
12. To format the title and background of the dashboard, see [Formatting the Dashboard](#).
13. Click **Save**.

 **Note:**

If you do not save the dashboard, any views and queries you create from the dashboard are not saved and do not appear in the left navigation pane.

## Creating a Springboard

This section describes how you can create a springboard.

To create a springboard:

1. Create a Dashboard. See [Creating a Dashboard and Adding Views](#) for the detailed procedure.
2. Click the + icon.
3. In the dialog box that appears, click **Create Springboard View**.
4. Upload an image from a file. Click the **Stretched** radio button to stretch the image to fit within the borders. Click the **Fixed** radio button to retain the original size of the image.
5. Choose an **Image Caption**. Click the **Top** or **Bottom** radio buttons to select the placement of the caption.
6. Click the **Link with Dashboard** drop-down list to select a Dashboard with which you might want to link this Springboard with.
7. Optionally, click the **Replace the same view** check box to replace the springboard view with the dashboard selected.

 **Note:**

Oracle Business Activity Monitoring ships with a collection of pre-assembled dashboards that aid with the monitoring of some common business processes. See [Process Analytics Springboard](#) for more information.

## Process Analytics Springboard

The Process Analytics Springboard is a collection of dashboards within the process analytics project.

The Process Analytics Springboard is a collection of pre-assembled dashboards that aid the monitoring of some common business processes. This springboard consists of the following dashboards:

- Process Health Dashboard
- Open Process Summary Dashboard
- Workload Analysis Dashboard
- Cycle Time Dashboard
- Closed Process Analytics Dashboard

### Process Health Dashboard

The Process Health Dashboard helps you identify processes that need attention and fixing. Each Application or Process shows the number of active processes in relation to process instances that have recoverable errors or are suspended.

- In Progress process instances include the total number of processes that are "open", "due soon" and "overdue" but otherwise functioning normally
- Recoverable process instances are instances that may have experienced errors but can be recovered by taking suitable action.
- Suspended process instances are instances that have been paused for some reason other than an explicit error or fault.

## Configuring a Process Health Dashboard

This task describes how you can configure a Process Health Dashboard

To configure a Process Health Dashboard:

1. Click the split pane on the right side of the panel, if it is not already expanded.
2. By default, the process is ordered by the total number of instances from right to left, so as you progress down the page, each process would have fewer instances.
3. Under the Parameters panel, there are all processes listed with check boxes next to them. By default, all processes are selected.
4. You can uncheck the processes you don't want to view by clicking the check box
5. You can search for a process by clicking in the Search String field.
6. Once you make the necessary selections, click Apply.
7. The bar chart displays the number of instances on the X axis and the process names on the Y axis. Each bar is a color-coded split among Active, Recoverable, and Suspended instances.

### Note:

There is no drill down from this dashboard.

## Open Process Summary Dashboard

The Open Process Summary dashboard is a single top-level table view of the total instances that are currently open for all processes that you have access to.

This dashboard provides counts of instances requiring admin action in tabs above the table view. The tabs from left to right are: Total Open, On Track, DUE this week, OVERDUE, OPENED today, and CLOSED today.

## Configuring the Open Process Summary Dashboard

This topic describes how you can configure an Open Process Summary Dashboard

To configure the Open Process Summary Dashboard:

1. Click the split pane on the right side of the panel, if it is not already expanded
2. By default, the table is sorted alphabetically using the first column – Application. You can change the sorting by clicking the Up or Down Sort buttons next to each column name, namely: Application, Process, On Track, Due this Week, Overdue, Recoverable, Suspended, Opened Today, and Closed Today.
3. Under the Parameters panel, there are all processes listed with check boxes next to them. By default, all processes are selected.
4. You can uncheck the processes you don't want to view by clicking the check box.
5. You can search for a process by clicking in the Search String field.
6. Once you make the necessary selections, click Apply.

## Workload Analysis Dashboard

The Workload Analysis Dashboard combines the Process Activity Dashboard and the Open Process Bottleneck dashboard.

Using this dashboard, you can:

- Determine the underlying resource creating a bottleneck.
- Determine if there is an external system not responding in time.

### Using the Workload Analysis Dashboard

You can toggle Top 10 and Bottleneck views using the tabs available at the top left of the dashboard view. The Top 10 view lists processes By Task and By Assignee. The Bottleneck view is a Treemap view that you can sort By Process, By Task, and By Assignee.

#### Top 10 View — View by Task

Each chart shows the top ten tasks ordered as follows:

- ON TRACK is ordered from the most to the least Open/On Track task items (green section) for an activity.
- DUE THIS WEEK is ordered by the most to the least Due This Week task items (yellow section) for an activity.
- OVERDUE is ordered by the most to the least Overdue task items (red section) for an activity.

#### Top 10 View — View by Assignee

Each chart shows the top ten tasks ordered as follows:

- ON TRACK is ordered from the most to the least Open/On Track task items (green section) for a particular user.

- DUE THIS WEEK is ordered by the most to the least Due This Week task items (yellow section) for a particular user.
- OVERDUE is ordered by the most to the least Overdue task items (red section) for a particular user.

### **Bottleneck View**

This dashboard contains treemap views that show all processes and allows zooming in to a particular process to see user loads.

## Cycle Time Dashboard

This dashboard shows the cycle time for processes and long running activities or tasks for processes that you have access to.

The cycle time shows the trend in number of days to complete a process, instance, or task.

You can either 'View by Process' or 'View by Task'. The 'By Process' tab contains the Process Cycle Time and Process Workload line charts. The By Task tab shows the Task Cycle Time and Task Workload line charts.

### **View By Process**

Both the Process Cycle Time and Process Workload charts show the number of days on the Y-axis and the months from January to December on the X-axis.

All processes are color-coded and you can zoom in to a particular process to see the Inflow and Outflow with the number of opened vs. closed instances alongside the process averages for opened and closed.

### **View by Task**

The Task Cycle Time line chart shows the amount of time taken per task.

The Task Workload line chart shows the number of tasks opened and closed, with the number of tasks on the Y-axis and the time on the X-axis.

## Configuring the Cycle Time Dashboard

This topic describes how you can configure the Cycle Time dashboard.

To configure the Cycle Task dashboard:

1. Click the split pane on the right side of the panel, if it is not already expanded.
2. In the Date Range (days before today) box, enter a date range. The default value is 30.
3. Under the Parameters panel, there are all processes listed with check boxes next to them. By default, all processes are selected. Uncheck the processes you don't want to view by clicking the check box.
4. In the Show Top N box, enter a value for the maximum number of results you want to view for the top processes. The default value is 10.

## Closed Process Analytics Dashboard

The Closed Process Analytics Dashboard is a table view of closed processes that you have access to.

It enables you to view metrics within a historical context and compare data within two intervals (this month vs. the same month last year, or this month vs. the average over the last 12 months).

There are three tabs: Today, This Week, and This Month. By default, the Today tab is open.

The dashboard has Count Boxes on top, within all three tabs. These are, from left to right: Total Closed This Month, Completed This Month, Aborted This Month, and Errored This Month.

## Configuring the Closed Process Analytics Dashboard

This topic describes how you can configure a Closed Process Analytics dashboard.

To configure a Closed Process Analytics dashboard:

1. Click the split pane on the right side of the panel, if it is not already expanded.
2. Under the Parameters panel, there are all processes listed with check boxes next to them. By default, all processes are selected. Uncheck the processes you don't want to view by clicking the check box.
3. Click Apply.

## Configuring the Workload Analysis Dashboard

This topic describes how you can configure the Workload Analysis Dashboard.

To configure the Workload Analysis Dashboard:

1. Click the split pane on the right side of the panel, if it is not already expanded.
2. Under the Parameters panel, there are all processes listed with check boxes next to them. By default, all processes are selected. Uncheck the processes you don't want to view by clicking the check box.
3. You can search for a process by clicking in the Search String field.
4. Under the Parameters panel, all Assignees are listed with check boxes next to them. By default, there are two Assignee groups: Roles and Groups, and Users. Both options are selected by default. You can uncheck the assignees you don't want to view by clicking the check box.

## Adding Other Content to Dashboard Cells

In the center of each cell is an **Add** icon which helps you add content to dashboard cells.

Clicking this icon displays a menu of items you can add to a cell:

- **Create a View** creates a new view and adds it to the cell.  
The steps are the same as if you had dragged and dropped an icon from the **Views** palette, except that you must select a view category as well as a view type. See [Creating a Dashboard and Adding Views](#) for more information.
- **Browse** opens the Resource Catalog, from which you can add a view in the project to the cell.
- **Open URL** references the content at the specified URL and displays it in the cell. Be sure to include the full URL, including the protocol, such as `http://`, or an error will result.
- **Insert Image** uploads the specified graphics file and displays it in the cell. You can also specify another dashboard to link to, so that when the user clicks on the image, the other

dashboard opens in another browser window. Supported image file types are .png, .gif, and .jpg.

Click **Save** after making any of these changes.

 **Note:**

If you do not save the dashboard, any views and queries you create from the dashboard are not saved and do not appear in the left navigation pane.

## Formatting Dashboard Cells

In the top right corner of each cell is a **Dashboard Actions** icon which helps you format dashboard cells.

Clicking this icon displays a menu of cell formatting options:

- **Add New Cell Left** adds a new cell to the left of the current cell and either splits the cell or adjusts cells in the same row.
- **Add New Cell Right** adds a new cell to the right of the current cell and either splits the cell or adjusts cells in the same row.
- **Add New Cell Below** adds a new cell below the current cell and either splits the cell or adjusts cells in the same column.
- **Add New Cell Up** adds a new cell above the current cell and either splits the cell or adjusts cells in the same column.
- **Change Width** displays a slider you can use to adjust the current cell width.
- **Change Height** displays a slider you can use to adjust the current cell height.
- **Clear** deletes the contents of the current cell from the dashboard but leaves the empty cell in place.
- **Delete the Cell** deletes the current cell from the dashboard and adjusts any cells in the same row or column.

You can also add a cell at the bottom of the dashboard by clicking the **Add New Cell** icon. This icon is to the left of the **Save** button in the top right corner of the dashboard.

Click **Save** after making any of these changes.

## Formatting the Dashboard

In the top left corner of the dashboard is a **Change Dashboard Properties** icon which helps you format dashboards.

Clicking this icon displays the Dashboard Properties window. Changing dashboard properties is optional.

Clicking **Title** displays the following properties:

- **Font** — Select a font from the drop-down list. The default is Tahoma.
- **Font Size** — Specifies the point size of the font. The default is 11.

- **Foreground Color** — Specifies the text color. Click the down arrow to display a palette. The default is white.
- **Background Color** — Specifies the background color of the dashboard banner. Click the down arrow to display a palette. The default is blue.
- **B I U** — Specifies whether text is bold, italic, underlined, or a combination. The default is plain text.
- **Subscript, Superscript, Strikethrough** — Specifies whether text is subscripted, superscripted, struck through, or a combination. The default is plain text.
- **Outdent, Indent** — Outdents or indents title text.
- **Text** — Specifies the title text to be displayed.

Clicking **Background** displays the following properties:

- **Default** — Specifies the default background, which is plain white.
- **Color** — Specifies the background color of the dashboard. Specify a hexadecimal or RGB color code or click the down arrow to display a palette. The default is white.
- **Image** — Specifies an image file for the background. Supported image file types are .png, .gif, and .jpg.
- **Centered, Tiled** — Specifies whether the background image is centered and used once, or whether it repeats with a tiled appearance.

To save the properties you have edited, click **OK**, then click **Save**.

## Creating a Tabbed Dashboard

A tabbed dashboard contains other complete dashboards, which you view by clicking the tabs. Before you can build a tabbed dashboard, you must create the other dashboards it will contain.

**To create a tabbed dashboard:**

1. Go to the **Designer** page.
2. Click **Dashboards** in the left panel navigator, or right-click **Dashboards** and select the **Create** menu item.

The Dashboards dialog opens.

3. Type a **Name**, optionally edit the **Display Name**, and optionally add a **Description**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the dashboard is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

4. Select **Tabbed Dashboard** as the dashboard type.

To create a regular dashboard, see [Creating a Dashboard and Adding Views](#).

5. Click **Create**.

A blank new tabbed dashboard opens.

6. Click the **Select Dashboards to Add** icon in the top right corner of the dashboard.

The Select Dashboards to Add dialog opens.

7. Check the boxes to the left of the dashboards to be included.
8. Click **OK**.  
The selected dashboards are added in the order of their internal names, and the tabbed dashboard is saved.
9. To remove an included dashboard from the tabbed dashboard, click the X on its tab.  
The dashboard is removed, and the tabbed dashboard is saved.

## Editing a Dashboard

This section outlines the procedure to edit a dashboard.

Use the following procedure to open a dashboard for editing, edit it, and save it.

### To edit a dashboard:

1. In the left navigation pane, click the arrow to the left of **Dashboards**.  
All saved dashboards in the current project are displayed in a list.
2. Click the dashboard name, or right-click the dashboard and select the **Edit** menu item.
3. Make the desired changes.
4. Click **Save**.

You can edit the properties or runtime interactions of a view from the dashboard using the **Business View Actions** icon. However, to switch the view type, you must edit the view by opening it from the left navigation pane. See [Creating and Using Business Views](#) for more information.

You can edit the configuration of the query on which a view is based from the dashboard using the **Open Query Editor** icon. See [Creating Business Queries](#) for more information about query configuration.

## Renaming a Dashboard

This section outlines the procedure to rename a dashboard.

Use the following procedure to change the **Display Name** of a dashboard.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

### To rename a dashboard:

1. In the left navigation pane, click the arrow to the left of **Dashboards**.  
All saved dashboards in the current project are displayed in a list.
2. To rename the dashboard, right-click the dashboard name and select the **Rename** menu item.  
The name becomes text in an editable field.
3. Type the new name and press **Enter**.  
The new name appears in the list.

When you edit the dashboard, the new **Display Name** is displayed on the tab. This name is also displayed when you view the dashboard.

However, the internal **Name** remains unchanged. When you edit the dashboard, this name is displayed on the left in the header. The dashboard URL uses the internal name, so view actions, view drill-across targets, alert actions, and links from other dashboards that reference the renamed dashboard are unaffected.

## Deleting a Dashboard

This section outlines the procedure to delete a dashboard.

Use the following procedure to delete a dashboard. Any view actions, view drill-across targets, alert actions, or links from other dashboards that reference the deleted dashboard must be edited.

### To delete a dashboard:

1. In the left navigation pane, click the arrow to the left of **Dashboards**.  
All saved dashboards in the current project are displayed in a list.
2. Click the dashboard icon and click the **Delete** icon, or right-click the dashboard name and select the **Delete** menu item.  
A dialog asks you to confirm the dashboard deletion.
3. Click **OK**.  
The dashboard disappears from the list.

To delete a view or other content from a dashboard cell, click the **Dashboard Actions** icon in the cell and select **Clear** from the menu.

## Securing a Dashboard

A dashboard inherits security settings from the project in which it is created.

For more information about projects, see [Planning and Creating Projects](#).

### To change security settings for a dashboard:

1. In the left navigation pane, click the arrow to the left of **Dashboards**.  
All saved dashboards in the current project are displayed in a list.
2. Right-click the dashboard and select **Security Settings** from the pop-up menu.  
The security tab for the dashboard opens.
3. To add a role or group to whom you can explicitly grant or deny permissions, follow these steps:
  - a. Click the **Add** icon in the Grant Permissions or Deny Permissions table.  
The Add Application Roles, Groups, and Users dialog opens.  
See [Managing Oracle BAM Users](#) for information about how to add users to roles and groups.
  - b. Type a **Name** for the role or group you are adding.
  - c. Select from the drop-down **List**: Application Role or Group.
  - d. Click **Search** to populate the Available Members list.
  - e. To add a member to the Selected Members list, select the member and click the single right arrow.

- f. To add all members to the Selected Members list, select the member and click the double right arrow.
  - g. To remove members from the Selected Members list, use the single and double left arrows.
  - h. When the Selected Members list is final, click **OK**.  
The Add Application Roles, Groups, and Users dialog closes, and the **Name** you specified appears in the table.
4. To remove a role or group, select the table row and click the **Remove** icon.
  5. To grant permissions, select **Read**, **Write**, **Remove**, or **Security** for the users, roles, and groups listed in the Grant Permissions table.
  6. To deny permissions, select **Read**, **Write**, **Remove**, or **Security** for the users, roles, and groups listed in the Deny Permissions table.
  7. Click **Save**.

Make sure all the views in the dashboard grant the same or greater permissions to users. See [Securing a Business View](#) for more information.

## Displaying a Dashboard URL

This section helps you find the dashboard URL for any dashboard you view in the BAM composer.

Use the following procedure to display the URL of a dashboard.

### To display the URL of a dashboard:

1. In the left navigation pane, click the arrow to the left of **Dashboards**.  
All saved dashboards in the current project are displayed in a list.
2. To display the URL of the dashboard, right-click the dashboard name and select the **Show Dashboard URL** menu item.

The ADF and JET-based URLs appear in a small window. The syntax for an ADF dashboard is as follows:

```
http://host:port/bam/composer/faces/proxy?page?project=project&dashboard=dashboard  
&DashboardParameters=(Parameter1=value;Parameter2=value)
```

For example:

```
http://www.example.com:8001/bam/composer/faces/proxy?page?  
project=MyProject&dashboard=MyDashboard
```

The syntax for the JET dashboard is as follows:

```
http://host:port/bam/composer/jet/dashboard?project=project&dashboard=dashboard  
&DashboardParameters=(Parameter1=value;Parameter2=value)
```

For example:

```
http://www.example.com:8001/bam/composer/jet/dashboard?  
project=MyProject&dashboard=MyDashboard
```

You can also hide the parameter panel on the right by adding the `hideParameterPanel` parameter to the URL. For example:

```
http://www.example.com:8001/bam/composer/faces/proxy?page?
project=MyProject&dashboard=MyDashboard&hideParameterPanel=true
```

This is not applicable for JET-based dashboards.

3. Click the X in the top right corner to close the window.

The URL uses the internal **Name** of the dashboard. Therefore, renaming the dashboard, which changes the **Display Name**, does not affect view actions, view drill-across targets, alert actions, and links from other dashboards that reference the renamed dashboard.

You can use the URL to add a BAM dashboard link to a web page. For example:

```
<html>
<body>
<iframe src="http://www.example.com:8001/bam/composer/faces/proxy?page?
project=MyProject&dashboard=MyDashboard width="100%" height="100%">
</iframe>
</body>
</html>
```

You can set the `height` and `width` attributes of the `iframe`.

## Viewing a Dashboard

This section outlines the procedure to view a dashboard from the Designer page.

### To open a dashboard for viewing:

1. In the left navigation pane, click the arrow to the left of **Dashboards**.  
All saved dashboards in the current project are displayed in a list.
2. Click the dashboard icon and click the **Open** icon, or right-click the dashboard name and select the **Open** menu item.  
The dashboard opens in a new browser window.
3. To display parameters for which you can change values, click the arrow in the center of the right edge of the dashboard.  
The Parameters list is displayed.
4. Change parameter values as desired.  
The views in the dashboard change in response to the parameter values. For more information about parameters, see [Creating Parameters](#).

The main part of BAM from which you open dashboards for viewing is the BAM Viewer (the Home page). From the Designer page, you can configure how a dashboard is handled in the BAM Viewer:

- To add the dashboard to the Favorites list in the BAM Viewer, right-click it and select the **Favorites** menu item.
- To automatically launch the dashboard whenever you log in to BAM, right-click it and select the **Start on Launch** menu item.

See [Using Oracle Business Activity Monitoring as a Viewer](#) for more information about using the BAM Viewer.

# 10

## Creating Alerts

This chapter describes how to create and use alerts in Oracle Business Activity Monitoring (BAM). It describes how to specify the events and conditions under which an alert fires, include messages with alerts, manage chained alerts, and use actions.

This chapter includes the following sections:

- [Understanding Alerts](#)
- [Alert Prerequisites and Uses](#)
- [Creating an Alert](#)
- [Activating and Deactivating an Alert](#)
- [Creating Dependent Alerts](#)
- [Editing an Alert](#)
- [Renaming an Alert](#)
- [Deleting an Alert](#)
- [Securing an Alert](#)
- [Viewing Alert History from the Home Page](#)
- [Launching Alerts by Invoking Web Services](#)
- [Alert and Web Service Example](#)
- [Sending Alerts to External Email Accounts](#)
- [Alert Events](#)
- [Alert Conditions](#)
- [Alert Actions](#)

## Understanding Alerts

An *alert* performs one or more actions when launched by an event and filtered by one or more conditions.

An *event* can be an amount of time, a specific time, a date and time, a repeating event between two dates, a change in a data object, output from a continuous query, or a manual event. A *condition* restricts the alert to an event occurring between two times or dates or to a specific day of the week. An *action* can send a notification, perform a data object operation, invoke a web service, call an external method, or launch other alerts.



### Note:

If you do not set a time zone preference, time-based alerts are based on coordinated universal time (UTC). See [Setting Language\\_ Time Zone\\_ and Accessibility Preferences](#) for more information.

Each alert name in the left panel navigator has an icon indicating its status as described in [Table 10-1](#).

**Table 10-1 Alert Icons**

Icon	Description
	<b>Active</b> indicates that the alert is active and fires in response to its configured event and conditions. Newly created alerts are active by default.
	<b>Inactive</b> indicates that the alert is inactive and does not fire. You can reactivate the alert. See <a href="#">Activating and Deactivating an Alert</a> .
	<b>Invalid</b> indicates that an alert has become orphaned or broken due to an error. The alert might require correction. For example, an alert is invalid if it is based on a data object, KPI, or continuous query that has changed. Alerts displayed with this icon do not fire again until they are edited and corrected.
	<b>Expired</b> means that the alert does not fire again. This icon is displayed for time-based alerts that fire only once, after the alert has fired. However, these alerts can be edited and reused, resetting the state to Active.

Inactive and expired alerts behave differently. An alert can be deactivated only if it is running. Deactivation benefits users who do not want actions to be executed for some time interval. Alerts that are inactive, but still valid, can be activated again.

An expired alert no longer fires because its event or conditions will not occur again. However, to reuse an expired alert, you can edit it to make it valid and save it. The alert is reloaded and is ready to fire again.

## Alert Prerequisites and Uses

Alert prerequisites are a set of tasks you must complete before you create an alert, to ensure its smooth functioning.

Before you can create an alert, you must do the following:

- If the alert event is a change to a data field, add to your project the data object containing that data field. For more information about projects, see [Planning and Creating Projects](#). For more information about data objects, see [Working with Data Objects](#).
- If the alert event is a continuous query, create the query. For more information about continuous queries, see [Creating a Continuous Query](#).
- If the alert action is to send a dashboard to users, create the dashboard. For more information about dashboards, see [Creating Dashboards](#).
- If the alert action is deleting data object rows, add that data object to your project. For more information about projects, see [Planning and Creating Projects](#). For more information about data objects, see [Working with Data Objects](#).

After you create alerts, you can use them in the following ways:

**In a query**

You can configure an alert action for a continuous query. This alert action is specific to the query and does not appear in the left panel navigator under Alerts. As an alternative, you can configure an alert to watch a continuous query as its triggering event; see [Select a CQL Query](#). For more information about continuous queries, see [Creating a Continuous Query](#).

**In a KPI**

You can configure alert actions for a key performance indicator (KPI). These alert actions are specific to the KPI and do not appear in the left panel navigator under Alerts. For more information about KPIs, see [Creating KPIs](#).

**With a web service**

You can configure a web service to trigger an alert. For more information, see [Launching Alerts by Invoking Web Services](#).

**To trigger another alert**

The action of one alert can be the event trigger of another. For more information, see [Creating Dependent Alerts](#).

## Creating an Alert

This section outlines the procedure to create an alert.

**To create an alert:**

1. Go to the **Designer** page.
2. Click **Alerts** in the left panel navigator, or right-click **Alerts** and select the **Create** menu item.

The Alerts dialog opens.

3. Type an **Alert Name**, optionally edit the **Alert Display Name**, and click **Create**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the alert is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

A tab opens for the new alert.

4. Click the down arrow to the right of **Select an Event Trigger**. You must specify only one event. The options are:
  - In a specific amount of time
  - At a specific time today
  - On a certain day at a specific time
  - Every interval between two times
  - Every date interval starting on a certain date at a specific time
  - Select a CQL Query
  - When a data field in a data object meets specified conditions
  - When this rule is launched

See [Alert Events](#) for detailed descriptions of each event.

5. Supply additional information for the event. For example, if you select **At a specific time today**, a **<select time>** setting appears. Possible settings are:
  - **<select time interval>** — Click this setting and select hours, minutes, and seconds. The default is one second.
  - **<select time>** — Click this setting and select hours, minutes, and seconds. The default is the current time.
  - **<select date>** — Click this setting and type a date, or click the Select Date icon and select a date from the pop-up calendar. The default is today.
  - **DAY** — Click this setting and change it to WEEK, MONTH, or YEAR if necessary.
  - **<continuous query>** — Click this setting, select a continuous query from the pop-up drop-down list, and select **OK**.
  - **<this data field has a condition of x>** — Click this setting and use the Data Based Event Editor to select a data object, data field, and condition.
  - **<select user>** — Click this setting and use the Identity Browser to select a user.

For datetime alerts, select a **query**. This ensures that the alert action you specify is only triggered if the query you select yields some results.

Selecting **When this rule is launched** doesn't display any additional settings, but specifies that this is a dependent alert. For more information, see [Creating Dependent Alerts](#).

An **<add action>** setting appears after you have fully specified the event.

6. Click the **<add action>** setting.  
The Action Editor appears.
7. Optionally select one or more conditions. The options are:
  - If it is between two times
  - If it is between two days
  - If it is a particular day of the week

See [Alert Conditions](#) for detailed descriptions of each condition.

 **Note:**

If a calculated field function which is not supported by CQL is used in an alert, then the alert query becomes a scheduled query, and will execute every thirty seconds after the first trigger, even in the absence of new triggers.

8. Select one or more actions. The options are:
  - Send a dashboard link via email
  - Send a message via email
  - Delete rows from a data object
  - Insert values into a data object
  - Send a parameterized message
  - Launch a rule
  - Run a web service

- Launch a Custom Action

See [Alert Actions](#) for detailed descriptions of each action.

9. Select **OK** to close the Action Editor.

To change a condition or action, click the down arrow to the right of the gear icon and select **Edit** from the pop-up menu. The Action Editor is redisplayed.

10. Supply additional information for the conditions and actions. For example, if you select **Send a message via email**, then **<create message>** and **<select user>** settings appear. Possible settings for the conditions are:

- **<select time>** — Click this setting and select hours, minutes, and seconds. The default is the current time.
- **<select date>** — Click this setting and type a date, or click the Select Date icon and select a date from the pop-up calendar. The default is today.
- **MONDAY** — Click this setting and change it to a different day of the week if necessary.

Possible settings for the actions are:

- **<select dashboard>** — Click this setting and use the Send Dashboard Link via Email Editor to select email addresses and a dashboard.
- **<select user>** — Click this setting and use the Identity Browser to select a user.
- **<create message>** — Click this setting and use the Message Editor to select email addresses, create an email message, include a dashboard, or set parameters, depending on the action.
- **<select data object>** — Click this setting and use the Map Fields dialog to map event data object fields to action data object fields.
- **<set parameters>** — Click this setting and use the Message Editor to select dashboard parameters.
- **<select rule>** — Click this setting and select a dependent alert. For more information, see [Creating Dependent Alerts](#).
- **<this data field has a condition of x>** — Click this setting and use the Data Based Event Editor to select a data object, data field, and condition.
- **<configure web service>** — Click this setting and use the Configure Web Service dialog to configure a web service.
- **<select odi service>** — Click this setting and select an Oracle Data Integrator service.
- **<custom action>** — Click this setting and use the Configure Custom Action dialog to specify a Java class that performs a custom action.

11. Click **Save**.

The alert is added to list in the left panel navigator and is active by default.

## Activating and Deactivating an Alert

Oracle BAM allows you to activate and deactivate alerts, thus making alert management easier.

When you create an alert, it is active by default. If you want an alert to be temporarily inactive but you do not want to delete it, you can deactivate it.

Administrative users can change the activity status of their own alerts and alerts created by other admin and non-admin users.

## To Change the Activity Status from the Navigator

1. Right-click the alert in the left panel navigator.
2. Select **Activate** or **Deactivate** from the pop-up menu.

On this menu, an active alert shows a blue check box for a grayed **Activate** option and a gray check box for a selectable **Deactivate** option. An inactive alert shows a gray check box for a selectable **Activate** option and a blue check box for a grayed **Deactivate** option.

## To Change the Activity Status from the Alert Tab

1. In the left panel navigator, click the arrow to the left of **Alerts**.  
All saved alerts in the current project are displayed in a list.
2. Click the alert name, or right-click the alert name and select the **Edit** menu item.
3. Check or uncheck the box to the left of the alert name in the upper left corner of the alert tab.

If this box is checked, the alert is active. If this box is not checked, the alert is inactive.

4. Click **Save**.

Activating an alert does not cause an alert to launch, it only enables it so that if the event occurs, the alert launches.



### Note:

An alert fires only if its triggering event conditions are met from the point in time the alert is defined (or reenabled) and forward. An alert does not fire if its conditions were met before it was defined or while it was inactive.

## Creating Dependent Alerts

You can create alerts that launch other alerts.

There is no limit to how many alerts you can chain in this way.

You can chain two alerts by creating:

- A dependent alert that must be launched by another alert
- An alert with an action to launch a dependent alert

### To create a dependent alert:

1. Create an alert with the event **When this rule is launched**. This event requires no additional information.
2. Create an alert that includes the action **Launch a rule**.
3. Click **<select rule>** in the action.

The Select Dependent Rule dialog opens.

4. Select a dependent alert. Only alerts that use the **When this rule is launched** event are listed.

5. Click **OK**.

## Editing an Alert

This section outlines the procedure to edit an alert.

### To edit an alert:

1. In the left panel navigator, click the arrow to the left of **Alerts**.  
All saved alerts in the current project are displayed in a list.
2. Click the alert name, or right-click the alert and select the **Edit** menu item.
3. Make the desired changes.
4. Click **Save**.

## Renaming an Alert

This section outlines the procedure to rename an alert.

Use the following procedure to change the **Display Name** of an alert.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

### To rename an alert:

1. In the left navigation pane, click the arrow to the left of **Alerts**.  
All saved alerts in the current project are displayed in a list.
2. To rename the alert, right-click the alert name and select the **Rename** menu item.  
The name becomes text in an editable field.
3. Type the new name and press **Enter**.  
The new name appears in the list.

When you edit the alert, the new **Display Name** is displayed on the tab. This name is also displayed when you view the alert in the Alert History list.

However, the internal **Name** remains unchanged. When you edit the alert, this name is displayed on the left in the header. The ManualRuleFile web service also references the internal name. See [Launching Alerts by Invoking Web Services](#) for more information.

## Deleting an Alert

Administrative role users can delete their own alerts and alerts created by other admin and non-admin users.

1. In the left panel navigator, click the arrow to the left of **Alerts**.  
All saved alerts in the current project are displayed in a list.
2. Click the alert icon and click the **Delete** icon, or right-click the alert name and select the **Delete** menu item.  
A dialog asks you to confirm the alert deletion.

3. Click **OK**.  
The alert disappears from the list.

## Securing an Alert

An alert inherits security settings from the project in which it is created.

For more information about projects, see [Planning and Creating Projects](#).

### To change security settings for an alert:

1. In the left panel navigator, click the arrow to the left of **Alerts**.  
All saved alerts in the current project are displayed in a list.
2. Right-click the alert and select **Security Settings** from the pop-up menu.  
The security tab for the alert opens.
3. To add a role or group to whom you can explicitly grant or deny permissions, follow these steps:
  - a. Click the **Add** icon in the Grant Permissions or Deny Permissions table.  
The Add Application Roles, Groups, and Users dialog opens.  
See [Managing Oracle BAM Users](#) for information about how to add users to roles and groups.
  - b. Type a **Name** for the role or group you are adding.
  - c. Select from the drop-down **List**: Application Role or Group.
  - d. Click **Search** to populate the Available Members list.
  - e. To add a member to the Selected Members list, select the member and click the single right arrow.
  - f. To add all members to the Selected Members list, select the member and click the double right arrow.
  - g. To remove members from the Selected Members list, use the single and double left arrows.
  - h. When the Selected Members list is final, click **OK**.  
The Add Application Roles, Groups, and Users dialog closes, and the **Name** you specified appears in the table.
4. To remove a role or group, select the table row and click the **Remove** icon.
5. To grant permissions, select **Read**, **Write**, **Remove**, or **Security** for the users, roles, and groups listed in the Grant Permissions table.
6. To deny permissions, select **Read**, **Write**, **Remove**, or **Security** for the users, roles, and groups listed in the Deny Permissions table.
7. Click **Save**.

## Viewing Alert History from the Home Page

You can view a recent history of alert activity on the Alerts tab in the BAM Viewer (the Home page).

See [Viewing Alert History](#) for more information.

## Launching Alerts by Invoking Web Services

You can use the ManualRuleFire web service to launch alerts triggered by the [When this rule is launched](#) event. For more information, refer to:

```
http://host:port/OracleBAMWS/WebServices/ManualRuleFire?WSDL
```

[Using the ManualRuleFire Web Service](#) further describes this web service.

You identify the alert to launch using a web service parameter of the following format:

```
projectname.alertname
```

Both of these names are internal names, not display names.

This is different from configuring an alert to launch a web service. For more information, see [Run a web service](#).

For an example of both the **Run a web service** action and the ManualRuleFire web service, see [Alert and Web Service Example](#).

## Alert and Web Service Example

This example shows a **Run a web service** alert action that invokes the ManualRuleFire web service. This is not the most efficient way for one alert trigger another (see [Creating Dependent Alerts](#)). It is intended only to clarify how both the **Run a web service** action and the ManualRuleFire web service are configured.

For more information, see [Run a web service](#) and [Using the ManualRuleFire Web Service](#).

**To configure the ManualRuleFire web service in the Run a web service action:**

1. Create the alert. See [Creating an Alert](#).
2. Select the action **Run a web service**.
3. Click **<configure web service>** in the alert expression.  
The Configure Web Service dialog opens.
4. Type the ManualRuleFire web service endpoint URL in the **WSDL or WSIL URL** field. The format is as follows:

```
http://host:port/OracleBAMWS/WebServices/DataObjectOperationsByID?WSDL
```

For example:

```
http://localhost:7001/OracleBAMWS/WebServices/DataObjectOperationsByID?WSDL
```

5. Check the **Secure Web Service** box and enter the required **Username** and **Password** credentials. In a development environment, these are often the administrator username and password used for both Oracle WebLogic Server and BAM.
6. Click **Display Operations** to display the available services of the web service.
7. Select the ManualRuleFire service from the **Service Name** drop-down list.
8. Select the FireRuleByName operation from the **Operations** drop-down list.
9. The **Endpoint URL** field is populated.

10. The **Parameters** table is populated with the `strRuleName` parameter. Type `projectname.alertname` as the value for this parameter.
11. Optionally select **Policies** by checking boxes to the left of the policy names.
12. Click **OK**.

## Sending Alerts to External Email Accounts

BAM can send alerts to external email accounts if an email driver is configured in Oracle User Messaging Service using Oracle Enterprise Manager Fusion Middleware Control.

See [Configuring Oracle User Messaging Service](#) for more information.

This feature is available only for actions involving email: **Send a dashboard link via email**, **Send a message via email**, and **Send a parameterized message**.

## Alert Events

Events launch the alert and trigger the action. Each alert contains only one event.

Oracle BAM provides the following events:

- [In a specific amount of time](#)
- [At a specific time today](#)
- [On a certain day at a specific time](#)
- [Every interval between two times](#)
- [Every date interval starting on certain date at a specific time](#)
- [Select a CQL Query](#)
- [When a data field in a data object meets specified conditions](#)
- [When this rule is launched](#)

### In a specific amount of time

When you select the event **In a specific amount of time**, you must select a time interval in seconds, minutes, or hours. You can select a query for the alert. This ensures that the action you specify when the alert is triggered only if the query yields results.

### At a specific time today

When you select the event **At a specific time today**, you must select the time at which to launch the alert.

### On a certain day at a specific time

When you select the event **On a certain day at a specific time**, you must select both the date and the time at which to launch the alert.

### Every interval between two times

When you select the event **Every interval between two times**, you must configure the following settings.

- **select time interval**  
Set the number of seconds, minutes, or hours between each alert launch.
- **select time**  
Set the times of day between which to launch the alert.

## Every date interval starting on certain date at a specific time

When you select the event **Every date interval starting on a certain date at a specific time**, you must configure the following settings.

- **select date interval**  
Set the alert to launch every Day, Week, Month, or Year.
- **select date**  
Set the date on which to launch the alert.
- **select time**  
Set the time of day at which to launch the alert.

## Select a CQL Query

When you select the event **Select a CQL Query**, you must select a continuous query from the pop-up drop-down list. The alert fires when the query receives new data with values outside of a normal range.

As an alternative, you can define an alert action in the continuous query itself. This alert action is specific to the query and does not appear in the left panel navigator under Alerts.

For more information about continuous queries, see [Creating a Continuous Query](#).

## When a data field in a data object meets specified conditions

When you select the event **When a data field in a data object meets specified conditions**, you must configure the following settings.

- **this data field has a condition of x**  
In the Data Based Event Editor, select the data object to monitor. Then you can set the condition under which the alert should fire.
  - **Row Filters** — Create a filter on a data field in the data object to express a condition that, when met, launches the alert. All of the functionality available in query filters is provided. See [Filtering Data](#) for more information.
  - **Group Filters** — The Group Filter is similar to the Row Filter in that it provides all of the filtering functionality available in query filters. The special feature here is that it allows filters to be created on a field where an aggregation function, such as AVG or SUM, has been applied. See [Filtering Data](#) for more information about building filter expressions.

 **Note:**

For aggregation functions with row-level operations, Oracle recommends that you use the **Group By** setting when you use a row filter. However, it is best to apply a Group filter instead. You cannot mix aggregation functions with row level operations without the Group By setting applied.

- **Groups** — Choose one or more data fields on which to create a grouping, adding further complexity to any filters created in the Row Filters or Group Filters tabs.
- run as **<user\_name>** (This option appears only if you, the user creating the alert, have an administrator role.)

Select the Oracle BAM user whom the alert runs as. You can select only one run-as user. The default run-as user is you, the logged-in Oracle BAM user who is creating the alert.

You can select a different run-as user if another user has greater permissions to access the data object being monitored. The run-as user can view the alert history only if that user has Read privileges on the alert and the project containing the alert. See [Securing an Alert](#) for more information.

Names preceded with a hash (#) are distribution lists and cannot be selected.

Only recipients with security permissions equal to or higher than the run-as user receive the alert notifications, honoring row-level security as implemented by the architect of the data object.

If there are changes in the data object rows that none of the alert recipients have permissions to access, no recipients are notified.

 **Note:**

The event **When a data field in a data object meets specified conditions** responds only to row inserts and updates. It does not respond to row deletes.

 **Note:**

Calculated fields are now supported within this alert action.

## When this rule is launched

The event **When this rule is launched** creates an alert that is dependent on another alert to launch it. The launching alert uses the **Launch a rule** action to launch the dependent alert. Several alerts can be chained using **When this rule is launched** in a hierarchy. For more information, see [Creating Dependent Alerts](#).

## Alert Conditions

Conditions are optional settings for constraining the time period in which the alert is fired.

You can select any number and combination of conditions. Oracle BAM provides the following conditions:

- [If it is between two times](#)
- [If It is between two days](#)
- [If it is a particular day of the week](#)

## If it is between two times

Select two times between which the alert should launch.

## If It is between two days

Select two dates between which the alert should launch.

## If it is a particular day of the week

Select a day of the week on which the alert should launch.

## Alert Actions

Actions are the results of the conditions and events of the alert expression having been met.

You can configure any number and combination of actions. Oracle BAM provides the following actions:

- [Send a dashboard link via email](#)
- [Send a message via email](#)
- [Delete rows from a data object](#)
- [Insert values into a data object](#)
- [Send a parameterized message](#)
- [Launch a rule](#)
- [Run a web service](#)
- [Launch a Custom Action](#)

## Send a dashboard link via email

Select a dashboard and select one or more recipients.

Recipients can be selected from BAM users or, if an email driver is configured in Oracle User Messaging Service using Oracle Enterprise Manager Fusion Middleware Control, from external email accounts. See [Sending Alerts to External Email Accounts](#) for more information.

**To create a dashboard link and send it using email:**

1. Create the alert. See [Creating an Alert](#).
2. Select the action **Send a dashboard link via email**.
3. Click either **<select dashboard>** or **<select user>** in the alert expression.  
The Send Dashboard Link via Email Editor opens.
4. Enter comma-separated email addresses in the **To** text box, or click the Search icon to display the Identity Browser and select BAM users.

5. Click the arrow to the right of **Select Project to List Dashboards** and select a project from the drop-down list.  
Dashboards in the selected project are listed.
6. Select a dashboard from the list.
7. Click **OK**.

## Send a message via email

Create an email message to send, optionally select a dashboard, and select one or more recipients.

Recipients can be selected from BAM users or, if an email driver is configured in Oracle User Messaging Service using Oracle Enterprise Manager Fusion Middleware Control, from external email accounts. See [Sending Alerts to External Email Accounts](#) for more information.

Messages can include variables that are set when the alert is launched, such as the time that an event occurred and the data that launched the event. To use data variables, the event must be based on data: **Select a CQL Query** or **When a data field in a data object meets specified conditions**.

**To create an alert that includes a message:**

1. Create the alert. See [Creating an Alert](#).
2. Select the action **Send a message via email**.
3. Click either **<create message>** or **<select user>** in the alert expression.  
The Message Editor opens.
4. Type comma-separated email addresses in the **To** text box, or click the Search icon to display the Identity Browser and select BAM users.
5. Type a subject in the **Subject** line.
6. Type the message in the **Message Text** box.
7. Optionally click the arrow to the right of **Select Project to List Dashboards** and select a project from the drop-down list.  
Dashboards in the selected project are listed.
8. Select a dashboard from the list.
9. Optionally include special fields in the message.

Special fields are listed in the **List** drop-down list. The special fields listed change when dashboards are selected in the lower portion of the Message Editor.

To insert a special field into the message:

- a. Select a special field from the list.
- b. Click **Insert into subject** or **Insert into text**.

You can insert multiple values of the same type, for example, multiple links to different dashboards.

- **Target User** inserts the names, BAM user IDs, or email addresses of the message recipients.

If you entered BAM users in the **To** text box, names are inserted for the users with names defined, otherwise the IDs are inserted. If you entered email addresses, these are inserted as is.

- **Date/Time Sent** inserts the date and time the message is sent.
- **Send Dashboard Name** inserts name of the selected dashboard.
- **Send Dashboard Owner** inserts the name of the owner of the selected dashboard.
- **Send Dashboard Link** inserts a link to the selected dashboard.

10. Click **OK**.

## Delete rows from a data object

Select the data object, and specify when a row is removed from the data object.

**To delete rows from a data object:**

1. Create the alert. See [Creating an Alert](#).
2. Select the action **Delete rows from a data object**.
3. Click **<this data field has a condition of x>** in the alert expression.  
The Purge Rows in Data Object Action Editor opens.
4. In the **Row Filters** tab, create a filter on a data field in the data object to express a condition that, when met, specifies a row to be deleted. All of the functionality available in query filters is provided. See [Filtering Data](#) for more information.
5. Click **OK**.

## Insert values into a data object

This action maps event data object fields to action data object fields and transfers values from one to the other. The data object that receives the inserted values is called a *write-back data object*. Select the action data object, and specify the event output fields that have values to be added to columns in the action data object.

This action is presented only if the event is based on data: **Select a CQL Query** or **When a data field in a data object meets specified conditions**.

[Figure 10-1](#) shows the Map Fields dialog with an example data insert operation.

**Figure 10-1** Map Fields Dialog

Type	Data Object Column	Event Output Field	Upsert Key
DATETIME	DATAOBJECT_CREATED		<input type="checkbox"/>
DATETIME	DATAOBJECT_MODIFIED		<input type="checkbox"/>
VARCHAR	region	region	<input type="checkbox"/>
VARCHAR	state		<input type="checkbox"/>
INT	sales		<input type="checkbox"/>

**To insert values into a data object:**

1. Create the alert. See [Creating an Alert](#).
2. Select the action **Insert values into a data object**.
3. Click **<select data object>** in the alert expression.  
The Map Fields dialog opens.
4. Select a data object from the drop-down list.  
The **Define Mappings** table is populated with the data object columns.
5. Select the **Operation Type**: insert or upsert.
6. For each column to receive values, select an **Event Output Field** from the drop-down list.
7. If the **Operation Type** is upsert, specify one or more columns as an **Upsert Key**.
8. Click **OK**.

## Send a parameterized message

This action emails dashboards that require parameter inputs to Oracle BAM users. You configure the email message and the parameter values that are passed to the dashboard.

For information about creating prompts and parameters in dashboards, see [Creating Parameters](#).

Create an email message to send, select a dashboard, select one or more recipients, and specify action and dashboard parameter values.

Recipients can be selected from BAM users or, if an email driver is configured in Oracle User Messaging Service using Oracle Enterprise Manager Fusion Middleware Control, from external email accounts. See [Sending Alerts to External Email Accounts](#) for more information.

This action is presented only if the event is based on data: **Select a CQL Query** or **When a data field in a data object meets specified conditions**.

[Figure 2](#) and [Figure 3](#) show the Message Editor with an example parameterized message.

Figure 10-2 Message Editor Displaying Action Parameter Values Tab

The screenshot shows a 'Message Editor' dialog box with the following sections:

- Subject(if email):** A text field containing 'Check Process Data'.
- Message Text:** A text area containing the message: 'Data in [Send Name of BPMAnalytics.ProcessDashboardForProcess] is out of range.'
- List:** A dropdown menu showing 'Send Dashboard Name', with 'Insert into Subject' and 'Insert into Text' buttons.
- Select Project To List Dashboards:** A dropdown menu.
- Dashboard List:** A list box containing 'Detail\_ProcessesFaultedToday', 'Detail\_ProcessesNotDueSoon', 'ProcessDashboardForProcess' (highlighted), and 'Detail\_ProcessesDueSoon'.
- Action Parameter Values:** A tabbed section with 'Report Parameter Values' selected. It includes:
  - User:** 'John Steinbeck' with a search icon and 'Select Data Object Field' dropdown.
  - Dashboard:** 'BPMAnalytics.ProcessDashboard' with a search icon and 'Select Data Object Field' dropdown.
  - Select Project To List Dashboards:** A dropdown menu.
  - Dashboard List:** A list box containing 'Detail\_ProcessesFaultedToday', 'Detail\_ProcessesNotDueSoon', 'ProcessDashboardForProcess' (highlighted), and 'Detail\_ProcessesDueSoon'.
- Buttons:** 'Ok' and 'Cancel' buttons at the bottom right.

Figure 10-3 Message Editor Displaying Report Parameter Values Tab

**Message Editor**

Subject(if email)  
Check Process Data

Message Text  
Data in [Send Name of BPMAalytics.ProcessDashboardForProcess] is out of range.

List  
Send Dashboard Name [v] [Insert into Subject] [Insert into Text]

Select Project To List Dashboards [v]

Detail\_ProcessesFaultedToday  
Detail\_ProcessesNotDueSoon  
ProcessDashboardForProcess  
Detail\_ProcessesDueSoon

Action Parameter Values **Report Parameter Values**

Add Parameter

Name	Type	Value
Select a Process	Text	Request Quote

[Ok] [Cancel]

**To create an alert that includes a parameterized message:**

1. Create the alert. See [Creating an Alert](#).
2. Select the action **Send a parameterized message**.
3. Click either **<create message>** or **<set parameters>** in the alert expression.  
The Message Editor opens.
4. Type a subject in the **Subject** line.
5. Type the message in the **Message Text** box.
6. Optionally click the arrow to the right of **Select Project to List Dashboards** and select a project from the drop-down list.  
Dashboards in the selected project are listed.
7. Select a dashboard from the list.
8. Optionally include special fields in the message.  
Special fields are listed in the **List** drop-down list. The special fields listed change when dashboards are selected in the middle portion of the Message Editor.  
To insert a special field into the message:
  - a. Select a special field from the list.
  - b. Click **Insert into subject** or **Insert into text**.

You can insert multiple values of the same type, for example, multiple links to different dashboards.

- **Target User** inserts the user names of the message recipients.
  - **Date/Time Sent** inserts the date and time the message is sent.
  - **Send Dashboard Name** inserts name of the selected dashboard.
  - **Send Dashboard Owner** inserts the name of the owner of the selected dashboard.
  - **Send Dashboard Link** inserts a link to the selected dashboard.
  - **Send Parameterized Dashboard Link** inserts a link to the selected dashboard that includes parameter values.
  - *data field* inserts the value of the specified field.
9. In the Action Parameter Values tab, populate the **User** text box in one of these ways:
    - Type comma-separated email addresses.
    - Click the Search icon to display the Identity Browser and select BAM users.
    - Click the arrow to the right of **Select Data Object Field** and select a data field with user name values.
  10. In the Action Parameter Values tab, populate the **Dashboard** text box in one of these ways:
    - Click the arrow to the right of **Select Data Object Field** and select a data field with dashboard name values.
    - Click the arrow to the right of **Select Project to List Dashboards** and select a project from the drop-down list.
  11. In the Report Parameter Values tab, click **Add Parameter** once for each parameter the dashboard requires.

A row is added to the parameters table for each click.

For special values use the underscore (\_), for example, `_ALL_`, `_BLANK_`, and `_NULL_`.
  12. In each row, type the parameter name in the **Name** field.
  13. In each row, select the parameter type in the **Type** field: Text or Data Object Field.
  14. For each parameter of type Text, type the parameter value in the **Value** field.
  15. For each parameter of type Data Object Field, click the **Value** field and select the parameter value from the drop-down list of data fields.
  16. Click **OK**.

 **Note:**

The **Send a parameterized message for every matching row in a data object** action is no longer supported. However, you can achieve the same result by combining the **When a data field in a data object meets specified conditions** event with the **Send a parameterized message** action.

## Launch a rule

The event **When this rule is launched** creates an alert that is dependent on another alert to launch it. The launching alert uses the **Launch a rule** action to launch the dependent alert. Several alerts can be chained using **When this rule is launched** in a hierarchy. For more information, see [Creating Dependent Alerts](#).

## Run a web service

Configure the web service that the alert calls when fired.

Configuring an alert to launch a web service, as this action does, is different from configuring a web service to launch an alert. For more information, see [Launching Alerts by Invoking Web Services](#).

[Figure 10-4](#) shows the Configure Web Service dialog with an example web service.

**Figure 10-4** Configure Web Service Dialog

Name	Type	Value
dataObject	Text	sampledo
xmlPayload	Text	<sampledo><region>R.1

For an example of both the **Run a web service** action and the ManualRuleFire web service, see [Alert and Web Service Example](#).

 **Note:**

Oracle BAM cannot determine if the web service is hosted on a server which is behind a secure server. It is your responsibility to indicate whether the web service is behind an HTTP basic authentication-based server. You must enter valid credentials if they are required.

To accomplish one-way SSL, the alert's web service client must be able to look up the certificate presented to it in a trust store.

**To configure the web service:**

1. Create the alert. See [Creating an Alert](#).
2. Select the action **Run a web service**.
3. Click **<configure web service>** in the alert expression.  
The Configure Web Service dialog opens.
4. Type the web service endpoint URL in the **WSDL or WSIL URL** field. The URL must begin with the HTTP protocol and must end in a valid extension (?WSDL, .WSDL or .WSIL).

For example:

```
http://host_name:port_number/OracleBAMWS/WebServices/DataObjectOperationsByID?WSDL
```

```
http://api.google.com/GoogleSearch.wsdl
```

```
http://host_name:port_number/inspection.wsil
```

5. If it is a secure web service, check the **Secure Web Service** box and enter the required **Username** and **Password** credentials.
6. Click **Display Operations** to display the available services of the web service.
7. Select a service from the **Service Name** drop-down list.
8. Select an operation from the **Operations** drop-down list.
9. The **Endpoint URL** field is populated after you type the WSDL/WSIL and credentials, get the list of operations, and select an operation.

It is populated with the endpoint URL defined in the WSDL file of the web service. If you find this endpoint URL outdated (for example, the web service implementation moved to a different endpoint but you do not have the new WSDL, but know the new endpoint URL) or incorrect, or want to override it, you can edit this URL. When the web service is invoked by Oracle BAM Event Engine, the configured endpoint URL is used to invoke the web service.

10. The **Parameters** table is also populated after you type the WSDL/WSIL and credentials, get the list of operations, and select an operation. Type an appropriate value for each parameter.
11. Optionally select **Policies** by checking boxes to the left of the policy names. The following policies are available:
  - oracle/wss\_username\_token\_client\_policy
  - oracle/log\_policy
  - oracle/wss\_http\_token\_client\_policy

12. Click **OK**.

**Note:**

If the web service does not respond to the call, then there are no logs available pertaining to the non-response or failure.

## Launch a Custom Action

If the actions provided in BAM don't meet your requirements, you can extend the action set.

### To configure a custom action:

1. Create a JAR file in which one of the Java classes implements this interface:

```
oracle.beam.common.alertsengine.externalactions.IAlertExternalAction
```

This interface declares the following methods, which the class must implement:

- `initialize()` — Executes the first time an alert that uses this action is triggered. The action is loaded into memory and cached internally.
- `takeAction()` — Executes whenever an alert that uses this action is triggered.

If the triggered alert is based on a data object change or a dashboard change, the changed row is passed to custom action as a `HashMap`. The `HashMap` key is a data object column name and the `HashMap` value is the value of the column in the changed row.

If the JAR file is modified, the BAM server must be restarted for the modifications to take effect. You must ensure that in a cluster environment, the JAR must be created on a shared resource that is accessible from any BAM node in the cluster.

2. Create the alert. See [Creating an Alert](#).
3. Select the action **Launch a Custom Action**.
4. Click **<custom action>** in the alert expression.  
The Configure Custom Action dialog opens.
5. Type a **Name** for the custom action.  
This name is added to the list of actions in the Action Editor if the action registration succeeds.
6. Optionally type a **Description**. This does not appear in the BAM console.
7. Type the **Library Name**, which is the full, absolute path of the JAR file, including the package name.
8. Type the **Class Name**, which is the full, absolute name of the Java class that implements the `IAlertExternalAction` interface, including the package name.
9. Type any **Arguments**, which are passed to the custom action code through the `initialize()` method.
10. Click **OK**.

# 11

## Creating Parameters

This chapter describes how to create parameters and prompts for queries, views, and dashboards in Oracle Business Activity Monitoring (BAM) to filter the data presented.

This chapter includes the following sections:

- [Understanding Parameters](#)
- [Parameter Prerequisites and Uses](#)
- [Creating a Value Parameter](#)
- [Creating a List Parameter](#)
- [Using a Parameter in a Query](#)
- [Editing a Parameter](#)
- [Deleting a Parameter](#)
- [Securing a Parameter](#)
- [Using Parameters in Dashboard URLs](#)
- [Driving Parameters from Other Views](#)
- [Using Prompts in Dashboards](#)

## Understanding Parameters

A *parameter* is a variable representing the value of a data field.

A *filter* is a condition applied to data retrieved by a query or view, which can reference one or more parameters. A *prompt* is a request for the user to specify the value of a parameter. For example, you can filter a product sales query by country, create a parameter for the country, and prompt the user to choose the country when the query is referenced by a view in a dashboard.

A parameter can have a default, which can be a specific value or a special value such as ALL or NULL. For example, a product sales view could show sales for all countries by default.

There are two types of parameters: value parameters and list parameters. A *value parameter* can accept only one value and does not explicitly reference a specific data field. A *list parameter* can present and in some cases accept multiple values and references a specific data field.

Users can use datetime parameters, including the system datetime, to limit data based on a time span. Other datetime values are based on the time zone selected in user preferences.

## Parameter Prerequisites and Uses

Parameter prerequisites are a set of tasks you must complete before you create a parameter, to ensure its smooth functioning.

Before you can create a list parameter, you must add to your project the data object containing the data field the parameter will represent. For more information about projects, see [Planning and Creating Projects](#). For more information about data objects, see [Working with Data Objects](#).

After you create parameters, you can use them in the following ways:

**In a query**

A query can reference parameters in filters. See [Creating Business Queries](#) for more information.

**In a view**

A view can reference a query that references parameters. However, you cannot do anything with a parameter in an individual view. See [Creating and Using Business Views](#) for more information.

**In a dashboard**

A dashboard can contain a view that references a query with parameters. The dashboard can prompt the user for the parameter values. See [Creating Dashboards](#) for more information.

**In a parameterized message alert action**

For queries or views that contain filters with parameters, only one alert action option is available. See [Send a parameterized message](#) for more information.

## Creating a Value Parameter

This section outlines the procedure to create a Value Parameter.

**To create a value parameter:**

1. Go to the **Designer** page.
2. Select **Parameters** in the left panel navigator, or right-click **Parameters** and select the **Create** menu item.

The Parameters dialog opens.

3. Type a **Name** and optionally edit the **Display Name**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the parameter is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

4. Select **Value Parameter** from the **Parameter Type** drop-down list, and click **Create**.

A tab opens for the new parameter.

5. On the parameter tab, specify the following **Parameter Definition** settings:
  - **Name** — The name you selected in the Parameters dialog appears here.
  - **Display Name** — The display name you selected in the Parameters dialog appears here, where you can change it.
  - **Description** — Type an optional description for the parameter. This description is only displayed when the parameter is edited.
  - **Data Type** — Select the data type: String (VARCHAR), Datetime, Integer, Decimal, or Float.

- **Required** — Check this box to require a value for this parameter. This value can be a default or a value chosen by the user.
- **Default Value** — You must select a default value if **Required** is not checked. If **Required** is checked, a default is optional.

Depending on the data type, the default value selection is different. The following options are displayed for all data types:

- An **Input Value** text box in which you type the default value.
- **ALL**, which returns all values.
- **NULL**, which returns null values.

The VARCHAR type also presents the user with **Blank** because these fields can be blank.

The DATETIME type presents the user with a **System Datetime** option and a **Select Date and Time** icon.

6. Click **Save**.

## Creating a List Parameter

This section outlines the procedure to create a List parameter.

**To create a list parameter:**

1. Go to the **Designer** page.
2. Select **Parameters** in the left panel navigator, or right-click **Parameters** and select the **Create** menu item.

The Parameters dialog opens.

3. Type a **Name** and optionally edit the **Display Name**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the parameter is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

4. Select **List Parameter** from the **Parameter Type** drop-down list, and click **Create**.

A tab opens for the new parameter.

5. On the parameter tab, specify the following **Parameter Definition** settings:
  - **Name** — The name you selected in the Parameters dialog appears here.
  - **Display Name** — The display name you selected in the Parameters dialog appears here, where you can change it.
  - **Description** — Type an optional description for the parameter. This description is only displayed when the parameter is edited.
  - **Data Type** — Select the data type: String (VARCHAR), Datetime, Integer, Decimal, or Float.
  - **Required** — Check this box to require a value for this parameter. This value can be a default or a value chosen by the user.
6. Specify the following **Value Options** settings:

- **Data Object** — Select the data object containing the data field the parameter will represent.
  - **Field** — Select the data field the parameter will represent. Only fields of the selected **Data Type** are displayed.
  - **Value Options** — Select an initial value to display to the user in a prompt for this parameter. This can be one of the default values or a different value.
  - **Choose Values** — Check this box to require the user to choose a value for a **Required** parameter with no default. This makes responding to a prompt mandatory. If **Required** is checked and a default value is preconfigured, the user need not choose a value.
  - **ALL** — Check this box to allow ALL as a value.
  - **NULL** — Check this box to allow NULL as a value.
  - **Blank** — Check this box to allow a VARCHAR type field to be blank.
  - **System Datetime** — Check this box to allow the system datetime as the value for a DATETIME field.
  - **User can select multiple values** — Check this box to allow the user to select more than one value for this parameter when using a dashboard in the Viewer.
7. Specify the following **Default Value** settings:
    - **Select Default Values** — You must select at least one default value if **Required** is not checked. If **Required** is checked, a default is optional. To select a value, check the box to the left of the value in the drop-down list.
  8. Click **Save**.

## Using a Parameter in a Query

To be used, a parameter must be part of a query filter. When a query uses a parameter, a dashboard that contains a view that references the query prompts the user for the parameter value.

**To reference a parameter in a query:**

1. Create a parameter. See [Creating a Value Parameter](#) or [Creating a List Parameter](#).
2. Create or edit a query. See [Creating Business Queries](#) for more information.
3. In the Filters panel of the query tab, click the **New Filter** icon.  
The Add/Edit Filter dialog appears.
4. Select the data field for the parameter from the drop-down list on the left.

### **Note:**

Some browser versions do not support the drop-down function that the Application Development Framework in this instance utilizes. This is a known issue.

5. Select the operation from the drop-down list in the center. The default operation is **is equal to**.
6. Select the radio button to the left of **Select a Parameter**.

The **Select a Parameter** option is activated, and the other radio button options are grayed.

7. Select the parameter from the drop-down list.

Value parameters of the same data type as the field you selected are included. List parameters based on the data field you selected are included.

8. Click **OK** to save the filter.
9. Click **Save** to save the changes to the query.

If you configured an initial or default value for the parameter, the Input Values dialog appears.

10. Click **OK** to accept the parameter input value.

A preview of the query uses the value you entered. See [Creating Business Queries](#) for more information.

## Editing a Parameter

When you edit a parameter, the changes propagate to all queries that reference the parameter and all views and dashboards that reference these queries.

Use the following procedure to open, edit, and save a parameter.

### To edit a parameter:

1. In the left panel navigator, click the arrow to the left of **Parameters**.  
All saved parameters in the current project are displayed in a list.
2. Click the parameter name, or right-click the parameter and select the **Edit** menu item.
3. Make the desired changes.
4. Click **Save**.

## Renaming a Parameter

This section outlines the procedure to rename a parameter.

Use the following procedure to change the **Display Name** of a parameter.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

### To rename a parameter:

1. In the left navigation pane, click the arrow to the left of **Parameters**.  
All saved parameters in the current project are displayed in a list.
2. To rename the parameter, right-click the parameter name and select the **Rename** menu item.  
The name becomes text in an editable field.
3. Type the new name and press **Enter**.  
The new name appears in the list.

When you edit the parameter, the new **Display Name** is displayed on the tab. This name is also displayed when you view the parameter in a dashboard.

However, the internal **Name** remains unchanged. When you edit the parameter, this name is displayed on the left in the header.

## Deleting a Parameter

If you delete a parameter, it is removed from any queries, views, or dashboards that reference it.

Use the following procedure to delete a parameter.

### To delete a parameter:

1. In the left panel navigator, click the arrow to the left of **Parameters**.  
All saved parameters in the current project are displayed in a list.
2. Click the parameter icon and click the **Delete** icon, or right-click the parameter name and select the **Delete** menu item.  
A dialog asks you to confirm the parameter deletion.
3. Click **OK**.  
The parameter disappears from the list.

## Securing a Parameter

A parameter inherits security settings from the project in which it is created.

For more information about projects, see [Planning and Creating Projects](#).

### To change security settings for a parameter:

1. In the left panel navigator, click the arrow to the left of **Parameters**.  
All saved parameters in the current project are displayed in a list.
2. Right-click the parameter and select **Security Settings** from the pop-up menu.  
The security tab for the parameter opens.
3. To add a role or group to whom you can explicitly grant or deny permissions, follow these steps:
  - a. Click the **Add** icon in the Grant Permissions or Deny Permissions table.  
The Add Application Roles, Groups, and Users dialog opens.  
See [Managing Oracle BAM Users](#) for information about how to add users to roles and groups.
  - b. Type a **Name** for the role or group you are adding.
  - c. Select from the drop-down **List**: Application Role or Group.
  - d. Click **Search** to populate the Available Members list.
  - e. To add a member to the Selected Members list, select the member and click the single right arrow.
  - f. To add all members to the Selected Members list, select the member and click the double right arrow.
  - g. To remove members from the Selected Members list, use the single and double left arrows.

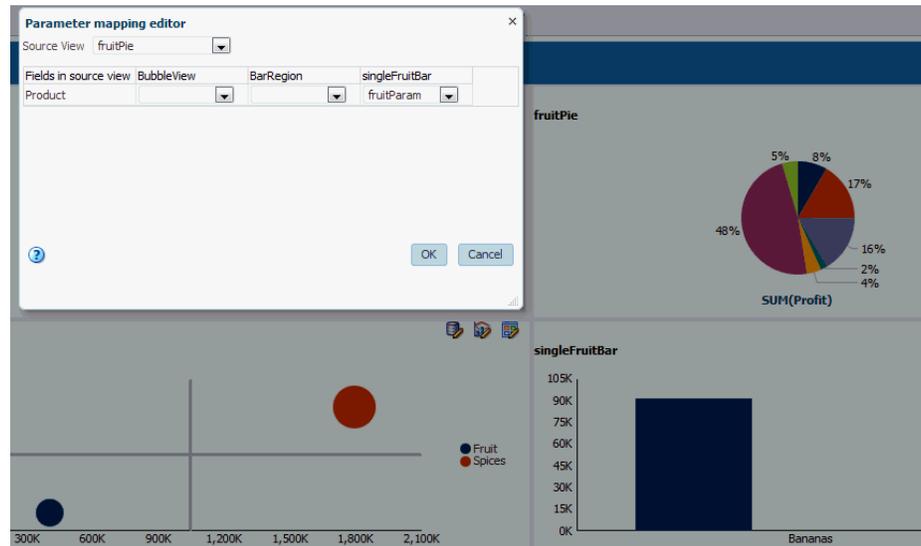


To be driven, a view must reference a query with a parameter used in a filter. Also, if drilling is enabled, driving will not work. Drilling overrides driving.

The view that drives the other views, the *driver*, need not reference a query with a parameter.

Figure 1 shows a dashboard with the Parameter Mapping Editor open. The Product data field value in the pie chart drives the fruitParam parameter value in the bar chart.

**Figure 11-1 Parameter Driving Example**



For another example of a dashboard that uses parameter driving, see [BPM Process View Dashboard](#).

**To configure driving in a dashboard:**

1. Click the **Parameter Mapping Editor** icon in the upper right corner of the dashboard tab. The Parameter Mapping Editor opens.
2. Select the view that will drive other views from the **Source View** drop-down list.
3. In the parameter mapping table, map the Source View data fields in the first column to the views in the remaining columns. Each cell in the table has a drop-down list of the parameters available for that view of the same data type as the Source View field.
4. Click **OK** to save the parameter mapping.
5. Click **Save** to save the dashboard.
6. Open the dashboard in the BAM Viewer (the Home page), in one of these ways:
  - On the Catalog tab, double-click a dashboard name.
  - On the Catalog tab, select a dashboard name and click the **Open** icon.
  - Under **Most Recent** or **Favorites**, click on a dashboard name if **List View** is selected or a thumbnail if **Carousel View** is selected.

Observe how values in the source view are reflected in the driven views.

 **Note:**

Drilling and driving operations can only be supported in the View Mode, and not in the Designer Mode. For new chart visualizations, driving is supported through the right click option. For graphs, no such menu option is available. You can click the graph element directly.

## Using Prompts in Dashboards

When you open a dashboard in the BAM Viewer (the Home page), prompts for the associated parameters are displayed in the right panel.

 **Note:**

Dashboard layout and size, label lengths, field lengths, and the number of prompts must be designed appropriately for prompts to be fully visible and operable. If there is not room on screen to display them all for a given report, you won't be able to see all the prompts.

### To use a prompt for a parameter:

1. Open a dashboard in one of these ways:
  - On the Catalog tab, double-click a dashboard name.
  - On the Catalog tab, select a dashboard name and click the **Open** icon.
  - Under **Most Recent** or **Favorites**, click on a dashboard name if **List View** is selected or a thumbnail if **Carousel View** is selected.
2. If the right panel is not visible, display it by clicking the arrow in the middle of the right edge of the dashboard.

The parameters associated with the dashboard are displayed in the right panel.

3. Change the parameter values as needed.

Each value parameter has a text box in which you enter a value. Each list parameter has a drop-down list from which you select a value.

The dashboard display changes each time you press Enter in a text box or select a new value from a drop-down list.
4. To revert to the parameter defaults, close and reopen the dashboard.

# Part III

## Integrating Oracle Business Activity Monitoring with Other Applications

This part describes how to integrate Oracle Business Activity Monitoring with Oracle Business Process Management and Oracle SOA Suite.

This part includes the following chapters:

- [Integrating with Oracle Business Process Management](#)
- [Integrating with Oracle SOA Suite](#)

# 12

## Integrating with Oracle Business Process Management

This chapter describes how Oracle Business Activity Monitoring (Oracle BAM) integrates Oracle Business Process Management (Oracle BPM). This integration captures BPM process metrics for analysis and display in BAM.

This chapter includes the following sections:

- [Understanding Oracle BAM and BPM Integration](#)
- [Configuring Cross-domain Communication for BAM, SOA, and BPM](#)
- [Enabling Process Metrics](#)
- [Understanding the Pre-assembled BPM Dashboards](#)
- [Understanding the Oracle BAM Data Objects for BPM](#)

### Understanding Oracle BAM and BPM Integration

When BAM and BPM are installed together and after process metrics are enabled, the BAM data objects for BPM are automatically created when BPM processes are deployed and populated with metric data as BPM processes run.

These data objects handle data from BPM, Case Management, and Human Workflow.

The data generated by BPM is used in pre-created process analytics reports. As a BAM designer you can also choose to include them into a custom BAM project.

For ready-to-use operational monitoring, BAM provides a set of pre-assembled dashboards in the default Process Analytics project.

Because each BPM process is unique, some instrumentation is required on the BPM side in BPM Composer or Oracle JDeveloper to capture the metric data in the BAM data objects. See *Using Process Analytics in Developing Business Processes for Oracle Business Process Management* for more information.

### Configuring Cross-domain Communication for BAM, SOA, and BPM

Oracle BAM can communicate with Business Process Management across domains.

This configuration is only supported for one BPM or SOA domain to one BAM domain. We do NOT support multiple separate BPM or SOA domains feeding into one BAM domain. This task assumes that:

- BPM and BAM domains are configured
- Managed servers on both nodes are configured and have different names
- The BAM server is still configured in the BPM (or SOA) domain, even if there is no data yet.

The high-level configuration steps for cross-domain configuration for BAM and BPM are:

1. From the Oracle Fusion Middleware Control console, start ALL managed servers on both nodes. See [Starting the Managed Servers](#) for more information.
2. Access the **AnalyticsConfig** folder for each node in the **System MBean Browser** and enable analytics for both nodes. See [Enabling Process Metrics](#) for more information.
3. On the Weblogic Server Administration Console, access the SOA domain from the **Domain Structure** panel.
4. Click the **Security** tab.
5. In the **Advanced** section, specify a value in the **Credential** field. Ensure that you specify the same value in the **Confirm Credential** field.
6. Click **Activate Changes** in the **Change Center** panel.
7. Next, follow the instructions in step 3 through 6 for the SOA and BAM domain.  
Global Trust is now enabled on both nodes.
8. In the Domain Structure panel, click the Services item to access **BAMForeignJndiProvider** settings.
9. On Node 1 (SOA/BPM Domain), point BAMForeignJndiProvider to Node 2 (BAM Domain with SOA Server).
10. Enter the username and password for Node 2.
11. On Node 2, point BPMForeignJndiProvider to Node 1.
12. Enter the username and password for Node 1.
13. Shut down all managed and Admin servers on both nodes.
14. On Node 1, start the SOA and Admin Servers.
15. On Node 2, start BAM, SOA, and Admin Servers.
16. Deploy the BPM application to Node 1.
17. Compare the projects listed in the SOA only domain with the projects listed under the SOA and BAM domain and check for consistency.

You can test this configuration by accessing the Business Process Workspace to initiate a new process. If the configuration is successful, the Process Summary Dashboard in BAM Composer will display it.

## Enabling Process Metrics

This section outlines the procedure for enabling process metrics.

The steps for enabling BPM and Business Process Expression Language (BPEL) process metrics are the same.

 **Note:**

If a BPM or BPEL process configured with measurements is deployed before enabling process metrics or while the Oracle BAM server is shut down, then all data published to BAM from such processes is disabled even if BAM is restarted later. This is because some mandatory artifacts required for runtime analytics cannot be created unless BAM is running. To allow such processes to publish data to BAM, redeploy them when BAM is running.

Performing process cubes migration before enabling process analytics is strongly recommended. See *Migrate Process Cubes to BAM 12c Process Star Schema* in *Upgrading Oracle SOA Suite and Business Process Management* for more information.

**To enable process metrics:**

1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the Weblogic Domain node.
3. Select the domain in which the Oracle BAM server is installed.  
For example, the domain might be `soainfra` or `base_domain`.
4. Right-click on the domain and select **System MBean Browser**.  
The System MBean Browser page appears.
5. In the System MBean Browser, expand the Application Defined MBeans node.
6. Under Application Defined MBeans, expand the `oracle.as.soainfra.config` node.
7. Under `oracle.as.soainfra.config`, expand the `Server: server_name` node.
8. Under `Server: server_name`, expand the **AnalyticsConfig** node.
9. Under **AnalyticsConfig**, click **analytics**.  
The analytics attributes are listed.
10. Change the value of the **DisableProcessMetrics** attribute to `false`.
11. Click **Apply**.  
If no BAM server is detected, an error message appears and the **DisableProcessMetrics** attribute value is not changed.

## Understanding the Pre-assembled BPM Dashboards

The pre-assembled dashboards in the Process Analytics project show common types of process analysis.

This section describes the dashboards and their component parts, including business views, business queries, and parameters. [Understanding the Oracle BAM Data Objects for BPM](#) describes the referenced data objects.

The Process Analytics Dashboards dashboard is directly under the Dashboards category in the navigator pane. All other pre-assembled dashboards are in the `Process Analytics` folder.

You can customize the metrics, queries, and views created for these dashboards for your company's specific needs.

See [Creating Dashboards](#) for details on how to create dashboards.

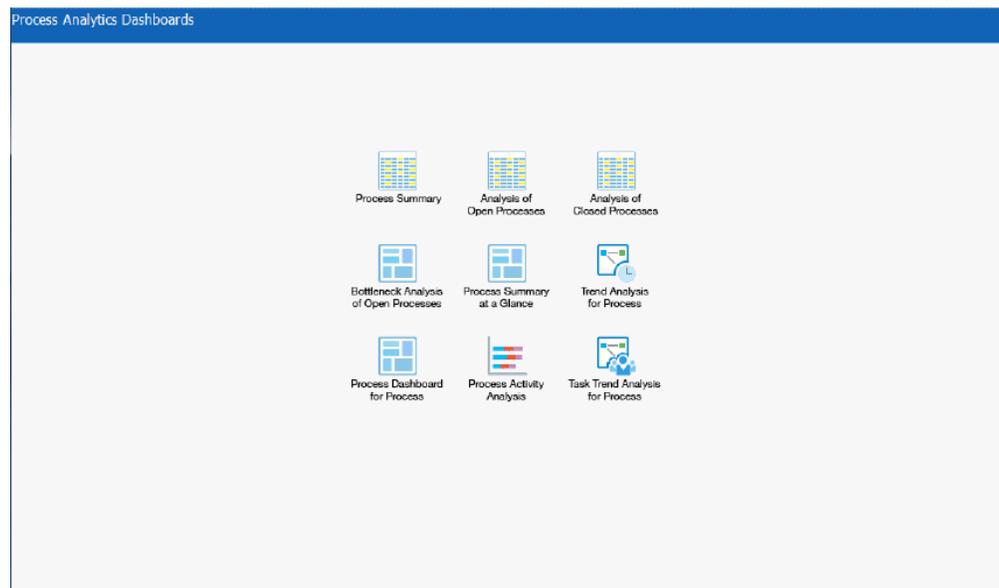
This section includes the following topics:

- [Process Analytics Dashboards Dashboard](#)
- [Process Summary Dashboard](#)
- [Analysis of Open Processes Dashboard](#)
- [Analysis of Closed Processes Dashboard](#)
- [Bottleneck Analysis of Open Processes Dashboard](#)
- [Process Summary at a Glance Dashboard](#)
- [Trend Analysis for Processes Dashboard](#)
- [Process Dashboard for Process Dashboard](#)
- [Process Activity Analysis Dashboard](#)
- [Task Trend Analysis for Process Dashboard](#)
- [BPM Process View Dashboard](#)
- [BPM Dashboard Parameters](#)

## Process Analytics Dashboards Dashboard

The Process Analytics Dashboards dashboard links to all the other pre-assembled BPM dashboards except the BPM Process View dashboard. Each of the nine non-empty cells contains an image file and a dashboard link.

**Figure 12-1 Process Analytics Dashboards Dashboard**



See [Adding Other Content to Cells](#) for information about how to add other content to dashboard cells, including images and links to other dashboards.

## Data Visibility Rules for the Process Analytics Dashboards Dashboard

You can access dashboard data based on your assigned application role. Not all application roles are privy to all levels of data. Oracle BAM ships with the following application roles:

- **Process Owner:** When a BPM composite is deployed, a corresponding process owner role <Composite Name>.ProcessOwner is automatically created in OracleBPMProcessRolesApp application stripe.
- **BPMProcessAdmin:** A seeded application role in OracleBPMProcessRolesApp application stripe. The 'Administrators' group is a member of BPM ProcessAdmin application role.
- **BPMContentViewer:** A seeded application role in the "BAMServer" application stripe.
- **Analytics viewer:** When a BPM Composite with analytics enabled is deployed, a corresponding composite analytics viewer role <Composite Name>.AnalyticsViewer is automatically created in the "BAMServer" application stripe.
- **BAMAdministrator:** A seeded application role in the "BAMServer" application stripe. The 'Administrators' group is a member of BAM Administrator application role.

Table 11-1 is depicts the data visibility for all dashboards for all memberships.

**Table 12-1 Data Visibility Rules for BAM, BPM, and OOTB Process Analytics Dashboards Dashboard**

<b>Membership</b>	<b>BAM Process View 14c Dashboard</b>	<b>BPM, BAM 14c OOTB Process Analytics Dashboards (Except Process View), AND Any Custom BAM 14c Dashboard using OOTB or Composite Specific Process Star Schema Logical DO.</b>
BPM Process Admin Application Role	Can see all data	Not relevant
Composite's ProcessOwner Application Role	Restricted to processes for which the user has the process owner role	Not relevant
BPM ContentViewer Application Role	Not relevant	Can see all data
Composite's AnalyticsViewer Application Role	Not relevant	Restricted to processes for which the user has the corresponding Composite AnalyticsViewer role
BAMAdministration Application Role	Not relevant	Can see all data
Administrator Group	Can see all data	Can see all data
All Other Authenticated Users	No data shown	No data shown

## Process Summary Dashboard

The Process Summary dashboard is a table display of all processes and their metrics for opened, closed, due soon, overdue, and faulted statuses. For due soon, overdue, and faulted statuses, graphics are displayed for values over 115, 18, and 20 process instances respectively.

**Figure 12-2 Process Summary Dashboard**

Application Name		Process Summary			Opened Today	Closed Today	Closure % Today	Faulted
Process Name	Not Due Soon	Due Soon	Overdue					
Business Account Opening	Open Business	153	106	15	12	18	0.19%	30
Debt Collection	Collect Debt	20	33	4	2	1	0.04%	3
Employee Expense Approval	Approve Expense	21	22	6	4	6	0.05%	1
Employee Onboarding	Employee Onboarding	146	111	17	8	16	0.15%	17
Incident Reporting	Report Incident	104	92	13	14	11	0.12%	13
Internal Service Request	Request Service	19	25	1	3	2	0.04%	6
Lean Processing	Process Loan	145	100	9	10	10	0.14%	16
Sales Quote	Request Quote	166	86	29	14	12	0.16%	22
Travel Request Management	Request Travel	111	69	16	13	8	0.12%	15

The Title properties of the dashboard are Arial font, size 5, bold.

The Process Summary dashboard contains or references the following entities:

- [Process Summary View](#)
- [Process Summary Query](#)

## Process Summary View

The Process Summary view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Process Summary Query](#)
- Thresholds properties:
  - SUMIS\_OPN\_NOT\_DUE\_SOON, 0-10000000, alta\_green.png graphic
  - SUMIS\_DUE\_SOON, 0-10000000, dark\_yellow.png graphic
  - SUMIS\_OVERDUE, 0-10000000, alta\_orange.png graphic
  - SUMIS\_FAULTED
    - \* 0–19, no graphic
    - \* 19-10000000, warning.png graphic
- Text properties:
  - Title Text — Process Summary, Arial font, size 18, bold, center aligned
  - Table Text — Arial font, size 16, center aligned
  - Column Header Text — Arial font, size 15, bold, center aligned
  - Column Header Display Text:
    - \* COMPOSITE\_NAME — Application Name

- \* PROCESS\_LABEL — Process Name
- \* SUMIS\_OPENED\_TODAY — Is Opened Today
- \* SUMIS\_CLOSED\_TODAY — Is Closed Today
- \* PERCENTOFTOTALIS\_CLOSED — Closure % Today
- \* SUMIS\_NOT\_DUE\_SOON — Open
- \* SUMIS\_DUE\_SOON — Due Soon
- \* SUMIS\_OVERDUE — Overdue
- \* SUMIS\_FAULTED — Faulted
- PERCENTOFTOTAL (IS\_CLOSED) has **Format Field As** set to Percentage

## Process Summary Query

The Process Summary query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [Process Data Object](#)
- The following measures and aggregations are selected:
  - Is Opened Today, SUM
  - Is Closed Today, SUM
  - Is Open (and Not Due Soon), SUM
  - Is Overdue, SUM
  - Is Closed, PERCENTOFTOTAL
  - Is Due Soon, SUM
  - Is Faulted, SUM
- The following dimensions are selected:
  - Composite Name
  - Process Display Name

## Analysis of Open Processes Dashboard

The Analysis of Open Processes dashboard is a table display of open processes that compares average cycle time with current open time. For overdue processes and maximum process open times, graphics are displayed for values over 30 process instances and 23 days respectively.

**Figure 12-3 Analysis of Open Processes Dashboard**

Application Name		Process Name		Analysis of Open Processes			
Application Name	Process Name	Open	Overdue	Average Open Time (days)	Minimum Open Time (days)	Maximum Open Time (days)	Estimated Cycle Time (days)
Business Account Opening	Open Business	274	15	6.5	0.0	20.0	6.0
Debt Collection	Collect Debt	57	4	5.7	0.5	13.3	5.2
Employee Expense Approval	Approve Expense	49	6	5.3	0.3	11.0	4.7
Employee Onboarding	Employee Onboarding	274	17	7.6	0.0	21.9	7.1
Incident Reporting	Report Incident	206	13	7.1	0.0	23.2	6.6
Internal Service Request	Request Service	45	1	5.1	0.5	11.3	4.6
Loan Processing	Process Loan	246	9	7.3	0.1	20.6	6.9
Sales Quote	Request Quote	273	28	7.8	0.1	25.0	7.3
Travel Request Management	Request Travel	190	10	7.0	0.0	22.7	6.5

The Title properties of the dashboard are Arial font, size 5, bold.

The Analysis of Open Processes dashboard contains or references the following entities:

- [Analysis of Open Processes View](#)
- [Analysis of Open Instances Query](#)

## Analysis of Open Processes View

The Analysis of Open Processes view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Analysis of Open Instances Query](#)
- Thresholds properties:
  - SUMIS\_OPEN, 0–10000000, alta\_green.png graphic
  - SUMIS\_OVERDUE, 0–10000000, alta\_orange.png graphic
  - MAXPROCESS\_OPEN\_TIME\_IN\_DAYS
    - \* 0–23, no graphic
    - \* 23–10000000, warning\_circle\_22x16.png graphic
- Text properties:
  - Title Text — Analysis of Open Processes, Arial font, size 18, bold, center aligned
  - Table Text — Arial font, size 16, center aligned
  - Column Header Text — Arial font, size 15, bold, center aligned
  - Column Header Display Text:
    - \* COMPOSITE\_NAME — Application Name
    - \* PROCESS\_LABEL — Process Name

- \* SUMIS\_OPEN — Is Open
- \* SUMIS\_OVERDUE — Is Overdue
- \* AVGESTIMATED\_COMPLETE\_TIME\_DAYS — Estimated Cycle Time (days)
- \* AVGPROCESS\_OPEN\_TIME\_IN\_DAYS — Average Open Time (days)
- \* MINPROCESS\_OPEN\_TIME\_IN\_DAYS — Minimum Open Time (days)
- \* MAXPROCESS\_OPEN\_TIME\_IN\_DAYS — Maximum Open Time (days)
- The following columns have **Format Field As** set to Number and **No of Digits After Decimal** set to 1:
  - \* AVG(ESTIMATED\_COMPLETE\_TIME\_DAYS)
  - \* AVG(PROCESS\_OPEN\_TIME\_IN\_DAYS)
  - \* MIN(PROCESS\_OPEN\_TIME\_IN\_DAYS)
  - \* MAX(PROCESS\_OPEN\_TIME\_IN\_DAYS)

## Analysis of Open Instances Query

The Analysis of Open Instances query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [Process Data Object](#)
- The following measures and aggregations are selected:
  - Is Open, SUM
  - Is Overdue, SUM
  - Process Open Time in Days, AVG, MIN, MAX
  - Estimated Completion Time in Days, AVG
- The following dimensions are selected:
  - Composite Name
  - Process Display Name
- Filters — **All are true** branch:
  - IS\_OPEN is equal to '1'

## Analysis of Closed Processes Dashboard

The Analysis of Closed Processes dashboard is a table display that compares the current process cycle time and estimated cycle time with historical data, which you can filter by date.

**Figure 12-4 Analysis of Closed Processes Dashboard**

Application Name		Process Name	Closed	Missed Due Date	Average Estimated Cycle Time (days)	Average Process Cycle Time (days)	Minimum Process Cycle Time (days)	Maximum Process Cycle Time (days)
Business Account Opening	Open Business	16	11	11.9	12.2	7.5	20.3	
Debt Collection	Collect Debt	1	0	9.0	9.3	9.3	9.3	
Employee Expense Approval	Approve Expense	6	5	9.7	9.9	7.1	14.5	
Employee Onboarding	Employee Onboarding	16	6	12.0	12.4	8.6	16.6	
Incident Reporting	Report Incident	11	7	11.6	11.8	7.5	29.1	
Internal Service Request	Request Service	2	1	8.5	8.7	8.2	9.2	
Loan Processing	Process Loan	10	5	12.2	12.5	6.6	18.3	

Application Name	Process Name	Total Closed	Average Process Cycle Time (days)
Business Account Opening	Open Business	113	12.1
Debt Collection	Collect Debt	92	9.7
Employee Expense Approval	Approve Expense	112	9.6
Employee Onboarding	Employee Onboarding	331	12.7

The Title properties of the dashboard are Arial font, size 5, bold.

The Analysis of Closed Processes dashboard contains or references the following entities:

- [Analysis of Processes Closed Today View](#)
- [Historical Process Completion Times View](#)
- [Analysis of Instances Closed Today Query](#)
- [Historical Process Completion Times Query](#)

## Analysis of Processes Closed Today View

The Analysis of Processes Closed Today view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Analysis of Instances Closed Today Query](#)
- Thresholds properties:
  - SUMIS\_CLOSED, 0–10000000, alta\_green.png graphic
  - SUMMISSED\_DUE\_DATE, 0–10000000, alta\_orange.png graphic
  - MAXPROCESS\_CYCLE\_TIME\_IN\_DAYS
    - \* 0–25, no graphic
    - \* 25–10000000, warning\_circle\_22x16.png graphic
- Text properties:
  - Title Text — Analysis of Closed Processes, Arial font, size 18, bold, center aligned
  - Table Text — Arial font, size 16, center aligned
  - Column Header Text — Arial font, size 15, bold, center aligned
  - Column Header Display Text:

- \* COMPOSITE\_NAME — Application Name
- \* PROCESS\_LABEL — Process Name
- \* SUMIS\_CLOSED — Is Closed
- \* SUMMISSED\_DUE\_DATE — Missed Due Date
- \* AVGESTIMATED\_COMPLETE\_TIME\_DAYS — Estimated Cycle Time (days)
- \* AVGPROCESS\_CYCLE\_TIME\_IN\_DAYS — Average Cycle Time (days)
- \* MINPROCESS\_CYCLE\_TIME\_IN\_DAYS — Minimum Cycle Time (days)
- \* MAXPROCESS\_CYCLE\_TIME\_IN\_DAYS — Maximum Cycle Time (days)
- The following columns have **Format Field As** set to Number and **No of Digits After Decimal** set to 1:
  - \* AVG(ESTIMATED\_COMPLETE\_TIME\_DAYS)
  - \* AVG(PROCESS\_CYCLE\_TIME\_IN\_DAYS)
  - \* MIN(PROCESS\_CYCLE\_TIME\_IN\_DAYS)
  - \* MAX(PROCESS\_CYCLE\_TIME\_IN\_DAYS)

## Historical Process Completion Times View

The Historical Process Completion Times view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Historical Process Completion Times Query](#)
- Text properties:
  - Title Text — Analysis of Closed Processes, Arial font, size 18, bold, center aligned
  - Table Text — Arial font, size 16, center aligned
  - Column Header Text — Arial font, size 15, bold, center aligned
  - Column Header Display Text:
    - \* COUNTX — Total Closed
    - \* COMPOSITE\_NAME — Application Name
    - \* PROCESS\_LABEL — Process Name
    - \* AVGPROCESS\_CYCLE\_TIME\_IN\_DAYS — Average Cycle Time (days)
  - The following columns have **Format Field As** set to Number and **No of Digits After Decimal** set to 1:
    - \* AVG(PROCESS\_CYCLE\_TIME\_IN\_DAYS)

## Analysis of Instances Closed Today Query

The Analysis of Instances Closed Today query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [Process Data Object](#)

- The following measures and aggregations are selected:
  - Is Closed, SUM
  - Missed Due Date, SUM
  - Estimated Completion Time in Days, AVG
  - Process Cycle Time in Days, AVG, MIN, MAX
- The following dimensions are selected:
  - Composite Name
  - Process Display Name
- Filters — **All are true** branch:
  - `IS_CLOSED_TODAY` is equal to '1'
  - `PROCESS_INSTANCE_STATUS` is equal to 'COMPLETED'

## Historical Process Completion Times Query

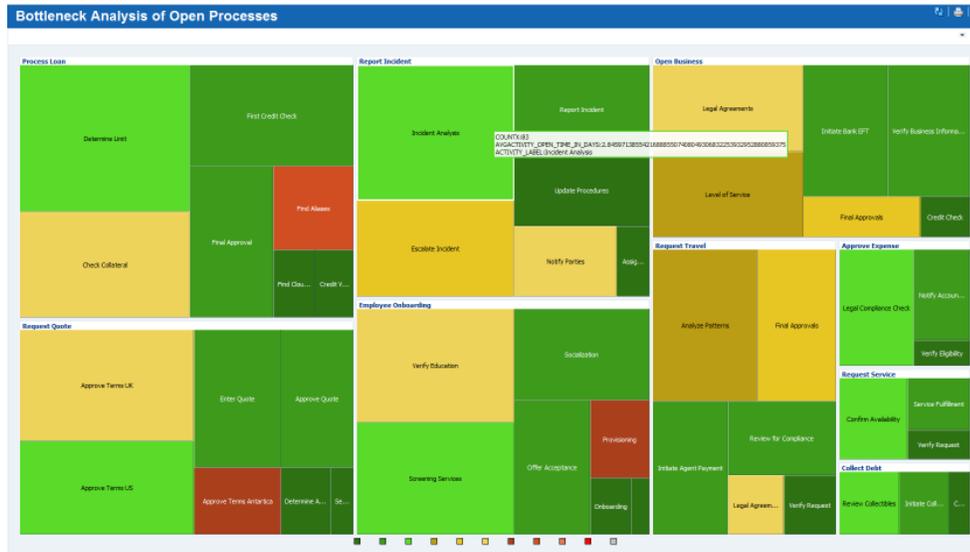
The Historical Process Completion Times query has the following characteristics and changes from default settings:

- Type — Group SQL Query
  - Data object — see [Process Data Object](#)
  - The following measures and aggregations are selected:
    - Process Cycle Time in Days, AVG
    - COUNT(\*)
  - The following dimensions are selected:
    - Composite Name
    - Process Display Name
  - Filters — **All are true** branch:
    - `PROCESS_END_TIME` is greater than or equal to \$Select data after
    - `PROCESS_END_TIME` is less than or equal to \$Select data before
    - `PROCESS_INSTANCE_STATUS` is equal to 'COMPLETED'
- See [Select data after Parameter](#) and [Select data before Parameter](#)

## Bottleneck Analysis of Open Processes Dashboard

The Bottleneck Analysis of Open Processes dashboard is a treemap that uses size and color to identify long-running activities with high volume.

**Figure 12-5 Bottleneck Analysis of Open Processes Dashboard**



The Title properties of the dashboard are Arial font, size 5, bold.

The Bottleneck Analysis of Open Processes dashboard contains or references the following entities:

- [Bottleneck Analysis of Open Processes View](#)
- [Bottleneck Analysis of Open Processes Query](#)

## Bottleneck Analysis of Open Processes View

The Bottleneck Analysis of Open Processes view has the following characteristics and changes from default settings:

- Type — Treemap
- Query — see [Bottleneck Analysis of Open Processes Query](#)
- Value — COUNT(\*)
- Color — AVG(ACTIVITY\_OPEN\_TIME\_IN\_DAYS)
- Title — (blank)
- Color Thresholds properties:
  - 0–1, 2d7113 hex color
  - 1–2, 3d9a1b hex color
  - 2–3, 5ada29 hex color
  - 3–4, ba9d14 hex color
  - 4–5, e7c624 hex color
  - 5–6, edd35a hex color
  - 6–7, a93f1b hex color
  - 7–8, fefaf9 hex color
  - Overflow, e47752 hex color

## Bottleneck Analysis of Open Processes Query

The Bottleneck Analysis of Open Processes query has the following characteristics and changes from default settings:

- Type — Tree Model Query
- Data object — see [Activity Data Object](#)
- The following measures and aggregations are selected:
  - Activity Open Time in Days, AVG
  - COUNT(\*)
- Hierarchy — ProcessActivityHierarchy
- Filters — **All are true** branch:
  - ACTIVITY\_START\_TIME is less than or equal to \$Select data before
  - ACTIVITY\_START\_TIME is greater than or equal to \$Select data after
  - ACTIVITY\_INSTANCE\_STATUS is equal to 'ACTIVE'
  - ACTIVITY\_OPEN\_TIME\_IN\_HOURS is greater than or equal to '0'

See [Select data before Parameter](#) and [Select data after Parameter](#)

## Process Summary at a Glance Dashboard

The Process Summary at a Glance dashboard is a treemap that uses size and color to show the number of active, completed, and faulted activities. You can filter the data by date.

**Figure 12-6 Process Summary at a Glance Dashboard**



The Title properties of the dashboard are Arial font, size 5, bold.

The Process Summary at a Glance dashboard contains or references the following entities:

- [Process Summary at a Glance View](#)
- [Process Summary at a Glance Query](#)

## Process Summary at a Glance View

The Process Summary at a Glance view has the following characteristics and changes from default settings:

- Type — Treemap
- Query — see [Process Summary at a Glance Query](#)
- Value — COUNT(\*)
- Color — FirstInGroup(PROCESS\_INSTANCE\_STATUS\_CODE)
- Title — Process Summary at a Glance, Arial font, 11
- Color Thresholds properties:
  - 0.5–1.5, 6db060 hex color
  - 1.5–2.5, 587fb8 hex color
  - 2.5–3.5, a93f1b hex color

## Process Summary at a Glance Query

The Process Summary at a Glance query has the following characteristics and changes from default settings:

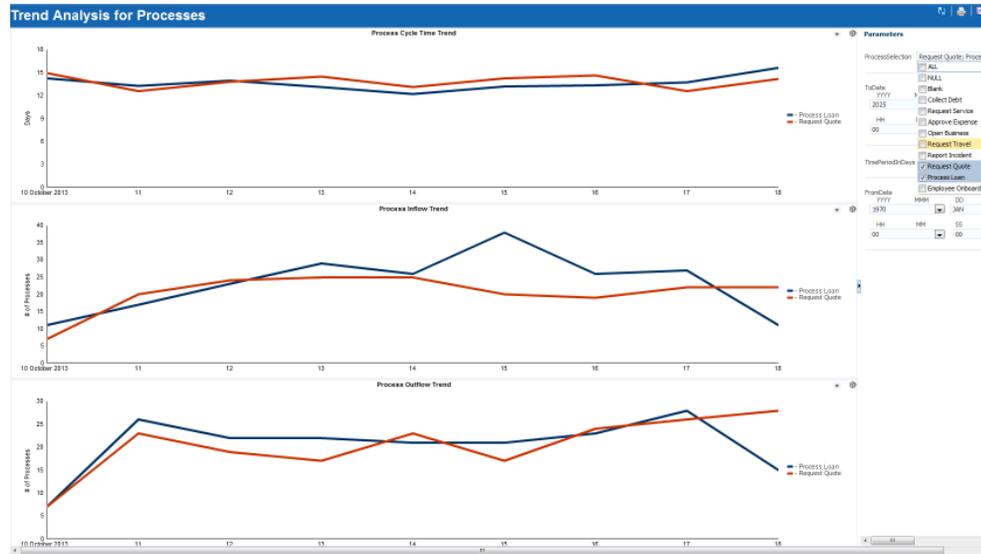
- Type — Tree Model Query
- Data object — see [Process Data Object](#)
- The following measures and aggregations are selected:
  - Process Status Instance Code, FirstInGroup
  - COUNT(\*)
- Hierarchy — CompositeProcessStatusHierarchy
- Filters — **All are true** branch:
  - PROCESS\_END\_TIME is less than or equal to \$Select data before
  - PROCESS\_END\_TIME is greater than or equal to \$Select data after

See [Select data before Parameter](#) and [Select data after Parameter](#)

## Trend Analysis for Processes Dashboard

The Trend Analysis for Processes dashboard is a set of line graphs that displays process cycle time, inflow, and outflow by day. You can filter the data by date and process.

**Figure 12-7 Trend Analysis for Processes Dashboard**



The Title properties of the dashboard are Arial font, size 5, bold.

The Trend Analysis for Processes dashboard contains or references the following entities:

- [Process Cycle Time Trend View](#)
- [Process Inflow Trend View](#)
- [Process Outflow Trend View](#)
- [Process Cycle Time Trend Query](#)
- [Process Inflow Trend Query](#)
- [Process Outflow Trend Query](#)

## Process Cycle Time Trend View

The Process Cycle Time Trend view has the following characteristics and changes from default settings:

- Type — Line
- Query — see [Process Cycle Time Trend Query](#)
- Title properties:
  - Title Text — Process Cycle Time Trend, Arial font, 11, center aligned
- X-Axis properties:
  - Tick Label — Arial font, size 11
- Y-Axis properties:
  - Title — Days
  - Tick Label — Arial font, size 11
- Label Text:
  - PROCESS\_END\_TIME — End Date

- AVG(PROCESS\_CYCLE\_TIME\_IN\_DAYS) — '-'

## Process Inflow Trend View

The Process Inflow Trend view has the following characteristics and changes from default settings:

- Type — Line
- Query — see [Process Inflow Trend Query](#)
- Title properties:
  - Title Text — Process Inflow Trend, Arial font, 11, center aligned
- X-Axis properties:
  - Tick Label — Arial font, size 11
- Y-Axis properties:
  - Title — # of Processes
  - Tick Label — Arial font, size 11
- Label Text:
  - PROCESS\_START\_TIME — Start Date
  - COUNT(\*) — '-'

## Process Outflow Trend View

The Process Outflow Trend view has the following characteristics and changes from default settings:

- Type — Line
- Query — see [Process Outflow Trend Query](#)
- Title properties:
  - Title Text — Process Outflow Trend, Arial font, 11, center aligned
- X-Axis properties:
  - Tick Label — Arial font, size 11
- Y-Axis properties:
  - Title — # of Processes
  - Tick Label — Arial font, size 11
- Label Text:
  - PROCESS\_END\_TIME — End Date
  - COUNT(\*) — '-'

## Process Cycle Time Trend Query

The Process Cycle Time Trend query has the following characteristics and changes from default settings:

- Type — Group SQL Query

- Data object — see [Process Data Object](#)
- The following measures and aggregations are selected:
  - Process Cycle Time in Days, AVG
- The following dimensions are selected:
  - Process End Time, Use Time Series, Time Unit Day, Quantity 1
- Filters — **All are true** branch:
  - `PROCESS_INSTANCE_STATUS` is equal to 'COMPLETED'
  - `PROCESS_DEPLOYMENT_STATUS` is equal to '1'
  - `DAYS_SINCE_PROCESS` is less than or equal to `$/Time Period (Days)`
  - `DAYS_SINCE_PROCESS` is greater than or equal to '0'
  - `PROCESS_END_TIME` is less than or equal to `$Select data before`
  - `PROCESS_END_TIME` is greater than or equal to `$Select data after`
  - `PROCESS_LABEL` is equal to `$Select a Process`

See [Time Period \(Days\) Parameter](#), [Select data before Parameter](#), [Select data after Parameter](#), and [Select a Process Parameter](#)

## Process Inflow Trend Query

The Process Inflow Trend query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [Process Data Object](#)
- The following measures and aggregations are selected:
  - COUNT(\*)
- The following dimensions are selected:
  - Process Start Time, Use Time Series, Time Unit Day, Quantity 1
- Legend — Process Display Name
- Filters — **All are true** branch:
  - `PROCESS_DEPLOYMENT_STATUS` is equal to '1'
  - `DAYS_SINCE_PROCESS_START` is less than or equal to `$/Time Period (Days)`
  - `DAYS_SINCE_PROCESS_START` is greater than or equal to '0'
  - `PROCESS_START_TIME` is less than or equal to `$Select data before`
  - `PROCESS_START_TIME` is greater than or equal to `$Select data after`
  - `PROCESS_LABEL` is equal to `$Select a Process`

See [Time Period \(Days\) Parameter](#), [Select data before Parameter](#), [Select data after Parameter](#), and [Select a Process Parameter](#)

## Process Outflow Trend Query

The Process Outflow Trend query has the following characteristics and changes from default settings:

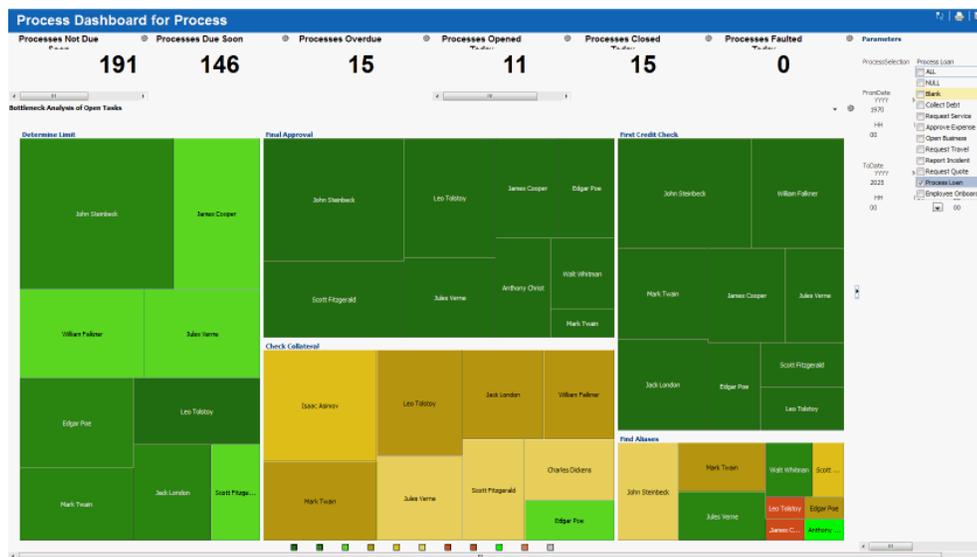
- Type — Group SQL Query
- Data object — see [Process Data Object](#)
- The following measures and aggregations are selected:
  - COUNT(\*)
- The following dimensions are selected:
  - Process End Time, Use Time Series, Time Unit Day, Quantity 1
- Legend — Process Display Name
- Filters — **All are true** branch:
  - PROCESS\_INSTANCE\_STATUS is equal to 'COMPLETED'
  - PROCESS\_DEPLOYMENT\_STATUS is equal to '1'
  - DAYS\_SINCE\_PROCESS is less than or equal to \$/Time Period (Days)
  - DAYS\_SINCE\_PROCESS is greater than or equal to '0'
  - PROCESS\_END\_TIME is less than or equal to \$Select data before
  - PROCESS\_END\_TIME is greater than or equal to \$Select data after
  - PROCESS\_LABEL is equal to \$Select a Process

See [Time Period \(Days\) Parameter](#), [Select data before Parameter](#), [Select data after Parameter](#), and [Select a Process Parameter](#)

## Process Dashboard for Process Dashboard

The Process Dashboard for Process dashboard is a set of process metrics and a treemap that uses size and color to identify task assignees with high cycle time and large volume. You can filter the data by process.

**Figure 12-8 Process Dashboard for Process Dashboard**



The Title properties of the dashboard are Arial font, size 5, bold.

The Process Dashboard for Process dashboard contains or references the following entities:

- [Processes Opened Today Total View](#)
- [Processes Closed Today Total View](#)
- [Processes Faulted Today Total View](#)
- [Processes Not Due Soon Total View](#)
- [Processes Due Soon Total View](#)
- [Processes Overdue Total View](#)
- [Bottleneck Analysis of Open Tasks View](#)
- [Processes Opened Today Total Query](#)
- [Processes Closed Today Total Query](#)
- [Processes Faulted Today Total Query](#)
- [Processes Not Due Soon Total Query](#)
- [Processes Due Soon Total Query](#)
- [Processes Overdue Total Query](#)
- [Bottleneck Analysis of Open Tasks Query](#)

## Processes Opened Today Total View

The Processes Opened Today Total view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Processes Opened Today Total Query](#)
- Text properties:
  - Title Text — Processes Opened Today, Arial font, size 15, center aligned
  - Table Text — Arial font, size 42, bold, center aligned
  - Column Header Visible — No

## Processes Closed Today Total View

The Processes Closed Today Total view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Processes Closed Today Total Query](#)
- Text properties:
  - Title Text — Processes Closed Today, Arial font, size 15, center aligned
  - Table Text — Arial font, size 42, bold, center aligned
  - Column Header Visible — No

## Processes Faulted Today Total View

The Processes Faulted Today Total view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Processes Faulted Today Total Query](#)
- Text properties:
  - Title Text — `Processes Faulted Today`, Arial font, size 15, center aligned
  - Table Text — Arial font, size 42, bold, center aligned
  - Column Header Visible — No

## Processes Not Due Soon Total View

The Processes Not Due Soon Total view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Processes Not Due Soon Total Query](#)
- Text properties:
  - Title Text — `Processes Open`, Arial font, size 15, center aligned
  - Table Text — Arial font, size 42, bold, center aligned
  - Column Header Visible — No

## Processes Due Soon Total View

The Processes Due Soon Total view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Processes Due Soon Total Query](#)
- Text properties:
  - Title Text — `Processes Due Soon`, Arial font, size 15, center aligned
  - Table Text — Arial font, size 42, bold, center aligned
  - Column Header Visible — No

## Processes Overdue Total View

The Processes Overdue Total view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Processes Overdue Total Query](#)
- Text properties:
  - Title Text — `Processes Overdue`, Arial font, size 15, center aligned
  - Table Text — Arial font, size 42, bold, center aligned
  - Column Header Visible — No

## Bottleneck Analysis of Open Tasks View

The Bottleneck Analysis of Open Tasks view has the following characteristics and changes from default settings:

- Type — Treemap
- Query — see [Bottleneck Analysis of Open Tasks Query](#)
- Value — COUNT(\*)
- Color — AVG(DAYS\_SINCE\_ASSIGNMENT)
- Title — Bottleneck Analysis of Open Tasks, Arial font, 11
- Color Thresholds properties:
  - 0–0.5, 2d7113 hex color
  - 0.5–1.0, 3d9a1b hex color
  - 1.0–1.5, 5ada29 hex color
  - 1.5–2.0, ba9d14 hex color
  - 2.0–2.5, e7c624 hex color
  - 2.5–3.0, edd35a hex color
  - 3.0–3.5, a93f1b hex color
  - 3.5–4.0, fefaf9 hex color
  - Overflow, e47752 hex color

## Processes Opened Today Total Query

The Processes Opened Today Total query has the following characteristics and changes from default settings:

- Type — Group SQL Query
  - Data object — see [Process Data Object](#)
  - The following measures and aggregations are selected:
    - Is Opened Today, SUM
  - Filters — **All are true** branch:
    - `PROCESS_LABEL is equal to $Select a Process`
- See [Select a Process Parameter](#)

## Processes Closed Today Total Query

The Processes Closed Today Total query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [Process Data Object](#)
- The following measures and aggregations are selected:
  - Is Closed Today, SUM

- Filters — **All are true** branch:
  - PROCESS\_LABEL is equal to \$Select a ProcessSee [Select a Process Parameter](#)

## Processes Faulted Today Total Query

The Processes Faulted Today Total query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [Process Data Object](#)
- The following measures and aggregations are selected:
  - Is Faulted Today, SUM
- Filters — **All are true** branch:
  - PROCESS\_LABEL is equal to \$Select a ProcessSee [Select a Process Parameter](#)

## Processes Not Due Soon Total Query

The Processes Not Due Soon Total query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [Process Data Object](#)
- The following measures and aggregations are selected:
  - Is Open and Not Due Soon, SUM
- Filters — **All are true** branch:
  - PROCESS\_LABEL is equal to \$Select a ProcessSee [Select a Process Parameter](#)

## Processes Due Soon Total Query

The Processes Due Soon Total query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [Process Data Object](#)
- The following measures and aggregations are selected:
  - Is Due Soon, SUM
- Filters — **All are true** branch:
  - PROCESS\_LABEL is equal to \$Select a ProcessSee [Select a Process Parameter](#)

## Processes Overdue Total Query

The Processes Overdue Total query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [Process Data Object](#)
- The following measures and aggregations are selected:
  - Is Overdue, SUM
- Filters — **All are true** branch:
  - `PROCESS_LABEL` is equal to `$$Select a Process`

See [Select a Process Parameter](#)

## Bottleneck Analysis of Open Tasks Query

The Bottleneck Analysis of Open Tasks query has the following characteristics and changes from default settings:

- Type — Tree Model Query
- Data object — see [HWF Task Assignment Data Object](#)
- The following measures and aggregations are selected:
  - Days Since Assignment, AVG
  - `COUNT(*)`
- Hierarchy — TaskAssignee
- Filters — **All are true** branch:
  - `ASSIGNMENT_STATE` is equal to `'ASSIGNED'`
  - `ASSIGNMENT_START_DATE` is less than or equal to `$$Select data before`
  - `ASSIGNMENT_START_DATE` is greater than or equal to `$$Select data after`
  - `DAYS_SINCE_ASSIGNMENT` is greater than or equal to `'0'`
  - `PROCESS_LABEL` is equal to `$$Select a Process`

See [Select data before Parameter](#), [Select data after Parameter](#), and [Select a Process Parameter](#)

## Process Activity Analysis Dashboard

The Process Activity Analysis dashboard is a set of task metrics and bar charts that shows the number of open, due soon, and overdue tasks by task or assignee. You can filter the data by process.

**Figure 12-9 Process Activity Analysis Dashboard**



The Title properties of the dashboard are Arial font, size 5, bold.

The Process Activity Analysis dashboard contains or references the following entities:

- [Tasks Opened Today Total View](#)
- [Tasks Closed Today Total View](#)
- [Tasks Not Due Soon Total View](#)
- [Tasks Due Soon Total View](#)
- [Tasks Overdue Total View](#)
- [Tasks Not Due Soon View](#)
- [Tasks Due Soon View](#)
- [Tasks Overdue View](#)
- [Assignees Not Due Soon View](#)
- [Assignees Due Soon View](#)
- [Assignees Overdue View](#)
- [Tasks Opened Today Total Query](#)
- [Tasks Closed Today Total Query](#)
- [Tasks Not Due Soon Total Query](#)
- [Tasks Due Soon Total Query](#)
- [Tasks Overdue Total Query](#)
- [Tasks Not Due Soon Query](#)
- [Tasks Due Soon Query](#)
- [Tasks Overdue Query](#)
- [Assignees Not Due Soon Query](#)
- [Assignees Due Soon Query](#)

- [Assignees Overdue Query](#)

## Tasks Opened Today Total View

The Tasks Opened Today Total view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Tasks Opened Today Total Query](#)
- Text properties:
  - Title Text — `Tasks Opened Today`, Arial font, size 15, center aligned
  - Table Text — Arial font, size 42, bold, center aligned
  - Column Header Visible — No

## Tasks Closed Today Total View

The Tasks Closed Today Total view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Tasks Closed Today Total Query](#)
- Text properties:
  - Title Text — `Tasks Closed Today`, Arial font, size 15, center aligned
  - Table Text — Arial font, size 42, bold, center aligned
  - Column Header Visible — No

## Tasks Not Due Soon Total View

The Tasks Not Due Soon Total view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Tasks Not Due Soon Total Query](#)
- Text properties:
  - Title Text — `Tasks Open`, Arial font, size 15, center aligned
  - Table Text — Arial font, size 42, bold, 276d31 hex color, center aligned
  - Column Header Visible — No

## Tasks Due Soon Total View

The Tasks Due Soon Total view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Tasks Due Soon Total Query](#)
- Text properties:
  - Title Text — `Tasks Due Soon`, Arial font, size 15, center aligned

- Table Text — Arial font, size 42, bold, fcd100 hex color, center aligned
- Column Header Visible — No

## Tasks Overdue Total View

The Tasks Overdue Total view has the following characteristics and changes from default settings:

- Type — Collapsed List
- Query — see [Tasks Overdue Total Query](#)
- Text properties:
  - Title Text — `Tasks Overdue`, Arial font, size 15, center aligned
  - Table Text — Arial font, size 42, bold, fcd100 hex color, center aligned
  - Column Header Visible — No

## Tasks Not Due Soon View

The Tasks Not Due Soon view has the following characteristics and changes from default settings:

- Type — Horizontal Bar
- Query — see [Tasks Not Due Soon Query](#)
- General properties:
  - Predefined Style — Green, Yellow, Red
- Title properties:
  - Title Text — `Top 7 Tasks Within Due Date`, Arial font, 11, bold, center aligned
- X-Axis properties:
  - Tick Label — Arial font, size 11
- Y-Axis properties:
  - Tick Label — Arial font, size 11
- Legend properties:
  - Display Legend — No

## Tasks Due Soon View

The Tasks Due Soon view has the following characteristics and changes from default settings:

- Type — Horizontal Bar
- Query — see [Tasks Due Soon Query](#)
- General properties:
  - Predefined Style — Green, Yellow, Red
- Title properties:
  - Title Text — `Top 7 Tasks Due Soon`, Arial font, 11, bold, center aligned
- X-Axis properties:

- Tick Label — Arial font, size 11
- Y-Axis properties:
  - Tick Label — Arial font, size 11
- Legend properties:
  - Display Legend — No

## Tasks Overdue View

The Tasks Overdue view has the following characteristics and changes from default settings:

- Type — Horizontal Bar
- Query — see [Tasks Overdue Query](#)
- General properties:
  - Predefined Style — Green, Yellow, Red
- Title properties:
  - Title Text — Top 7 Tasks Overdue, Arial font, 11, bold, center aligned
- X-Axis properties:
  - Tick Label — Arial font, size 11
- Y-Axis properties:
  - Tick Label — Arial font, size 11
- Legend properties:
  - Display Legend — No

## Assignees Not Due Soon View

The Assignees Not Due Soon view has the following characteristics and changes from default settings:

- Type — Horizontal Bar
- Query — see [Assignees Not Due Soon Query](#)
- General properties:
  - Predefined Style — Green, Yellow, Red
- Title properties:
  - Title Text — Top 7 Assignees by Tasks Within Due Date, Arial font, 11, bold, center aligned
- X-Axis properties:
  - Tick Label — Arial font, size 11
- Y-Axis properties:
  - Tick Label — Arial font, size 11
- Legend properties:
  - Display Legend — No

## Assignees Due Soon View

The Assignees Due Soon view has the following characteristics and changes from default settings:

- Type — Horizontal Bar
- Query — see [Assignees Due Soon Query](#)
- General properties:
  - Predefined Style — Green, Yellow, Red
- Title properties:
  - Title Text — Top 7 Assignees by Tasks Due Soon, Arial font, 11, bold, center aligned
- X-Axis properties:
  - Tick Label — Arial font, size 11
- Y-Axis properties:
  - Tick Label — Arial font, size 11
- Legend properties:
  - Display Legend — No

## Assignees Overdue View

The Assignees Overdue view has the following characteristics and changes from default settings:

- Type — Horizontal Bar
- Query — see [Assignees Overdue Query](#)
- General properties:
  - Predefined Style — Green, Yellow, Red
- Title properties:
  - Title Text — Top 7 Assignees by Tasks Overdue, Arial font, 11, bold, center aligned
- X-Axis properties:
  - Tick Label — Arial font, size 11
- Y-Axis properties:
  - Tick Label — Arial font, size 11
- Legend properties:
  - Display Legend — No

## Tasks Opened Today Total Query

The Tasks Opened Today Total query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [HWF Task Data Object](#)

- The following measures and aggregations are selected:
    - Is Opened Today, SUM
  - Filters — **All are true** branch:
    - PROCESS\_LABEL is equal to \$Select a Process
- See [Select a Process Parameter](#)

## Tasks Closed Today Total Query

The Tasks Closed Today Total query has the following characteristics and changes from default settings:

- Type — Group SQL Query
  - Data object — see [HWF Task Data Object](#)
  - The following measures and aggregations are selected:
    - Is Closed Today, SUM
  - Filters — **All are true** branch:
    - PROCESS\_LABEL is equal to \$Select a Process
- See [Select a Process Parameter](#)

## Tasks Not Due Soon Total Query

The Tasks Not Due Soon Total query has the following characteristics and changes from default settings:

- Type — Group SQL Query
  - Data object — see [HWF Task Data Object](#)
  - The following measures and aggregations are selected:
    - Is Open and not Due Soon, SUM
  - Filters — **All are true** branch:
    - PROCESS\_LABEL is equal to \$Select a Process
- See [Select a Process Parameter](#)

## Tasks Due Soon Total Query

The Tasks Due Soon Total query has the following characteristics and changes from default settings:

- Type — Group SQL Query
  - Data object — see [HWF Task Data Object](#)
  - The following measures and aggregations are selected:
    - Is Due Soon, SUM
  - Filters — **All are true** branch:
    - PROCESS\_LABEL is equal to \$Select a Process
- See [Select a Process Parameter](#)

## Tasks Overdue Total Query

The Tasks Overdue Total query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [HWF Task Data Object](#)
- The following measures and aggregations are selected:
  - Is Overdue, SUM
- Filters — **All are true** branch:
  - `PROCESS_LABEL is equal to $Select a Process`

See [Select a Process Parameter](#)

## Tasks Not Due Soon Query

The Tasks Not Due Soon query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [HWF Task Data Object](#)
- The following measures and aggregations are selected:
  - Is Open and not Due Soon, SUM
  - Is Due Soon, SUM
  - Is Overdue, SUM
- Top N — 7, Apply on SUM(Is Open and not Due Soon)
- The following dimensions are selected:
  - TITLE
- Filters — **All are true** branch:
  - `PROCESS_LABEL is equal to $Select a Process`

See [Select a Process Parameter](#)

## Tasks Due Soon Query

The Tasks Due Soon query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [HWF Task Data Object](#)
- The following measures and aggregations are selected:
  - Is Open and not Due Soon, SUM
  - Is Due Soon, SUM
  - Is Overdue, SUM
- Top N — 7, Apply on SUM(Is Due Soon)
- The following dimensions are selected:

- TITLE
  - Filters — **All are true** branch:
    - PROCESS\_LABEL is equal to \$Select a Process
- See [Select a Process Parameter](#)

## Tasks Overdue Query

The Tasks Overdue query has the following characteristics and changes from default settings:

- Type — Group SQL Query
  - Data object — see [HWF Task Data Object](#)
  - The following measures and aggregations are selected:
    - Is Open and not Due Soon, SUM
    - Is Due Soon, SUM
    - Is Overdue, SUM
  - Top N — 7, Apply on SUM(Is Overdue)
  - The following dimensions are selected:
    - TITLE
  - Filters — **All are true** branch:
    - PROCESS\_LABEL is equal to \$Select a Process
- See [Select a Process Parameter](#)

## Assignees Not Due Soon Query

The Assignees Not Due Soon query has the following characteristics and changes from default settings:

- Type — Group SQL Query
  - Data object — see [HWF Task Assignment Data Object](#)
  - The following measures and aggregations are selected:
    - Is Open and not Due Soon, SUM
    - Is Due Soon, SUM
    - Is Overdue, SUM
  - Top N — 7, Apply on SUM(Is Open and not Due Soon)
  - The following dimensions are selected:
    - ASSIGNEE\_DISPLAY\_NAME
  - Filters — **All are true** branch:
    - PROCESS\_LABEL is equal to \$Select a Process
    - IS\_GROUP is equal to \$Assignee Is Group
- See [Select a Process Parameter](#) and [Assignee Is Group Parameter](#)

## Assignees Due Soon Query

The Assignees Due Soon query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [HWF Task Assignment Data Object](#)
- The following measures and aggregations are selected:
  - Is Open and not Due Soon, SUM
  - Is Due Soon, SUM
  - Is Overdue, SUM
- Top N — 7, Apply on SUM(Is Due Soon)
- The following dimensions are selected:
  - ASSIGNEE\_DISPLAY\_NAME
- Filters — **All are true** branch:
  - PROCESS\_LABEL is equal to \$Select a Process
  - IS\_GROUP is equal to \$Assignee Is Group

See [Select a Process Parameter](#) and [Assignee Is Group Parameter](#)

## Assignees Overdue Query

The Assignees Overdue query has the following characteristics and changes from default settings:

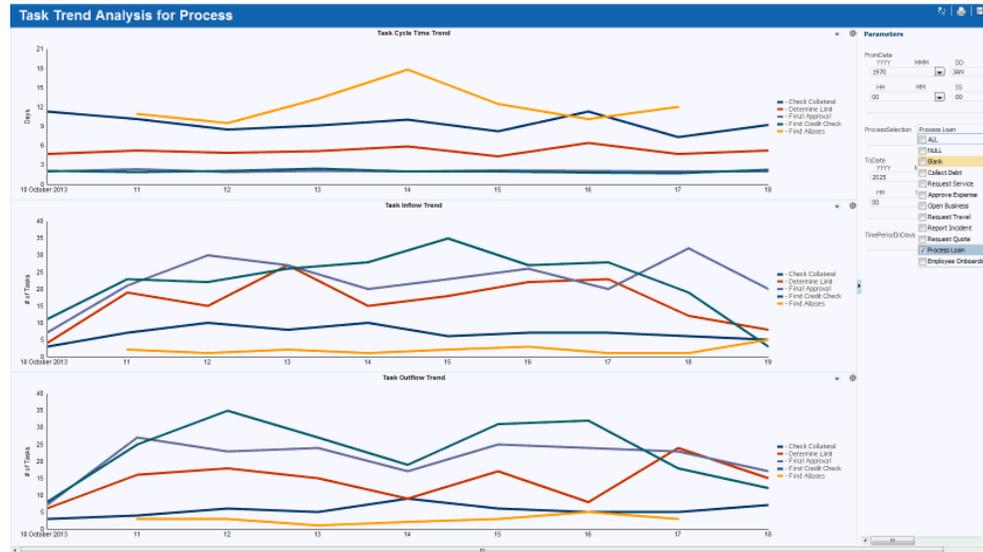
- Type — Group SQL Query
- Data object — see [HWF Task Assignment Data Object](#)
- The following measures and aggregations are selected:
  - Is Open and not Due Soon, SUM
  - Is Due Soon, SUM
  - Is Overdue, SUM
- Top N — 7, Apply on SUM(Is Overdue)
- The following dimensions are selected:
  - ASSIGNEE\_DISPLAY\_NAME
- Filters — **All are true** branch:
  - PROCESS\_LABEL is equal to \$Select a Process
  - IS\_GROUP is equal to \$Assignee Is Group

See [Select a Process Parameter](#) and [Assignee Is Group Parameter](#)

## Task Trend Analysis for Process Dashboard

The Process Activity Analysis dashboard is a set of line graph is that displays task cycle time, inflow, and outflow by day. You can filter the data by date and process.

**Figure 12-10 Task Trend Analysis for Process Dashboard**



The Title properties of the dashboard are Arial font, size 5, bold.

The Process Activity Analysis dashboard contains or references the following entities:

- [Task Cycle Time Trend View](#)
- [Task Inflow Trend View](#)
- [Task Outflow Trend View](#)
- [Task Cycle Time Trend Query](#)
- [Task Inflow Trend Query](#)
- [Task Outflow Trend Query](#)

## Task Cycle Time Trend View

The Task Cycle Time Trend view has the following characteristics and changes from default settings:

- Type — Line
- Query — see [Task Cycle Time Trend Query](#)
- Title properties:
  - Title Text — Task Cycle Time Trend, Arial font, 11, center aligned
- X-Axis properties:
  - Tick Label — Arial font, size 11
- Y-Axis properties:
  - Title — Days
  - Tick Label — Arial font, size 11
- Label Text:
  - END\_DATE — End Date

- `AVG(TASK_TOTAL_TIME_IN_DAYS)` — '-'

## Task Inflow Trend View

The Task Inflow Trend view has the following characteristics and changes from default settings:

- Type — Line
- Query — see [Task Inflow Trend Query](#)
- Title properties:
  - Title Text — `Task Inflow Trend`, Arial font, 11, center aligned
- X-Axis properties:
  - Tick Label — Arial font, size 11
- Y-Axis properties:
  - Title — # of Tasks
  - Tick Label — Arial font, size 11
- Label Text:
  - `START_DATE` — Start Date
  - `COUNT(*)` — '-'

## Task Outflow Trend View

The Task Outflow Trend view has the following characteristics and changes from default settings:

- Type — Line
- Query — see [Task Outflow Trend Query](#)
- Title properties:
  - Title Text — `Task Outflow Trend`, Arial font, 11, center aligned
- X-Axis properties:
  - Tick Label — Arial font, size 11
- Y-Axis properties:
  - Title — # of Tasks
  - Tick Label — Arial font, size 11
- Label Text:
  - `END_DATE` — End Date
  - `COUNT(*)` — '-'

## Task Cycle Time Trend Query

The Task Cycle Time Trend query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [HWF Task Data Object](#)

- The following measures and aggregations are selected:
    - Task Total Time in Days, AVG
  - The following dimensions are selected:
    - END\_DATE, Use Time Series, Time Unit Day, Quantity 1
  - Legend — TITLE
  - Filters — **All are true** branch:
    - TASK\_STATE is equal to 'COMPLETED'
    - DAYS\_SINCE\_TASK is less than or equal to \$/Time Period (Days)
    - DAYS\_SINCE\_TASK is greater than or equal to '0'
    - END\_DATE is less than or equal to \$Select data before
    - END\_DATE is greater than or equal to \$Select data after
    - PROCESS\_LABEL is equal to \$Select a Process
- See [Time Period \(Days\) Parameter](#), [Select data before Parameter](#), [Select data after Parameter](#), and [Select a Process Parameter](#)

## Task Inflow Trend Query

The Task Inflow Trend query has the following characteristics and changes from default settings:

- Type — Group SQL Query
  - Data object — see [HWF Task Data Object](#)
  - The following measures and aggregations are selected:
    - COUNT(\*), AVG
  - The following dimensions are selected:
    - START\_DATE, Use Time Series, Time Unit Day, Quantity 1
  - Legend — TITLE
  - Filters — **All are true** branch:
    - DAYS\_SINCE\_TASK is less than or equal to \$/Time Period (Days)
    - DAYS\_SINCE\_TASK is greater than or equal to '0'
    - END\_DATE is less than or equal to \$Select data before
    - END\_DATE is greater than or equal to \$Select data after
    - PROCESS\_LABEL is equal to \$Select a Process
- See [Time Period \(Days\) Parameter](#), [Select data before Parameter](#), [Select data after Parameter](#), and [Select a Process Parameter](#)

## Task Outflow Trend Query

The Task Outflow Trend query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [HWF Task Data Object](#)

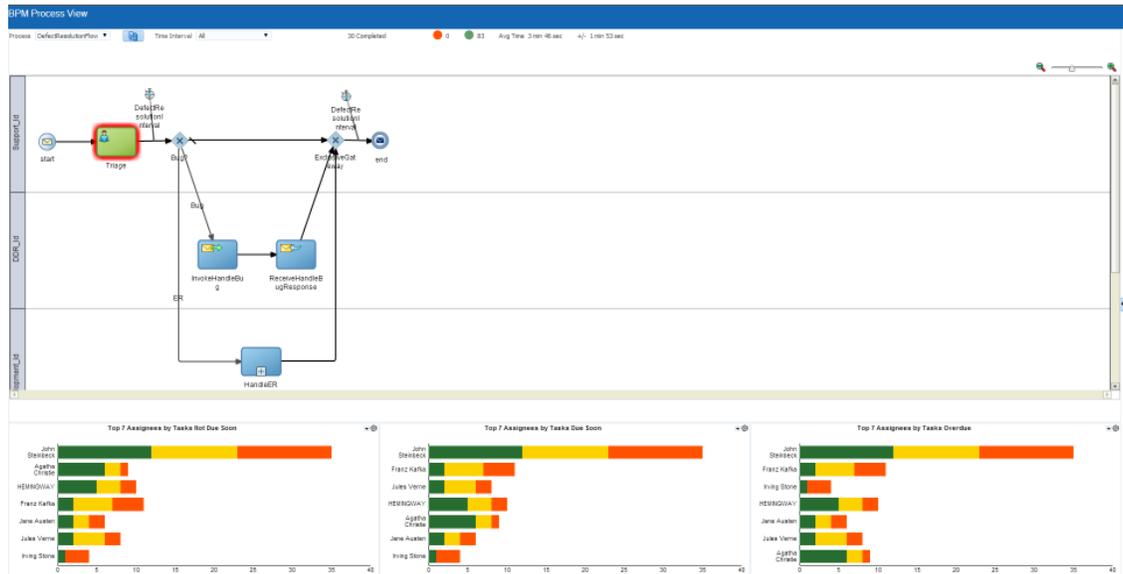
- The following measures and aggregations are selected:
  - COUNT(\*), AVG
- The following dimensions are selected:
  - END\_DATE, Use Time Series, Time Unit Day, Quantity 1
- Legend — TITLE
- Filters — **All are true** branch:
  - TASK\_STATE is equal to 'COMPLETED'
  - DAYS\_SINCE\_TASK is less than or equal to \$/Time Period (Days)
  - DAYS\_SINCE\_TASK is greater than or equal to '0'
  - END\_DATE is less than or equal to \$Select data before
  - END\_DATE is greater than or equal to \$Select data after
  - PROCESS\_LABEL is equal to \$Select a Process

See [Time Period \(Days\) Parameter](#), [Select data before Parameter](#), [Select data after Parameter](#), and [Select a Process Parameter](#)

## BPM Process View Dashboard

The BPM Process View dashboard displays a BPM process along with corresponding metrics and bottleneck analysis. It also includes bar charts with the number of open, due soon, and overdue tasks for assignees.

**Figure 12-11 BPM Process View Dashboard**



The Title properties of the dashboard are Arial font, size 5, bold.

In the Parameter Mapping Editor, the PROCESS\_NAME field in the BPM process drives values of the Process Name parameters in the other views. See [Driving Parameters from Other Views](#) and [Process Name Parameter](#).

The BPM Process View dashboard contains or references the following entities:

- [Assignees Not Due Soon PV View](#)
- [Assignees Due Soon PV View](#)
- [Assignees Overdue PV View](#)
- [Assignees Not Due Soon PV Query](#)
- [Assignees Due Soon PV Query](#)
- [Assignees Overdue PV Query](#)

## Assignees Not Due Soon PV View

The Assignees Not Due Soon view has the following characteristics and changes from default settings:

- Type — Horizontal Bar
- Query — see [Assignees Not Due Soon PV Query](#)
- General properties:
  - Predefined Style — Green, Yellow, Red
- Title properties:
  - Title Text — Top 7 Assignees Within Due Date, Arial font, 11, bold, center aligned
- X-Axis properties:
  - Tick Label — Arial font, size 11
- Y-Axis properties:
  - Tick Label — Arial font, size 11
- Legend properties:
  - Display Legend — No

## Assignees Due Soon PV View

The Assignees Due Soon view has the following characteristics and changes from default settings:

- Type — Horizontal Bar
- Query — see [Assignees Due Soon PV Query](#)
- General properties:
  - Predefined Style — Green, Yellow, Red
- Title properties:
  - Title Text — Top 7 Assignees Due Soon, Arial font, 11, bold, center aligned
- X-Axis properties:
  - Tick Label — Arial font, size 11
- Y-Axis properties:
  - Tick Label — Arial font, size 11
- Legend properties:

- Display Legend — No

## Assignees Overdue PV View

The Assignees Overdue view has the following characteristics and changes from default settings:

- Type — Horizontal Bar
- Query — see [Assignees Overdue PV Query](#)
- General properties:
  - Predefined Style — Green, Yellow, Red
- Title properties:
  - Title Text — Top 7 Assignees Overdue, Arial font, 11, bold, center aligned
- X-Axis properties:
  - Tick Label — Arial font, size 11
- Y-Axis properties:
  - Tick Label — Arial font, size 11
- Legend properties:
  - Display Legend — No

## Assignees Not Due Soon PV Query

The Assignees Not Due Soon PV query has the following characteristics and changes from default settings:

- Type — Group SQL Query
- Data object — see [HWF Task Assignment Data Object](#)
- The following measures and aggregations are selected:
  - Is Open and not Due Soon, SUM
  - Is Due Soon, SUM
  - Is Overdue, SUM
- Top N — 7, Apply on SUM(Is Open and not Due Soon)
- The following dimensions are selected:
  - ASSIGNEE\_DISPLAY\_NAME
- Filters — **All are true** branch:
  - PROCESS\_LABEL is equal to \$Process Name
  - IS\_GROUP is equal to \$Assignee Is Group

See [Process Name Parameter](#) and [Assignee Is Group Parameter](#)

## Assignees Due Soon PV Query

The Assignees Due Soon PV query has the following characteristics and changes from default settings:

- Type — Group SQL Query

- Data object — see [HWF Task Assignment Data Object](#)
  - The following measures and aggregations are selected:
    - Is Open and not Due Soon, SUM
    - Is Due Soon, SUM
    - Is Overdue, SUM
  - Top N — 7, Apply on SUM(Is Due Soon)
  - The following dimensions are selected:
    - ASSIGNEE\_DISPLAY\_NAME
  - Filters — **All are true** branch:
    - PROCESS\_LABEL is equal to \$Process Name
    - IS\_GROUP is equal to \$Assignee Is Group
- See [Process Name Parameter](#) and [Assignee Is Group Parameter](#)

## Assignees Overdue PV Query

The Assignees Overdue PV query has the following characteristics and changes from default settings:

- Type — Group SQL Query
  - Data object — see [HWF Task Assignment Data Object](#)
  - The following measures and aggregations are selected:
    - Is Open and not Due Soon, SUM
    - Is Due Soon, SUM
    - Is Overdue, SUM
  - Top N — 7, Apply on SUM(Is Overdue)
  - The following dimensions are selected:
    - ASSIGNEE\_DISPLAY\_NAME
  - Filters — **All are true** branch:
    - PROCESS\_LABEL is equal to \$Process Name
    - IS\_GROUP is equal to \$Assignee Is Group
- See [Process Name Parameter](#) and [Assignee Is Group Parameter](#)

## BPM Dashboard Parameters

Multiple queries in the preassembled BPM dashboards use the same parameters.

This section includes the following topics:

- [Assignee Is Group Parameter](#)
- [Process Name Parameter](#)
- [Select a Process Parameter](#)
- [Select data after Parameter](#)
- [Select data before Parameter](#)

- [Time Period \(Days\) Parameter](#)

## Assignee Is Group Parameter

The Assignee Is Group parameter has the following characteristics and changes from default settings:

- Type — List
- Data type — VARCHAR
- Required — Yes
- Data object — /oracle/processanalytics/internal/physical/HWF Assignment (physical)
- Field — IS\_GROUP
- Value Options — FALSE, TRUE
- Choose values — Yes
- Multiple values — Yes
- Default — FALSE

## Process Name Parameter

The Process Name parameter has the following characteristics and changes from default settings:

- Type — Value
- Data type — VARCHAR
- Required — Yes
- Default — Input value, ALL

## Select a Process Parameter

The Select a Process parameter has the following characteristics and changes from default settings:

- Type — List
- Data type — VARCHAR
- Required — Yes
- Data object — /oracle/processanalytics/internal/physical/Process Definition
- Field — PROCESS\_LABEL
- Special Options — ALL
- Multiple values — Yes
- Default — Process Loan

## Select data after Parameter

The Select data after parameter has the following characteristics and changes from default settings:

- Type — Value

- Data type — DATETIME
- Required — Yes
- Default — Input value, 1/1/1970 12:00 AM

## Select data before Parameter

The Select data before parameter has the following characteristics and changes from default settings:

- Type — Value
- Data type — DATETIME
- Required — Yes
- Default — Input value, 1/1/2025 12:00 AM

## Time Period (Days) Parameter

The Time Period (Days) parameter has the following characteristics and changes from default settings:

- Type — Value
- Data type — VARCHAR
- Required — Yes
- Default — Input value, 7

# Understanding the Oracle BAM Data Objects for BPM

All the BAM data objects for BPM are included in the default Process Analytics project in the Designer page.

To view information about a BAM data object while in either the Designer or Administration page, first click the arrow to the left of **Data Objects** in the left navigation pane. Open any folders in the same manner. Then do one of the following:

- Click the data object name.
- Select the data object icon and click the **Edit** icon.
- Right-click the data object name or icon, and select the **Edit** from the menu.

All the BAM data objects for BPM are in the `/oracle/processanalytics` folder, arranged as follows:

- Activity
- Process
- HWF Task
- HWF Task Assignment
- internal (folder)
  - Activity Definition
  - BPM Activity (Secured)
  - BPM Process (Secured)

- Composite Definition
- Process Definition
- Role Definition
- physical (folder)
  - \* Activity (physical)
  - \* Activity Definition (physical)
  - \* Composite Definition
  - \* HWF Assignment (physical)
  - \* HWF Task Definition
  - \* Process (physical)
  - \* Process Definition
  - \* Role Definition (physical)

All data objects in the physical folder are simple data objects. All other data objects under /oracle/processanalytics are logical data objects.



**Note:**

It is strongly recommended that you do *not* change any data objects in the `oracle` folder. Doing so could break dependent entities such as the preassembled BPM dashboards and other Oracle samples.

The top-level logical data objects are referenced by the preassembled BPM dashboards and are described in these sections:

- [Process Data Object](#)
- [Activity Data Object](#)
- [HWF Task Data Object](#)
- [HWF Task Assignment Data Object](#)

Because the columns in these data objects are so numerous, only the calculated fields are listed. To display column information for a data object, see [Viewing Data Object Information](#) and [The Columns Tab](#).

For general information about data objects, see [Working with Data Objects](#) and [Creating and Managing Oracle BAM Data Objects](#).

## Process Data Object

The Process data object has the following characteristics:

- Type — Logical Data Object
- Full name — /oracle/processanalytics/Process
- Referenced data objects:
  - /oracle/processanalytics/internal/physical/Process
  - /oracle/processanalytics/internal/physical/Process Definition

- /oracle/processanalytics/internal/physical/Composite Definition
- Hierarchies:
  - CompositeProcessStatusHierarchy — Composite Name, Process Display Name, Process Instance Status
  - CompositeProcessHierarchy — Composite Name, Process Display Name
- Calculated fields:
  - IS\_OPENED\_TODAY — if ((DAYOFYEAR({Process Start Time})==DAYOFYEAR(NOW())) && (YEAR({Process Start Time})==YEAR(NOW()))) then(1) else(0)
  - IS\_CLOSED\_TODAY — if ((DAYOFYEAR({Process End Time})==DAYOFYEAR(NOW())) && (YEAR({Process End Time})==YEAR(NOW())) && ({Process Instance Status}=="COMPLETED")) then(1) else(0)
  - IS\_FAULTED\_TODAY — if ((DAYOFYEAR({Process End Time})==DAYOFYEAR(NOW())) && (YEAR({Process End Time})==YEAR(NOW())) && ({Process Instance Status}=="FAULTED")) then(1) else(0)
  - IS\_OPEN — if ({Process Instance Status}=="ACTIVE") then(1) else(0)
  - IS\_CLOSED — if ({Process Instance Status}=="COMPLETED") then(1) else(0)
  - IS\_FAULTED — if ({Process Instance Status}=="FAULTED") then(1) else(0)
  - IS\_OPEN\_NOT\_DUE\_SOON — if (({Process Instance Status}=="ACTIVE") && (DATEDIFF(SQL\_TSI\_DAY, NOW(), {Process Due Time})) > 5) then(1) else(0)
  - IS\_DUE\_SOON — if (({Process Instance Status}=="ACTIVE") && ({Process Due Time} > NOW()) && (DATEDIFF(SQL\_TSI\_DAY, NOW(), {Process Due Time})) <= 5) then(1) else(0)
  - IS\_OVERDUE — if (({Process Instance Status}=="ACTIVE") && ({Process Due Time} < NOW())) then(1) else(0)
  - MISSED\_DUE\_DATE — if (({Process Instance Status}=="ACTIVE") && ({Process Due Time} < NOW())) then(1) else(if ({Process Instance Status}=="COMPLETED") && ({Process Due Time} < {Process End Time})) then(1) else(0)
  - PROCESS\_CYCLE\_TIME\_IN\_HOURS — {Process Running Time (millisecs)}/3600000
  - PROCESS\_CYCLE\_TIME\_IN\_DAYS — {Process Running Time (millisecs)}/86400000
  - PROCESS\_OPEN\_TIME\_IN\_HOURS — DATEDIFF(SQL\_TSI\_SECOND, {Process Start Time}, NOW())/3600
  - ESTIMATED\_COMPLETE\_TIME\_DAYS — IF({Process Estimated Time}! =NULL) THEN(DATEDIFF(SQL\_TSI\_DAY, {Process Start Time}, {Process Estimated Time})) ELSE(DATEDIFF(SQL\_TSI\_DAY, {Process Start Time}, NOW()))
  - ESTIMATED\_COMPLETE\_TIME\_HOURS — IF({Process Estimated Time}! =NULL) THEN(DATEDIFF(SQL\_TSI\_HOUR, {Process Start Time}, {Process Estimated Time})) ELSE(DATEDIFF(SQL\_TSI\_HOUR, {Process Start Time}, NOW()))
  - DAYS\_SINCE\_PROCESS — IF({Process Instance Status}=="ACTIVE") THEN(DATEDIFF(SQL\_TSI\_DAY, {Process Start Time}, NOW())) ELSE(DATEDIFF(SQL\_TSI\_DAY, {Process End Time}, NOW()))

- DAYS\_SINCE\_PROCESS\_START — DATEDIFF (SQL\_TSI\_DAY, {Process Start Time}, NOW ())
- PROCESS\_INSTANCE\_STATUS\_CODE — if ({Process Instance Status}=="ACTIVE") then (1) else (if ({Process Instance Status}=="COMPLETED") then (2) else (3))
- DATA\_AVAILABILITY\_DATE — if (true) then ({Process Start Time}) else ({Process Start Time})

## Activity Data Object

The Activity data object has the following characteristics:

- Type — Logical Data Object
- Full name — /oracle/processanalytics/Activity
- Referenced data objects:
  - /oracle/processanalytics/internal/physical/Activity (physical)
  - /oracle/processanalytics/internal/physical/Activity Definition (physical)
  - /oracle/processanalytics/internal/physical/Process Definition
  - /oracle/processanalytics/internal/physical/Composite Definition
  - /oracle/processanalytics/internal/physical/Role Definition (physical)
- Hierarchies:
  - ProcessActivityHierarchy — Process Display Name, Activity Display Name
- Calculated fields:
  - ACTIVITY\_CYCLE\_TIME\_IN\_HOURS — Activity Running Time (milliseconds)/3600000
  - ACTIVITY\_CYCLE\_TIME\_IN\_DAYS — Activity Running Time (milliseconds)/86400000
  - ACTIVITY\_OPEN\_TIME\_IN\_HOURS — DATEDIFF (SQL\_TSI\_SECOND, Activity Start Time, NOW ()) /3600
  - ACTIVITY\_OPEN\_TIME\_IN\_DAYS — DATEDIFF (SQL\_TSI\_SECOND, Activity Start Time, NOW ()) /3600

## HWF Task Data Object

The HWF Task data object has the following characteristics:

- Type — Logical Data Object
- Full name — /oracle/processanalytics/HWF Task
- Referenced data objects:
  - /oracle/processanalytics/internal/physical/HWF Task (physical)
  - /oracle/processanalytics/internal/physical/HWF Task Definition
  - /oracle/processanalytics/internal/physical/Process Definition
  - /oracle/processanalytics/internal/physical/Composite Definition
  - /oracle/processanalytics/internal/physical/Activity Definition (physical)
- Calculated fields:

- IS\_OPENED\_TODAY —  
if ((DAYOFYEAR (START\_DATE) == DAYOFYEAR (NOW ())) && (YEAR (START\_DATE) == YEAR (NOW ()))) then (1) else (0)
- IS\_CLOSED\_TODAY —  
if ((DAYOFYEAR (END\_DATE) == DAYOFYEAR (NOW ())) && (YEAR (END\_DATE) == YEAR (NOW ()))) then (1) else (0)
- IS\_OPEN — If (TASK\_STATE=="ASSIGNED") then (1) else (0)
- IS\_OPEN\_AND\_NOT\_DUE\_SOON —  
if ((TASK\_STATE=="ASSIGNED") && (DATEDIFF (SQL\_TSI\_DAY, NOW (), DUE\_DATE)) > 5) then (1) else (0)
- IS\_DUE\_SOON —  
if ((TASK\_STATE=="ASSIGNED") && (DUE\_DATE > NOW ()) && (DATEDIFF (SQL\_TSI\_DAY, NOW (), DUE\_DATE)) <= 5) then (1) else (0)
- IS\_OVERDUE — if ((TASK\_STATE=="ASSIGNED") && (DUE\_DATE < NOW ())) then (1) else (0)
- DAYS\_SINCE\_TASK —  
IF (TASK\_STATE=="ASSIGNED") THEN (DATEDIFF (SQL\_TSI\_DAY, START\_DATE, NOW ())) ELSE (DATEDIFF (SQL\_TSI\_DAY, END\_DATE, NOW ()))
- DAYS\_SINCE\_TASK\_START — DATEDIFF (SQL\_TSI\_DAY, START\_DATE, NOW ())
- TASK\_TOTAL\_TIME\_IN\_DAYS — TASK\_TOTAL\_TIME/86400

## HWF Task Assignment Data Object

The HWF Task Assignment data object has the following characteristics:

- Type — Logical Data Object
- Full name — /oracle/processanalytics/HWF Task Assignment
- Referenced data objects:
  - /oracle/processanalytics/internal/physical/HWF Assignment (physical)
  - /oracle/processanalytics/internal/physical/HWF Task (physical)
  - /oracle/processanalytics/internal/physical/HWF Task Definition
  - /oracle/processanalytics/internal/physical/Process Definition
  - /oracle/processanalytics/internal/physical/Composite Definition
  - /oracle/processanalytics/internal/physical/Activity Definition (physical)
- Hierarchies:
  - TaskAssignee — Title, Assignee Display Name
- Calculated fields:
  - IS\_OPEN\_AND\_NOT\_DUE\_SOON —  
if ((TASK\_STATE=="ASSIGNED") && (DATEDIFF (SQL\_TSI\_DAY, NOW (), DUE\_DATE)) > 5) then (1) else (0)
  - IS\_DUE\_SOON —  
if ((TASK\_STATE=="ASSIGNED") && (DUE\_DATE > NOW ()) && (DATEDIFF (SQL\_TSI\_DAY, NOW (), DUE\_DATE)) <= 5) then (1) else (0)
  - IS\_OVERDUE — if ((TASK\_STATE=="ASSIGNED") && (DUE\_DATE < NOW ())) then (1) else (0)

- DAYS\_SINCE\_ASSIGNMENT —  
IF (TASK\_STATE=="ASSIGNED") THEN (DATEDIFF (SQL\_TSI\_DAY, ASSIGNMENT\_START\_DATE,  
NOW ( ) ) ELSE (DATEDIFF (SQL\_TSI\_DAY, ASSIGNMENT\_END\_DATE, NOW ( ) ) )
- TOTAL\_TIME\_IN\_DAYS — TOTAL\_TIME/86400

# 13

## Integrating with Oracle SOA Suite

This chapter describes how Oracle Business Activity Monitoring (Oracle BAM) integrates with Oracle Service-Oriented Architecture (SOA). This integration captures Business Process Expression Language (BPEL) process metrics for analysis and display in Oracle BAM.

This chapter includes the following sections:

- [Understanding Oracle BAM and Oracle SOA Suite Integration](#)
- [Understanding the Preassembled Dashboards](#)
- [Understanding the Oracle BAM Data Objects for Oracle SOA Suite](#)

### Understanding Oracle BAM and Oracle SOA Suite Integration

When Oracle BAM and Oracle SOA Suite are installed together and process metrics are enabled, the BAM data objects for SOA are automatically created when SOA composite applications are deployed and populated with metric data as BPEL processes run.

See [Enabling Process Metrics](#).

As a BAM user in a Designer role, all you need to do is add these SOA data objects to your BAM project.

Because each BPEL process is unique, some instrumentation is required on the SOA side in Oracle JDeveloper to capture the metric data in the BAM data objects. See "Configuring BPEL Process Analytics" in *Developing SOA Applications with Oracle SOA Suite* for more information.

### Understanding the Preassembled Dashboards

To assist you in creating BAM dashboards that work with SOA data objects, BAM provides a set of preassembled dashboards in the Process Analytics project.

These dashboards show common types of process analysis. Although designed for use with BPM, these dashboards can display data for BPEL processes in addition to BPM processes. See [Understanding the Pre-assembled BPM Dashboards](#) for more information.

See [Creating Dashboards](#) for details on how to create dashboards.

### Understanding the Oracle BAM Data Objects for Oracle SOA Suite

Oracle SOA Suite uses the same schema for data objects as BPM. Therefore, all the BAM data objects in the default Process Analytics project in the Designer page can be used with SOA as well.

See [Understanding the Oracle BAM Data Objects for BPM](#) for more information.

For general information about data objects, see [Working with Data Objects](#) and [Creating and Managing Oracle BAM Data Objects](#).

# Part IV

## Using Oracle Business Activity Monitoring as an Administrator

This part describes how to create data objects and Enterprise Message Sources. It also describes other administrative tasks, such as monitoring and security. Finally, it includes reference chapters for commands and web services.

This part includes the following chapters:

- [Creating and Managing Oracle BAM Data Objects](#)
- [Creating Oracle BAM Enterprise Message Sources](#)
- [Monitoring and Managing Oracle BAM](#)
- [Securing Oracle BAM](#)
- [Using Commands with Oracle BAM](#)
- [Using Oracle BAM Web Services](#)

# 14

## Creating and Managing Oracle BAM Data Objects

This chapter describes how to create and manage data objects in Oracle Business Activity Monitoring (Oracle BAM), including creating different data object types, creating different column types and hierarchies, assigning row security and permissions, and managing data.

This chapter includes the following sections:

- [Understanding Oracle BAM Data Objects](#)
- [Creating a Data Object](#)
- [Adding Columns and Hierarchies to a Data Object](#)
- [Securing a Data Object](#)
- [Managing Data in Data Objects](#)
- [Using Data Object Folders](#)
- [Editing, Renaming, and Deleting Data Objects](#)
- [Operators and Functions for Calculated Fields](#)

### Understanding Oracle BAM Data Objects

A BAM data object models data for use within BAM. It contains the data to be monitored. The data can be saved in raw format in a database or in an external table. The data can also be a stream from an ongoing activity.

For a general introduction to data objects, including how to add data objects to projects and view data object information, see [Working with Data Objects](#). You should know how to work with data objects as a user in the Designer role before creating data objects as a user in the Administrator role.

Process data objects in BAM are based on the process star schema. This is the star schema standard adapted for use in process analytics. Fact tables contain measures such as process running times, and dimension tables contain dimensions such as process names.

Before you create a data object, you should determine the following:

- The type of data object you need: simple, derived, external, or logical
- For a simple data object, whether you need a stream, archived stream, or archived relation
- The columns you need and whether they are measures, dimensions, or attributes
- The data types of the columns you need: VARCHAR (or String), INT, DECIMAL, FLOAT, or DATETIME
- Whether you need primary columns for joins
- Whether you need index columns
- Whether you need hierarchies
- Whether you need calculated fields

- Whether data will be filtered
- How long data will be retained
- The types of row security and permissions you will be granting to other users

See the following for information about the ready-to-use data objects provided with BAM:

- See [Integrating with Oracle Business Process Management](#) for more information on ready-to-use data objects for BPM.
- See [Integrating with Oracle SOA Suite](#) for more information on ready-to-use data objects for SOA.

A data object contains no data when you first create it. You must load or stream data into data objects using the technologies discussed in the following topics:

- Using Process Analytics in *Developing Business Processes for Oracle Business Process Management*
- Configuring BPEL Process Analytics in *Developing SOA Applications with Oracle SOA Suite*
- [Insert values into a data object](#)
- [Creating Oracle BAM Enterprise Message Sources](#)
- [Import](#)
- [Using the Loadgen Utility](#)
- [Using Oracle BAM Web Services](#)

You can view definitions (metadata) for all BAM data objects using the following WSIL interface:

`http://host:port/OracleBAMWS/WebServices/Wsil/inspection.wsil`

 **Note:**

BAM has the following system data objects, which you must not delete or change:

- `AlertHistory` — Stores alert event history.
- `BAM_ENTITY` — Stores metadata for all BAM data objects.
- `PreseedingFileHistory` — Tracks the use of preseeding files, which are templates for data object and project creation.
- `EmsStatus` — Tracks Oracle Enterprise Manager Fusion Middleware Control start and stop status.

In addition, it is strongly recommended that you do *not* change any data objects in the `oracle` folder. Doing so could break dependent entities such as the preassembled BPM dashboards and other Oracle samples. Oracle BAM also ships with the **ColorThemes** data object which you *must not* delete but *can* edit. You can enter six-digit HEX color codes prefixed with a number or hash sign, like `#FF006A`, for example. You can store up to 15 colors per theme. Blank or empty colors are allowed, even when in the middle of a row. That is, you can assign colors 1 to 3, skip 4 and 5, then assign 6-10 and stop there.

## Creating a Data Object

Oracle BAM enables you to create different data object types to maintain various kinds of data, including external data, persisting outside BAM.

This section includes the following topics:

- [Creating a Simple Data Object](#)
- [Creating a Derived Data Object](#)
- [Creating an External Data Object](#)
- [Creating a Logical Data Object](#)

See [Data Object Types](#) for an explanation of each data object type, the difference between a stream and a relation, and what it means for a data object to be archived.

 **Note:**

You must not create a data object column named “Type” or “type”. It is a reserved word and even though you will not see a warning at creation time, you will be unable to save a query with this column name.

## Creating a Simple Data Object

To create a simple data object:

1. Go to the **Administrator** page.
2. Click **Data Objects**, or right-click **Data Objects** and select the **Create** menu item.

The Data Objects dialog appears.

3. Type a **Name** and optionally edit the **Display Name**.

The **Name** is case insensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the data object is created.

The **Display Name** is case insensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

4. Select **Simple Data Object** as the Type.
5. Select **STREAM** or **RELATION** as the Continuous Query Type.
6. If you selected **STREAM**, check or uncheck the **Archived** box.  
A relation is always archived.
7. If you selected **STREAM** and checked **Archived**, supply the replay information:
  - **Replay Unit** — Select Rows, Nanoseconds, Microseconds, Milliseconds, Seconds, Minutes, Days, Weeks, Months, or Years.
  - **Replay Amount** — Specify the number of the units. For example, if the units are Rows and you type 100, then the last 100 rows are archived.

The replay information specifies the part of the immediate past that is of interest for continuous queries, realtime KPIs, and alerts.

8. Optionally, type a **Category** for the data object.  
This value can label and group similar data objects, such as BPM data objects.
9. Specify maximum numbers of columns with different data types:
  - **Number of String Columns** — The default is 25.
  - **Number of Long String Columns** — The default is 25.
  - **Number of Integer Columns** — The default is 25.
  - **Number of Float Columns** — The default is 25.
  - **Number of Decimal Columns** — The default is 25.
  - **Number of Date/Time Columns** — The default is 10.
10. Optionally, type a **Description** of the data object.
11. Click **Create**.  
The data object opens in a new tab.

## Creating a Derived Data Object

To create a derived data object:

1. Go to the **Administrator** page.
2. Click **Data Objects** and select the **Create** menu item.  
The Data Objects dialog appears.
3. Type a **Name** and optionally edit the **Display Name**.  
The **Name** is case insensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the data object is created.  
The **Display Name** is case insensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.
4. Select **Derived Data Object** as the Type.
5. Select the **Parent** data object from which to derive this data object.  
A derived data object must have the same settings for **STREAM** or **RELATION** and **Archived** as its parent.
6. For an **Archived STREAM**, supply the replay information:
  - **Replay Unit** — Select Rows, Nanoseconds, Microseconds, Milliseconds, Seconds, Minutes, Days, Weeks, Months, or Years.
  - **Replay Amount** — Specify the number of the units. For example, if the units are Rows and you type 100, then the last 100 rows are archived.

The replay information specifies the part of the immediate past that is of interest for continuous queries, realtime KPIs, and alerts.

A derived data object must have replay information less than or equal to that of the parent data object.

7. Optionally, type a **Category** for the data object.  
This value can label and group similar data objects, such as BPM data objects.
8. Optionally, type a **Description** of the data object.
9. Click **Create**.  
The data object opens in a new tab.

 **Note:**

Derived data objects retain the inherited calculation fields and hierarchies of the parent data object, but they cannot be edited.

## Creating an External Data Object

An external data object is a connection to a table in an external database. You cannot change its data using Oracle BAM.

You can export and import its metadata only using BAMCommand. See [Using BAM Command](#) for more information.

To create an external data object:

1. Configure a data source in Oracle WebLogic Server.  
See *Configuring JDBC Data Sources in Administering JDBC Data Sources for Oracle WebLogic Server*.
2. Go to the **Administrator** page.
3. Click **Data Objects** and click the **Create** icon, or right-click **Data Objects** and select the **Create** menu item.

The Data Objects dialog appears.

4. Type a **Name** and optionally edit the **Display Name**.

The **Name** is case insensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the data object is created.

The **Display Name** is case insensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

5. Select **External Data Object** as the Type.
6. Select a **Schema Type**. You can choose to create an External Fact DO or an External Dimension DO.

An external fact data object refers to a table from an external database or different schema, used as a fact data object in BAM.

An external dimension data object refers to a table from an external database or different schema, used as Dimension data object in BAM. This data object generates a simple data object as a dimension table.

The External Dimension Data Object data is copied to the generated Simple Data Object. You can alter the number of rows that populate a Simple Data Object from an External Data Object by using Oracle Enterprise Manager Fusion Middleware Control.

7. Optionally, type a **Category** for the data object.  
This value can label and group similar data objects, such as BPM data objects.
8. Optionally, type a **Description** of the data object.
9. Click **Create**.  
The data object opens in a new tab.
10. Select the **Tables** subtab.  
This subtab applies only to external data objects.
11. Select the data source you configured in step 1 from the **Select a Data Source** drop-down list.
12. Optionally type a text string in the **Table Name Filter** field to filter the selections in the **Select a Table** drop-down list. You can use the percent (%) character as a wildcard.  
For example, type %ACTIVITY% to restrict the **Select a Table** list to data sources with ACTIVITY in their names.
13. Select the desired table from the **Select a Table** drop-down list.  
The columns in the selected table appear.
14. For each column, check or uncheck the **Selected Columns** box.  
Only checked columns are visible to BAM.
15. For each column, optionally edit the **Column Name**.
16. For each column, select the **Column Type**: ATTRIBUTE (the default), DIMENSION, or MEASURE.
17. Click **Save**.

## Creating a Logical Data Object

To create a logical data object:

1. Go to the **Administrator** page.
2. Click **Data Objects** and click the **Create** icon, or right-click **Data Objects** and select the **Create** menu item.  
The Data Objects dialog appears.
3. Type a **Name** and optionally edit the **Display Name**.  
The **Name** is case insensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the data object is created.  
The **Display Name** is case insensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.
4. Select **Logical Data Object** as the Type.
5. Optionally, type a **Category** for the data object.  
This value can label and group similar data objects, such as BPM data objects.
6. Optionally, type a **Description** of the data object.
7. Click **Create**.

The data object opens in a new tab.

8. Select the **Columns** subtab.
9. Click **Add Data Object Mapping**.

A data object mapping joins two data objects.

10. Select the following:
  - **Primary Data Object** — Select the primary data object name. This will constitute the Fact Data Object within the Logical Data Object.
  - **Primary Column** — Select the internal name of the primary column the in primary data object used in the join.
  - **Foreign Data Object** — Select the foreign data object name.
  - **Foreign Column** — Select the internal name of the primary column the in foreign data object used in the join.

 **Note:**

The Primary Data Object and Foreign Data Object selection list only lists Simple and Derived Data Objects. Logical and External Data Objects are not listed.

Check the **Standalone Mode** box if you want the Logical Data Object to be composed of only one data object. In this case, only the Primary Data Object field remains.

11. Optionally, check the **Left Outer Join** box. If unchecked, the Logical Data Object will have the *Inner Join* join type.
12. To join more data objects, repeat steps 9 and 10.
13. Click **Save**.

## Adding Columns and Hierarchies to a Data Object

This chapter includes information about columns and hierarchies and outlines how you can add them to a data object.

This section includes the following topics:

- [Column Data Types](#)
- [Adding Columns to a Simple or Derived Data Object](#)
- [Adding Columns to a Logical Data Object](#)
- [Adding Index Columns](#)
- [Adding Hierarchies](#)
- [Adding Calculated Fields](#)
- [Adding Data Object Parameters](#)
- [Viewing Maximum Numbers of Columns in a Derived Data Object](#)

### Column Data Types

The data types include:

- **VARCHAR** (or String) columns contain a sequence of text characters. The maximum length that Oracle BAM 14c supports is 2000 characters.

 **Note:**

There are two set of string length in BAM - `STRING` and `LONGSTRING`. The design length range is as follows:

- `STRING` –  $0 < 256$
- `LONGSTRING` –  $257 < 2000$

You cannot switch length from one range `STRING` to another range `LONGSTRING`. You may change the length of a column within its range only.

- **INT** columns contain numbers from  $-2^{63} - 1$  to  $2^{63} - 1$ .  
Boolean columns are INT columns with a size of 1 and values of 0 for false and 1 for true.
- **FLOAT** columns are double-precision floating point numbers.  
Oracle BAM Float truncates numeric data that has very high precision. If you do not want to see loss of precision, use the Oracle BAM Decimal type (`NUMBER` in an Oracle database) with the scale you want.

A `FLOAT` in Oracle BAM is stored as an Oracle database data type `BINARY_FLOAT`. The precision is not guaranteed to be exactly the same as entered by the user, and when data flows from one system to another, the data types get mapped to each other (as for example SQL float to Java float). Oracle BAM does not do any manipulation with the data sent by the database. It is the Oracle Java Database Connectivity (JDBC) driver that loses the last few digits.

- **DECIMAL** columns include decimal points with a scale number defined. The number is stored as a string, which uses a character for each digit in the value.  
The Oracle BAM Decimal data type is stored as a `NUMBER (38, X)` in the Oracle database. The first argument, 38, is the precision, and this is hard-coded. The second argument, X, is the scale, and you can adjust this value. The scale value cannot be greater than 38.
- **DATETIME** columns are timestamps generated to milliseconds.  
A `DATETIME` field is stored as an Oracle database data type `TIMESTAMP(6)`. Depending on how the `DATETIME` field is populated, Oracle BAM may fill in the time stamp value for you. For instance, in Oracle BAM you cannot specify the value for `DATETIME` when adding a row, but if the value for `DATETIME` is specified in a `BAMCommand` import file, the specified value is added as the value of the `DATETIME` field instead of the current time.

The format of `DATETIME` values in BAM is as follows, with 24-hour time:

```
yyyy-MM-dd'T'HH:mm:ss.SSSZ
```

For example, the following `DATETIME` value represents November 12, 2013 at 12:15:52 am, plus 103 milliseconds, in the GMT -8 (Pacific) time zone:

```
2013-11-12T00:15:52.103-08:00
```

## Adding Columns to a Simple or Derived Data Object

This procedure applies to simple and derived data objects. You cannot add columns to an external data object from within Oracle BAM. A different procedure applies to logical data objects; see [Adding Columns to a Logical Data Object](#).

 **Note:**

If you delete or hide a column in a simple or derived data object, then queries, views, and other entities that depend on the data object are broken and must be edited as described in [Managing Changes in a Project](#).

**To add columns to a simple or derived data object:**

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **Columns** subtab.
5. Click **Add Column**.  
A new column appears as a row in the table, named `column_1`.
6. Click the table cell under **Column Name** and edit the name.  
The **Column Name** is case insensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 30 characters. It cannot be changed after the column is created.
7. Optionally click the table cell under **Display Name** and edit the name.  
The **Display Name** is case insensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 30 characters. It can be changed at any time.
8. Select the **Column Type**: ATTRIBUTE (the default), DIMENSION, or MEASURE.
9. Select the **Data Type**: INT (the default), VARCHAR, DECIMAL DATETIME, or FLOAT.
10. If the selected **Data Type** makes the **Size** editable, optionally type a **Size** in bytes.
11. Check **Nullable** to allow the column to have null values. The default is checked.
12. Check **Hidden** to hide the column from Designer users. The default is unchecked.
13. Check **Unique** to require each column value to be unique. The default is unchecked.  
This setting is not available for a new column in a derived data object.
14. Optionally, click the **Default Value** cell to make it editable. If the column does not have any inherent value, the default value you specify here will be used.
15. Optionally type a **Comment**.
16. Repeat steps 5 through 14 to add other columns.
17. Click **Save**.

## Adding Columns to a Logical Data Object

This procedure applies to logical data objects. You cannot add columns to an external data object from within Oracle BAM. A different procedure applies to simple and derived data objects; see [Adding Columns to a Simple or Derived Data Object](#).



**Note:**

If you delete a column from a logical data object, a query that references the deleted column continues to run until the dashboard that references the query is closed.

**To add columns to a logical data object:**

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **Columns** subtab.
5. Click **Add Logical Column**.  
A dialog with Available Columns and Selected Columns lists appears.
6. To add a column to the Selected Columns list, select the column and click the single right arrow.
7. To add all columns to the Selected Columns list, select the column and click the double right arrow.
8. To remove columns from the Selected Columns list, use the single and double left arrows.
9. When the Selected Columns list is final, click **Finish**.  
The selected columns appear as rows in the table.
10. Optionally click the table cell under **Logical Data Object Column Name** and edit the name.  
The **Logical Data Object Column Name** is case insensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 30 characters. It cannot be changed after the column is created.
11. Optionally click the table cell under **Logical Data Object Column Display Name** and edit the name.  
The **Logical Data Object Column Display Name** is case insensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 30 characters. It can be changed at any time.
12. Select the **Column Type**: ATTRIBUTE (the default), DIMENSION, or MEASURE.
13. Repeat steps 5 through 12 to add other columns.
14. Click **Save**.

## Adding Index Columns

Indexes improve performance for large data objects containing many rows. Without any indexes, accessing data requires scanning all rows in a data object. Scans are inefficient for very large data objects. Indexes can help find rows with a specified value in a column.

If the data object has an index for the columns requested, the information is found without having to look at all the data. Indexes are most useful for locating rows by values in columns, aggregating data, and sorting data.

You can add index columns to simple and derived data objects.

#### To add an index column:

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **Indexes** subtab.
5. Click **Add Index**.  
The Edit Index dialog appears.
6. Type a **Name** for the index. An `INDEX` prefix is autofilled.  
The **Name** is case insensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 30 characters. It cannot be changed after the data object is created.
7. To add a column to the Selected Columns list, select the column in the Available Columns list and click the single right arrow.
8. To add all columns to the Selected Columns list, select the columns and click the double right arrow.
9. To remove columns from the Selected Columns list, use the single and double left arrows.
10. To change the position of a column in the Selected Columns list, select it and use the Move to Top, Move Up One, Move Down One, or Move to Bottom arrows.  
The position in the Selected Columns list corresponds to the position in the index.
11. When the Selected Columns list is final, click **Finish**.  
The Edit Index dialog closes, and the index appears as a row in the table. You can edit or remove an index by clicking the icons in the Action column.
12. Click **Save**.

## Adding Hierarchies

In Oracle BAM, you can add hierarchies to data objects to provide drill paths for business views. When a Designer user selects a hierarchy for drill-down in a view, the Viewer user can drill down and up the hierarchy of data, displaying data at different levels of detail.

You can create and edit multiple, independent hierarchies in any type of data object. There are two hierarchy types: non-datetime and datetime.



#### Note:

Hierarchies cannot consist of calculated fields.

## Creating a Non-Datetime Hierarchy

### To create a non-datetime hierarchy:

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **Hierarchies** subtab.
5. Click the **Add** icon.  
The Create Hierarchy dialog appears.
6. Type a **Name** for the hierarchy.
7. Select Non-Datetime Hierarchy as the **Type**.
8. To add a column to the Selected Columns list, select the column in the Available Columns list and click the single right arrow.
9. To add all columns to the Selected Columns list, select the columns and click the double right arrow.
10. To remove columns from the Selected Columns list, use the single and double left arrows.
11. To change the position of a column in the Selected Columns list, select it and use the Move to Top, Move Up One, Move Down One, or Move to Bottom arrows.  
The position in the Selected Columns list corresponds to the position in the hierarchy.
12. When the Selected Columns list is final, click **Save**.  
The Create Hierarchy dialog closes, and the hierarchy appears as a row in the table. You can edit or remove a hierarchy by clicking the icons in the Action column.
13. Click **Save**.

## Creating a Datetime Hierarchy

### To create a datetime hierarchy:

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **Hierarchies** subtab.
5. Click the **Add** icon.  
The Create Hierarchy dialog appears.
6. Type a **Name** for the hierarchy.
7. Select Datetime Hierarchy as the **Type**.

8. Select a DATETIME column from the **Fields** drop-down list.
9. Check all the boxes that apply in the **Groups** list: YEAR, QUARTER, MONTH, WEEK, DAYOFYEAR, DAYOFMONTH, DAYOFWEEK, HOUR, MINUTE, or SECOND.

**Note:** Drilling down in business views is only supported for YEAR, MONTH, DAYOFMONTH, HOUR, MINUTE, and SECOND.

10. When the Groups list is final, click **Save**.

The Create Hierarchy dialog closes, and the hierarchy appears as a row in the table. You can edit or remove a hierarchy by clicking the icons in the Action column.

11. Click **Save**.

## Adding Calculated Fields

You can create a column with a calculation based on other columns in the data object. The calculator includes aggregate functions, string functions, and datetime functions that you combine with existing column data to create calculated fields. You can also create calculated fields based on other calculated fields.

For example, if a field called **Activity Running Time** is in milliseconds, and you prefer to see the data in minutes, you can create a calculated field named **Activity Running Time in Minutes** and give it the following calculation expression:

```
Activity Running Time (millisecs)/60000
```

You can add calculated fields to logical data objects.

### **Note:**

If you modify a calculated field in a logical data object, a query that references the field continues to run until the dashboard that references the query is closed and the query is resaved.

### To add a calculated field:

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **Calculated Fields** subtab.
5. Click the **Add** icon.

The Expression Builder dialog appears.

6. Type a **Field Name** and a **Display Name**.

The **Field Name** is case insensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the field is created.

The **Display Name** is case insensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

7. Choose a **Column Type**. Oracle BAM provides the following Column Type options for calculated fields:

- Attribute
- Measure
- Dimension

For more information on data object columns and their uses, see [Data Object Columns](#). The default selected value is Measure.

8. To help you create an **Expression**, BAM provides these components:

- **Functions** are of these types:
  - Calculate Functions include references to dates and times, comparisons, and text operations such as concatenation.
  - Control Functions include programming decision operations such as IF, THEN, and ELSE.
  - Aggregation Functions include operations on groups of numbers, such as average, sum, count, maximum, or minimum.
- **Variables** reference column data.
- **Operators** perform mathematical operations such as addition and subtraction.

Click the **Insert into Expression** button to add the function, variable, or operator you select to the expression. See [Operators and Functions for Calculated Fields](#) for more information.

 **Note:**

If you select a Data Object Parameter, it is added into the expression box within brackets if the display name for the parameter contains certain special characters. This is so the special characters aren't treated as functions.

9. When the calculated field expression is final, click **OK**.

A basic syntax check is performed and any errors are displayed.

If the calculation passes the syntax check, the Expression Builder dialog closes and the new field appears as a row in the table. You can edit or remove a calculated field by clicking the icons in the Edit and Delete columns.

10. Click **Save**.

## Adding Data Object Parameters

As an administrator, you can define parameters at the data object level inside calculated fields. This is different from project-level parameters that Designers can specify.

To create a DO-level Parameter:

1. Go to the **Administrator** page.
2. Expand the **Data Objects** node in the left panel navigator.

3. Select the data object for which you want to create parameters.
4. Click the **Calculated Fields** tab.
5. In the **Data Object Parameters** panel, click **Add Parameter**.

The table is now populated with a new parameter that you can modify.

6. To edit the name of this parameter, click the `Name` field.

The `Name` is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the parameter is created.

The `Display Name` is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

7. Check the **Required** check box to mandate a user-specified value for this parameter.

This value can be a default or a value chosen by the user.

8. Specify a `Default Value` if needed.

9. Select a **Data Type** for this parameter.

A `Data Type` can be a `String`, `Datetime`, `Integer`, `Decimal`, or `Float` value.

10. Optionally, specify a `Description` for what this parameter does.

11. Click **Save**.

## Viewing Maximum Numbers of Columns in a Derived Data Object

When you create a simple data object, you specify the maximum numbers of columns with different data types, and you cannot change these numbers later. You cannot add more columns than what you have set by default in simple data object. For example, if the simple data object has 2 columns of type `varchar`, the derived data object will inherit 2 columns of type `varchar`. In this derived data object, you can not add more than 2 columns of type `varchar`.

For derived data objects, you can view these numbers on the **General** subtab.

**To view the numbers of columns:**

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. In the **General** tab, view the maximum numbers of columns with different data types:
  - **Number of String Columns** — The default is 25.
  - **Number of Integer Columns** — The default is 25.
  - **Number of Float Columns** — The default is 25.
  - **Number of Decimal Columns** — The default is 25.
  - **Number of Date/Time Columns** — The default is 10.

## Securing a Data Object

Permissions determine who has access to the metadata of a data object or to the data as a whole. Row security determines who has access to specific types of data. Permissions and row security apply to all types of data objects.

This section includes the following topics:

- [Setting Security Permissions for a Data Object](#)
- [Setting Row Security for a Data Object](#)

### Setting Security Permissions for a Data Object

**To change permissions for a data object:**

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Right-click the data object and select the **Security** menu item.  
The data object security settings open in a new tab.
4. To add a role or group to whom you can explicitly grant or deny permissions, follow these steps:
  - a. Click the **Add** icon in the Grant Permissions or Deny Permissions table.  
The Add Application Roles, Groups, and Users dialog opens.  
See [Managing Oracle BAM Users](#) for information about how to add users to roles and groups.
  - b. Type a **Name** for the role or group you are adding.
  - c. Select from the drop-down **List**: Application Role or Group.
  - d. Click **Search** to populate the Available Members list.
  - e. To add a member to the Selected Members list, select the member and click the single right arrow.
  - f. To add all members to the Selected Members list, select the member and click the double right arrow.
  - g. To remove members from the Selected Members list, use the single and double left arrows.
  - h. When the Selected Members list is final, click **OK**.  
The Add Application Roles, Groups, and Users dialog closes, and the **Name** you specified appears in the table.
5. To remove a role or group, select the table row and click the **Remove** icon.
6. To grant or deny permissions, check or uncheck the following permissions listed in the Grant Permissions and Deny Permissions tables.
  - **Read** — Permission to view the metadata of the data object.
  - **Write** — Permission to edit the metadata of the data object. Write permission automatically includes Read permission.

- **Remove** — Permission to delete the data object.
  - **Select** — Permission to select, or read, the data in the data object.  
This is the minimum permission needed for using the data object in business queries, business views, dashboards, and other Designer entities.
  - **Delete** — Permission to delete data object rows.
  - **Update** — Permission to update or insert data object rows.
  - **Security** — Permission to set these permissions for other users for this data object.
7. Click **Save**.

## Setting Row Security for a Data Object

You can add security filters to data objects so that only specific users can view specific rows in the data object. This can be useful when working with data objects that contain sensitive or confidential information not intended for all Designer or Viewer users.

If a user has a dashboard open, and you change that user's security filter, it does not affect the currently open dashboard. If the user reopens that dashboard, it has the new security filter settings applied. Security filter settings are included in the business queries behind the dashboard views.



### Note:

To ensure that row security changes are applied to alerts, resave the alerts or restart the Oracle BAM server.

### To add a row security filter to a data object:

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **Row Security** subtab.
5. Click the **Add Default** (plus with an arrow) icon to create a default filter for all roles with no filters assigned.  
BAMDefaultFilter appears in the Name column of the row security table. You can add a filter as if it were a role.
6. Click the **Add** icon to add roles to the row security table.  
The Add Application Roles dialog appears.  
See [Managing Oracle BAM Users](#) for information about how to add users to roles.
7. Click the **Search** button to display roles in the Available Members list according to the **List** selection.  
You can filter the search by typing a partial name in the **Name** text box.

8. Select roles and click the **Move Selected Items** arrow icon to move them to the Selected Members list. To move all members, click the **Move All Items** double arrow icon.  
To remove roles, click the **Remove Selected Items** or **Remove All Items** icon.
9. When the Selected Members list is final, click **OK**.
10. Click the **Edit** icon for the Name to which to apply the security filter.  
The Edit Security Filter dialog appears. All of the functionality available in query filters is provided. See [Filtering Data](#) for more information.  
To set a default row security filter for all users with no row security filters set, edit BAMDefaultFilter, which has no initial filter set.
11. When the filter is final, click **OK**.  
The Edit Security Filter dialog closes, and the filter appears in the row in the table for the Name you selected. You can edit or remove a filter by clicking the icons in the Action column.
12. Click **Save**.

## Managing Data in Data Objects

This section contains information about modifying or removing data from a data object both manually and automatically.

This section includes the following topics:

- [Adding Rows to a Data Object Automatically](#)
- [Adding Rows to a Data Object Manually](#)
- [Editing Rows in a Data Object](#)
- [Purging Data in a Data Object](#)
- [Exporting Data from a Data Object](#)
- [Setting Data Retention in a Data Object](#)
- [Scheduling Automatic Data Purges for a Data Object](#)
- [Filtering Data in a Data Object](#)
- [Specifying Slow-Changing Dimensions for a Data Object](#)

### Adding Rows to a Data Object Automatically

To add rows to a data object automatically, you must load or stream data using the technologies discussed in the following topics:

- "Using Process Analytics" in *Developing Business Processes for Oracle Business Process Management*
- "Configuring BPEL Process Analytics" in *Developing SOA Applications with Oracle SOA Suite*
- [Insert values into a data object](#)
- [Creating Oracle BAM Enterprise Message Sources](#)
- [Import](#)
- [Using the Loadgen Utility](#)

- [Using Oracle BAM Web Services](#)

## Adding Rows to a Data Object Manually

You can manually add rows to simple and derived data objects.

### To add a row to a data object manually:

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **Data** subtab.
5. Click **Add Row**.  
A row is added to the table.
6. Click on the row and edit the values in the cells.  
For a DATETIME column, you can type a value or click the **Select Date and Time** icon. The **Select Date and Time** dialog uses the 12-hour format as 1-12 AM/PM. Its 12:00 AM is 0:00 and 12:00 PM is 12:00 of 24 hours format.
7. Click **Save**.

## Editing Rows in a Data Object

You can edit rows in simple and derived data objects.

### To edit a row in a data object:

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **Data** subtab.
5. Select a row in the table.
6. Click **Edit Row**.  
The row becomes editable.
7. Edit the values in the cells.  
For a DATETIME column, you can type a value or click the **Select Date and Time** icon to choose a date and time.
8. Click **Save**.

 **Note:**

Editing Data Object rows always requires a Save for your changes to take effect. The fix changes warning may not always appear due to UI changes applied to BAM. This is a known issue for this release.

## Purging Data in a Data Object

You can purge data in simple and derived data objects.

To perform periodic purges automatically, see [Setting Data Retention in a Data Object](#).

### To purge a data object:

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **Data** subtab.
5. Click **Purge**.  
The Purge Data Object content dialog appears.
6. Select **Purge All** or **Purge Before**.
7. If you selected **Purge Before**, type a date and time or click the **Select Date and Time** icon to choose a date and time.
8. Click **Purge**.  
A warning message appears, asking you to confirm the purge.
9. Click **Yes**.

## Exporting Data from a Data Object

You can export data from all data object types.

See [Export](#) for information on exporting data using BAMCommand.

### To export data from a data object:

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **Data** subtab.
5. If you don't want to export all the data, select the rows you want to export.
6. Click one of the export options:

- **Export All Rows to CSV** exports all the data to a `.csv` file.

 **Note:**

You can customize the number for rows to be exported using the Preferences window. For more information, see [Modifying Data Tab Preferences](#).

- **Export Selected Rows to CSV** exports the selected rows to a `.csv` file.

The file is created. If your computer has software installed that can open `.csv` files, the file is opened. If you see that there are errors with comma-separated values in the output, ensure that the cells are formatted as text cells so that the software doesn't apply its own numeric formatting to it.

7. Save the file.

## Setting Data Retention in a Data Object

You can set data retention for simple and derived data objects.

To purge data manually, see [Purging Data in a Data Object](#).

**To set data retention for a data object:**

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **Retention** subtab.
5. Type the number of days to retain data in the **Retention Days** field.  
The default value of zero retains data indefinitely.

**Note:** You can now change the data retention schedule in Oracle Enterprise Manager to any time you need using MBean properties. For more information, see [Scheduling Automatic Data Purges for a Data Object](#).

6. Click **Save**.

## Scheduling Automatic Data Purges for a Data Object

You can schedule data purges for a data object using Oracle Fusion Middleware Enterprise Manager MBean properties.

To set an automatic data purge schedule that occurs everyday:

1. Log in to the Oracle Fusion Middleware Enterprise Manager console.
2. Click the domain button on the left-upper corner of the console.
3. From the context menu, click the **System MBean browser** item.
4. From the navigation tree, expand the **oracle.beam.server** node under the **Application Defined MBeans** node.

5. Expand the **Domain:[DomainName]** node, then expand the **Config** node
6. Click the **BEAMServerConfig** node.
7. In the resulting table, find the **DO Data Purge Time** field and enter the desired time value in the *HH24:mm:ss* format.
8. Click **Apply**.

## Filtering Data in a Data Object

You can filter data only in logical data objects.

**To filter data in a data object:**

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **Filter** subtab.
5. Add filters. All of the functionality available in query filters is provided. See [Filtering Data](#) for more information.
6. Click **Save**.

## Specifying Slow-Changing Dimensions for a Data Object

For a dimension table that changes infrequently, you can check the **Slow Changing Dimension** option to improve the performance of continuous queries.

You can set this option for simple and derived data objects.

**To specify slow-changing dimensions:**

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **General** subtab.
5. Check the **Slow Changing Dimension** box.
6. Click **Save**.

 **Note:**

If a logical data object joins one data object with **Slow Changing Dimension** set and another without, and an alert is configured to fire when a row is inserted into the logical data object, make sure the row is inserted into the data object with **Slow Changing Dimension** set first, or the alert may not fire.

## Using Data Object Folders

You can organize data objects in the left navigation pane by creating folders and subfolders for them. A folder is a **Display Name** convention for grouping data objects.

You can perform the following folder operations:

- Create one or more folders by including one or more forward slashes (/) in the **Display Name** when creating or renaming a data object. Renaming means changing the **Display Name**. Text between slashes becomes names for folders and subfolders.
- Create a new data object in an existing folder by including the folder in the **Display Name**.
- Create a new data object in an existing folder by right-clicking the folder and selecting **Create**. The **Display Name** includes the folder by default.
- Move data objects in and out of folders by renaming them, which means changing their **Display Name** values.

You cannot rename a folder in one step. To rename a folder, edit the **Display Name** of each of the data objects in the folder.

- Delete all data objects in a folder by right-clicking the folder and selecting **Delete**.

Folders have no other functions. For example, you can set user permissions only on individual data objects, not on folders.

## Editing, Renaming, and Deleting Data Objects

This section outlines the different ways in which you can modify data objects.

You can edit, rename, and delete data objects just as you can any other Oracle BAM entity.

### Editing a Data Object

Use the following procedure to open, edit, and save a data object. When you edit a data object, the changes propagate to all entities that use the data object.

**To edit a data object:**

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the subtab of interest.

5. Make any desired changes and click **Save**.
6. Click the X on the data object tab to close it.

## Renaming or Moving a Data Object

Renaming a data object changes the **Display Name**.

**To rename or move a data object:**

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects in the current project are displayed in a list.
3. Click the data object name, or right-click the data object and select the **Edit** menu item.  
The data object opens in a new tab.
4. Select the **General** subtab.
5. Edit the **Display Name**.

The **Display Name** is case insensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

To move a data object into a folder, include the folder name. See [Using Data Object Folders](#).

6. Click **Save**.
7. Click the X on the data object tab to close it.

## Deleting a Data Object

When deleting a data object, you must remove referrals to the data object from dashboards, alerts, and other BAM entities that use it.

Use the following procedure to delete a data object.

**To delete a data object:**

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Data Objects**.  
All data objects are displayed in a list.
3. Click the data object icon and click the **Delete** icon, or right-click the data object name and select the **Delete** menu item.  
A dialog asks you to confirm the data object deletion.
4. Click **OK**.  
The data object disappears from the list.

## Operators and Functions for Calculated Fields

This section provides syntax and examples for operators and functions you can use in calculations when creating calculated fields in a data object.

 **Note:**

When you use an aggregation function such as an average in a calculated field, the first data point will be null, because the aggregation of no data is null.

The Sum function is only supported for the integer data type. For decimal or float data types, exact precision for values after the decimal is not guaranteed.

This section contains the following topics:

- [Operators](#)
- [Ago](#)
- [Avg](#)
- [Ceiling](#)
- [Concat](#)
- [Count](#)
- [CountDistinct](#)
- [DateAdd](#)
- [DateDiff](#)
- [DayName](#)
- [DayOfMonth](#)
- [DayOfWeek](#)
- [DayOfYear](#)
- [Else](#)
- [Floor](#)
- [Hour](#)
- [If](#)
- [IfNull](#)
- [Length](#)
- [Lower](#)
- [Max](#)
- [Median](#)
- [Min](#)
- [Minute](#)
- [Month](#)
- [MonthName](#)
- [Noop](#)
- [Now](#)
- [PercentOfTotal](#)
- [Quarter](#)

- Rank
- Repeat
- Replace
- Round
- Second
- StdDev
- StdDevsFromMean
- Substring
- Sum
- Switch
- Then
- TrimEnd
- TrimStart
- Upper
- Variance
- VariancesFromMean
- Week
- Year

## Operators

Table 14-1 describes the operators you can use to build calculated columns.

**Table 14-1 Operators Used in Calculated Fields**

Operator	Function
+ (plus sign)	Add
- (minus sign)	Subtract
* (asterisk)	Multiply
/ (slash)	Divide
% (percent sign)	Modulus
( ) (parentheses)	Parentheses determine the order of operations
!= (exclamation point and equal sign)	Logical NOT
&& (double ampersand)	Logical AND
(double pipe)	Logical OR For example <pre>if ((CallbackClientTime == NULL)    (ReceiveInputTime == NULL)) then (-1) else (CallbackClientTime-ReceiveInputTime)</pre>
== (double equal sign)	Equality

**Table 14-1 (Cont.) Operators Used in Calculated Fields**

Operator	Function
= (equal sign)	Assignment

Field names containing any special characters, such as the operators listed in [Table 14-1](#), double quotation marks, or spaces, must be surrounded with curly braces `{}`. If field names contain only numbers, letters and underscores and begin with a letter or underscore they do not need curly braces. For example, if the field name is `Sales`, the correct way to enter this in a calculation is `Sales`. However, if the field name is `Sales+Costs`, the correct way to enter this in a calculation is `{Sales+Costs}`.

Double quotes must be escaped with another set of double quotes if used inside double quotes. For example, `Length("""Hello World, "" I said")`.

## Ago

Returns an aggregated value from the current time back to a specified time. The time unit can be `YEAR`, `QUARTER`, `MONTH`, `DAY`, or `HOUR`.

Oracle Structured Query Language (SQL) supports this function. Oracle Continuous Query Language (CQL) does not support this function.

Syntax:

```
AGO(expr,time_unit,offset)
```

Example:

```
AGO(SUM(SALES),YEAR,1)
```

## Avg

Returns the average of all values for the given field. `Avg` can accept one field parameter of type `Integer`, `Float`, or `Decimal`.

Syntax:

```
Avg(Number)
```

Example:

```
Avg(Revenue)
```

## Case

Creates a `Case` clause in a `Switch` statement. `Switch` can accept fields, expressions, and values of any type as parameters.

Syntax:

```
Switch(w)
  Case(x) : (y)
  Default(z)
```

Example:

```
SWITCH (TypeOfChange)
  CASE ("Loss") : (OldNum - NewNum)
  CASE ("Gain") : (OldNum + NewNum)
  CASE ("Revert") : (OldNum)
  DEFAULT (NewNum)
```

## Ceiling

Returns the smallest integer greater than or equal to the specified value. `Ceiling(2.9)` returns **3** and `Ceiling(-2.3)` returns **-2**. `Ceiling` can accept one field parameter of type `Integer`, `Float`, or `Decimal` or a numeric value may be entered.

### Syntax:

```
Ceiling (Number)
```

### Examples:

```
Ceiling (Total)
Ceiling (3.7)
```

## Concat

Concatenates several strings into one. `Concat` can accept multiple field parameters of type `String`, or string values may be entered.

### Syntax:

```
Concat (String1, String2, ... , StringN)
```

### Example:

```
Concat ("Hello ", "World")
```

## Count

Returns a count of all non-null values. `Count` can accept one field parameter of any type.

### Syntax:

```
Count (Field)
```

### Example:

```
Count (SaleComplete)
```

## CountDistinct

Returns a count of distinct values in a field. `CountDistinct` can accept one field parameter of any type.

### Syntax:

```
CountDistinct (Field)
```

### Example:

```
CountDistinct (Salesperson)
```

## DateAdd

Adds an offset to the field value. The first parameter for `DateAdd` must be a field of type `DateTime`, and the last seven parameters maybe a field of type `Integer` or an integer value. Zeros may be used where no offset is needed.

CQL does not support this function. This function does not add the milliseconds portion of a datetime value.

Syntax:

```
DateAdd(DateTime, Years, Months, Days, Hours, Minutes, Seconds, Milliseconds)
```

Example:

```
DateAdd({Last Modified}, 0, 0, 7, 0, 0, 0, 0)  
//adds 7 days to the Last Modified value
```

```
DateAdd({Last Modified}, 0, 0, DaysToFollowup, 0, 0, 0, 0)  
//adds DaysToFolowup number of days to the Last Modified value
```

## DateDiff

Returns the amount of time between two dates or times. The time unit can be `SQL_TSI_YEAR`, `SQL_TSI_QUARTER`, `SQL_TSI_MONTH`, `SQL_TSI_WEEK`, `SQL_TSI_DAY`, `SQL_TSI_HOUR`, `SQL_TSI_MINUTE`, or `SQL_TSI_SECOND`.

CQL does not support this function.

Syntax:

```
DATEDIFF(time_unit,time1,time2)
```

Example:

```
DATEDIFF(SQL_TSI_DAY,Process Start Time,NOW())
```

## DayName

Returns the day name for a date. `DayName` accepts one field parameter of type `DateTime`.

 **Note:**

All return values are the length of the longest day name. Shorter names include trailing spaces. To remove these extra spaces, use the [TrimEnd](#) function.

Syntax:

```
DayName(DateTime)
```

Example:

```
DayName({Last Modified})
```

## DayOfMonth

Returns the day of the month for a date, in the range 1 to 31. `DayOfMonth` accepts one field parameter of type `DateTime`.

Syntax:

```
DayOfMonth(DateTime)
```

Example:

```
DayOfMonth({Last Modified})
```

## DayOfWeek

Returns the day of the week for a date, in the range 1 to 7. `DayOfWeek` accepts one field parameter of type `DateTime`.

Syntax:

```
DayOfWeek(DateTime)
```

Example:

```
DayOfWeek({Last Modified})
```

## DayOfYear

Returns the day of the year for a date, in the range 1 to 366. `DayOfYear` accepts one field parameter of type `DateTime`.

Syntax:

```
DayOfYear(DateTime)
```

Example:

```
DayOfYear({Last Modified})
```

## Default

Creates a Default clause in a Switch statement. `Switch` can accept fields, expressions, and values of any type as parameters.

Syntax:

```
Switch(w)  
  Case(x) : (y)  
  Default(z)
```

Example:

## Else

Specifies the third clause of an If-Then-Else statement. `Else` can accept fields, expressions, and values of any type as parameters.

**Syntax:**

```
If (x)
  Then (y)
  Else (z)
```

**Example:**

```
IF(Process Instance Status=="ACTIVE")
  THEN(DATEDIFF(SQL_TSI_DAY,Process Start Time,NOW()))
  ELSE(DATEDIFF(SQL_TSI_DAY,Process End Time,NOW()))
```

## Floor

Returns the largest integer less than or equal to the value of the specified field. `Floor(2.9)` returns **2** and `Floor(-2.3)` returns **-3**. `Floor` can accept one field parameter of type Integer, Float, or Decimal or a numeric value may be entered.

**Syntax:**

```
Floor(Number)
```

**Examples:**

```
Floor(Sales)
Floor(46.75)
```

## Hour

Returns the hour value in the range 0-23. `Hour` accepts one field parameter of type DateTime.

**Syntax:**

```
Hour(DateTime)
```

**Example:**

```
Hour({Last Modified})
```

## If

Specifies the first clause of an If-Then-Else statement. `If` can accept fields, expressions, and values of any type as parameters.

**Syntax:**

```
If (x)
  Then (y)
  Else (z)
```

**Example:**

```
IF(Process Instance Status=="ACTIVE")
  THEN(DATEDIFF(SQL_TSI_DAY,Process Start Time,NOW()))
  ELSE(DATEDIFF(SQL_TSI_DAY,Process End Time,NOW()))
```

## IfNull

Returns a specified value, *y*, if the test value, *x*, is null. `IfNull` accepts two parameters that can be fields of any type or values of any type. However, the two parameters must be of the same data type.

**Syntax:**

```
IfNull(x,y)
```

**Example:**

```
IfNull(Quantity, 0)
```

## Length

Returns the length of the string. `Length` accepts one parameter that can be a field of type `String`, a string value in quotes, or an expression containing strings or fields of type `String`.

**Syntax:**

```
Length(String)
```

**Example:**

```
Length(Description)  
Length("string")  
Length(Concat(Description, "Description"))
```

## Lower

Converts the string to lowercase letters. `Lower` accepts one parameter that can be a field of type `String`, a string value in quotes, or an expression containing strings or fields of type `String`.

**Syntax:**

```
Lower(String)
```

**Example:**

```
Lower(Description)  
Lower("Description")  
Lower(Concat(Description, "Description"))
```

## Max

Returns the maximum value of the specified field or expression. `Max` accepts one field parameter of any type, or another valid expression.

**Syntax:**

```
Max(x)
```

**Example:**

```
Max(Quantity)  
Max(Concat(Description, " overstock"))
```

## Median

Returns the median of all values for the given field. `Median` can accept one field parameter of type `Integer`, `Float`, or `Decimal`.

**Syntax:**

```
Median (Number)
```

**Example:**

```
Median (Revenue)
```

## Min

Returns the minimum value of the specified field or expression. `Min` accepts one field parameter of any type, or another valid expression.

**Syntax:**

```
Min (x)
```

**Example:**

```
Min (Quantity)  
Min (Concat (Description, " overstock"))
```

## Minute

Returns the minute value in the range 0-59. `Minute` accepts one field parameter of type `DateTime`.

**Syntax:**

```
Minute (DateTime)
```

**Example:**

```
Minute ({Last Modified})
```

## Month

Returns the month value for a date in the range 1-12. `Month` accepts one field parameter of type `DateTime`.

**Syntax:**

```
Month (DateTime)
```

**Example:**

```
Month ({Last Modified})
```

## MonthName

Returns the month name for a date. `MonthName` accepts one field parameter of type `DateTime`.

**Syntax:**

```
MonthName (DateTime)
```

**Example:**

```
MonthName({Last Modified})
```

## Noop

Does nothing. Use in a control function when you want one alternative to do nothing.

**Syntax:**

```
NOOP (Param)
```

**Example:**

```
IF("Process Running Time (millisecs)"/3600000 > 48)
  THEN("Overdue")
  ELSE(NOOP(1))
```

## Now

Returns the current date and time. `Now` does not accept any parameters.

**Syntax:**

```
Now()
```

**Example:**

```
DateAdd(Now(), 0, 0, 7, 0, 0, 0, 0)
```

## PercentOfTotal

Returns the percent the value represents of the total values for the specified field. `PercentOfTotal` accepts one field parameter of type `Integer`, `Float`, or `Decimal`.

CQL does not support this function.

**Syntax:**

```
PercentOfTotal (Number)
```

**Example:**

```
PercentOfTotal (Quantity)
```

## Quarter

Returns the quarter value in the range 1-4. `Quarter` accepts one field parameter of type `DateTime`.

**Syntax:**

```
Quarter(DateTime)
```

**Example:**

```
Quarter({Last Modified})
```

## Rank

Returns the rank of a value in comparison to other values in the column. Values are not consecutive when ties occur. For example, if two values have a rank of 3, the next rank is 5.

CQL does not support this function.

Syntax:

```
RANK (column)
```

Example:

```
RANK (SALES)
```

## Repeat

Repeats a string for the specified number of times. `Repeat` accepts two parameter, the first of which may be a string value or a field of type `String`, the second of which may be an integer value or a field of type `Integer`. Either parameter can use an expression that returns a string for the first parameter and an integer for the second value.

Syntax:

```
Repeat (String, Integer)
```

Example:

```
Repeat ("string", 5)  
Repeat (Description, 2)  
Repeat (Description, Quantity)  
Repeat (Concat (Description, " overstock"), Quantity+2)
```

## Replace

Returns a string, `x`, with all occurrences of the string, `y`, replaced by the string `z`. `Replace` accepts three field parameters of type `String`, or string values.

Syntax:

```
Replace (String, String, String)
```

Example:

```
Replace (Description, "ing", "tion")
```

## Round

Rounds the specified value in the first parameter to the number of decimal places specified in the second parameter, rounding up if the number in the N+1 decimal place is 5 or greater, and rounding down otherwise. `Round` accepts two parameters that can be fields of type `Integer`, `Float`, or `Decimal`, or numeric values.

Syntax:

```
Round (Number, N)
```

Example:

```
Round(Sales,2)
```

In this example, if `Sales` value is 12.345, it is rounded to 12.35.

## Second

Returns the second value in the range 0-59. `Second` accepts one field parameter of type `DateTime`.

**Syntax:**

```
Second(DateTime)
```

**Example:**

```
Second({Last Modified})
```

## StdDev

Returns the standard deviation of the values in a column.

**Syntax:**

```
STDDEV(column)
```

**Example:**

```
STDDEV(PROCESS_OPEN_HOURS)
```

## StdDevsFromMean

Returns the number of standard deviations from the mean for a particular value.

CQL does not support this function.

**Syntax:**

```
STDDEVSFROMMEAN(column)
```

**Example:**

```
STDDEVSFROMMEAN(PROCESS_OPEN_HOURS)
```

## Substring

Returns a substring `z` characters long from string `x`, starting at position `y`. `Substring` requires three parameters, the first of which must be a string value, or a field of type `String`, and the second and third of which must be an integer or field of type `Integer`.

**Syntax:**

```
Substring("source_string", start_position, substring_length)
```

**Example:**

```
Substring("some string", 6, 3)  
// returns "str"
```

## Sum

Returns a summation of all values for the specified field. `Sum` accepts one field parameter of type `Integer`, `Float`, or `Decimal`.

**Syntax:**

```
Sum(Number)
```

**Example:**

```
Sum(Total)
```

## Switch

Creates a `Switch` statement. `Switch` can accept fields, expressions, and values of any type as parameters.

**Syntax:**

```
Switch(w)
  Case(x) : (y)
  Default(z)
```

**Example:**

## Then

Specifies the second clause of an `If-Then-Else` statement. `Then` can accept fields, expressions, and values of any type as parameters.

**Syntax:**

```
If(x)
  Then(y)
  Else(z)
```

**Example:**

```
IF(Process Instance Status=="ACTIVE")
  THEN(DATEDIFF(SQL_TSI_DAY,Process Start Time,NOW()))
  ELSE(DATEDIFF(SQL_TSI_DAY,Process End Time,NOW()))
```

## TrimEnd

Trims the whitespace characters (space, tab, carriage return, line feed, page feed, form feed, and so on) from the end of the string. `TrimEnd` accepts one field parameter of type `String`. You can also enter an expression that returns a string value.

**Syntax:**

```
TrimEnd(String)
```

**Example:**

```
TrimEnd(Description)
TrimEnd(Concat(Description,Subcategory))
```

## TrimStart

Trims the whitespace characters (space, tab, carriage return, line feed, page feed, form feed, and so on) from the beginning of the string. `TrimStart` accepts one field parameter of type `String`. You can also enter an expression that returns a string value.

**Syntax:**

```
TrimStart (String)
```

**Example:**

```
TrimStart (Description)  
TrimStart (Concat (Description, Subcategory))
```

## Upper

Converts a string to uppercase letters. `Upper` accepts one parameter of type `String`. You can also enter an expression that returns a string value.

**Syntax:**

```
Upper (String)
```

**Example:**

```
Upper ({License Plate Number})
```

## Variance

Returns the variance of the values in a column.

**Syntax:**

```
VARIANCE (column)
```

**Example:**

```
VARIANCE (PROCESS_OPEN_HOURS)
```

## VariancesFromMean

Returns the variance from the mean for a particular value.

CQL does not support this function.

**Syntax:**

```
VARIANCESFROMMEAN (column)
```

**Example:**

```
VARIANCESFROMMEAN (PROCESS_OPEN_HOURS)
```

## Week

Returns the week for a `DateTime` value, in the range 0 to 53, because there might be the beginning of a week 53, where Sunday is the first day of the week. Week 1 is the first week with a Sunday in this year.

For example, in the year 2006, January 1st is a Sunday, so there is no week 0. The year starts with week 1 and continues to week 53. Week 53 of 2006 includes only one day, which is December 31st (also a Sunday). The Monday through Saturday following this (January 1-6 of 2007) are in week 0 of 2007.

**Syntax:**

```
Week(DateTime)
```

**Example:**

```
Week({Last Modified})
```

## Year

Returns the year value in the range 1000-9999. `Year` accepts one parameter of type `DateTime`.

**Syntax:**

```
Year(DateTime)
```

**Example:**

```
Year({Last Modified})
```

# Creating Oracle BAM Enterprise Message Sources

This chapter describes how to create and use Enterprise Message Sources (EMS) in Oracle Business Activity Monitoring (Oracle BAM). It explains how to use XML formatting configuration parameters, specify date and time patterns and locale information, handle errors in EMS payloads, and use Java Message Service (JMS) resources hosted on remote providers.

This chapter includes the following sections:

- [Understanding Enterprise Message Sources](#)
- [Creating Enterprise Message Sources](#)
- [Using Enterprise Message Sources](#)
- [Creating an Outbound Connection JNDI](#)
- [Updating the Oracle BAM Adapter Deployment](#)
- [Using Foreign JMS Providers](#)
- [Configuring the JCA Alerts Mechanism in EMS](#)

## Understanding Enterprise Message Sources

An Enterprise Message Source (EMS) provides direct Java Message Service (JMS) connectivity to the Oracle BAM Server. JMS is the standard messaging API for passing data between application components and allowing business integration in heterogeneous and legacy environments.

The EMS does not configure Extract Transform and Load (ETL) scenarios, but rather maps from a message directly to a data object on the Oracle BAM Server. However, you can still use XML Stylesheet Language (XSL) to perform a transformation in between. Each EMS connects to a specific JMS topic or queue, and the information is delivered into a data object in Oracle BAM. Use the Oracle BAM Composer Administrator page to configure EMS definitions.

The default JMS provider is Messaging for Oracle WebLogic Server. Oracle Advanced Queuing (AQ) JMS is supported. Tibco JMS, a non-Oracle provider, is also supported.

See [Using Foreign JMS Providers](#) for more information.

The following message types are supported:

- Map message
- Text message with XML payload

The following XML formatting options are supported for Text message transformation:

- Pre-processing
- Message specification
- Column value (Column values can be provided either as elements or attributes in the XML payload.)

## Creating Enterprise Message Sources

Each EMS connects to a specific JMS topic or queue, and the information is delivered into a data object in Oracle BAM. Use the Oracle BAM Composer Administrator page to configure EMS definitions.

This section includes the following topics:

- [Preparing to Create an Enterprise Message Source](#)
- [Creating an Enterprise Message Source: Steps](#)
- [Configuring a DateTime Specification](#)
- [Using Advanced XML Formatting](#)
- [Configuring EMS Error Handling](#)

### Preparing to Create an Enterprise Message Source

Before defining an EMS, consider these points:

- If you are using a third-party JMS application, you must configure the JMS Adapter to specify the message source connection details. Oracle BAM uses the JMS Adapter to connect. See [Using Foreign JMS Providers](#) for more information.
- If a non-Oracle WebLogic Server JMS server is used (such as Tibco) then the durable subscription name should not be repeated in any of the Enterprise Message Sources created. Some JMS servers do not allow the clients to have more than one ConnectionFactory for a single topic destination, and Oracle BAM does not support reuse of the same ConnectionFactory for the same topic.
- The JMS server (where you host queues and topics) can be configured on a different system than that which hosts the Oracle BAM Server. (For Oracle Advanced Queuing, or AQ, it is acceptable to host on the same server as Oracle BAM because the database hosts the JMS server, but for other cases it is recommended to host the JMS server on another system).
- Do not configure two Enterprise Message Sources on the same topic or queue. If you require two Enterprise Message Sources on the same Queue, each EMS must have different Message Selector value specified; otherwise, the messages are duplicated on both of the Enterprise Message Sources.
- Oracle recommends that you update no more than one Data Object with one EMS any time. This maintains the sequence of events as well as removes any collisions as a result of multiple avenues writing to the same Data Object.

Even if the source events are published from multiple sources, they can all be written to the same JMS topic or queue with the same message selector (if needed) for the Data Object.

- When you define an EMS, you specify all of the fields in the messages to be received. Some messaging systems have a variable number of user-defined fields, while other systems have a fixed number of fields.

For any string type field, you can apply formatting to that field to break apart the contents of the field into separate, individual fields. This is useful for messaging systems where you cannot create user-defined fields and the entire message body is received as one large field. The formatting specifications allow you to specify the path to a location in the XML tree, and then extract the attributes or tags as fields.

- For large XML documents, normal XSLT processing implications on performance apply. It is therefore recommended that, where possible, data transformation be done at the source, before the JMS queue bound for consumption in BAM is populated.
- To have a distributed JMS queue for an EMS write to a data object as a single transaction, the connection factory value for the outbound connection JNDI pertaining to the queue must be XA enabled (`weblogic.jms.XAConnectionFactory`). This helps the JMS sessions to be automatically enlisted in the JTA transaction. Because BAM already uses the XA data source, the JMS and Persistence APIs seamlessly participate in the XA transaction.  
JMS messages are dequeued after delivery. If a transaction commits, database changes are committed and the JMS message is removed from the queue. If a rollback occurs, the message is returned to the queue for retry.

## Creating an Enterprise Message Source: Steps

To create an EMS:

1. Go to the **Administrator** page.
2. Click **Enterprise Message Sources**, or right-click **Enterprise Message Sources** and select the **Create** menu item.

The Enterprise Message Sources dialog appears.

3. Type a **Name** and optionally edit the **Display Name**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the EMS is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

4. Click **Create**.

The EMS opens in a new tab.

5. Specify the following **Connection Information** settings:

- **Outbound Connection JNDI** — Type the JNDI name of a JMS connection factory created using the JMS Adapter. The default is `eis/bam/wls/Queue`.

For a durable subscriber, the outbound connection JNDI should have a `ClientID` in `FactoryProperties`. You can use `eis/bam/wls/DurableTopic` or examine its properties.

See [Creating an Outbound Connection JNDI](#) for information about how to create an outbound connection JNDI.

- **Topic/Queue Name** — Type the JNDI name of a JMS topic or queue. The default is `jms/demoQueue`.

- **Durable Subscriber Name** — Type the name of the subscriber, for example, `BAMFilteredSubscription`. The Durable Subscriber Name should match the event-publisher subscriber name property if it is provided.

A durable subscription can preserve messages published on a topic while the subscriber is not active. It enables Oracle BAM to be disconnected from the JMS provider for periods of time, and then reconnect to the provider and process messages that were published during the disconnected period.

- **Auto-Start** — Select Yes to specify that the EMS starts reading messages and sending them to the data object as soon as the Oracle BAM Server starts or restarts. The default value is **No**.

 **Note:**

A system generated data object called EmsStatus that monitors the status of the EMS is automatically created when the server is started. Oracle strongly recommends that you refrain from deleting this data object.

6. Specify the following **Message Configuration** settings:
  - **Source Value Formatting: Use DateTime Specification** — See [Configuring a DateTime Specification](#).
  - **Error Handling** — See [Configuring EMS Error Handling](#).
  - **JMS Message Type** — Select Text Message or Map Message.  
In a Text Message, payload contents are in XML format, and you must specify **XML Message Formatting** settings.
  - **Message Selector** — Type a single name-value pair to have the JMS provider select, or filter, messages on its behalf using application-specific criteria. When this parameter is set, the application-defined message property value must match the specified criteria for it to receive messages. To set message property values, use the `setProperty()` method of the `Message` interface.  
  
The name value pair format should be `name='value'`, for example, `message='mymessage'`. The equals sign (=) is the name-value pair separator. Currently only one name-value pair is supported.
  - **Use Thread Pool for Parallel Message Processing** — Check this box to improve performance by allowing multiple messages to be processed at once.
7. If the **JMS Message Type** is Text Message, specify the following **XML Message Formatting** settings:
  - **Message Element Name** — Type the parent element that contains column values in either its sub-elements or attributes.
  - **Message Element Namespace** — To qualify messages, optionally type the namespace URI for the message element. Only message elements containing this namespace are persisted.
  - **Batch Messages** — Check this box to batch multiple messages in a single JMS message. Specify the wrapper element in **Batch Element Name**. To qualify batched messages, optionally specify the namespace URI in **Batch Element Namespace**.
  - **XSL Formatting: Use Pre-Processing** — Check this box to apply an XSL transformation to an incoming text message prior to message retrieval and column mapping. See [Using Advanced XML Formatting](#).
  - **XML Column Value** — Select Element Tag or Attribute. Column values can be specified using either elements or attributes in an XML payload.
8. Specify the following **Source To Data-Object Field Mapping** settings:
  - **Use Self-Described Payload** — Check this box if the EMS payload contains the data-object field mapping information.

A self-described payload must specify `dataObjectName`, `operationType`, and `keys` as attributes of the root element. Keys are optional for insert operations. Specify multiple keys separated by commas. For example:

```
<msgRoot dataObjectName="DATADO" operationType="insert" keys="region">
  <region>N100</region>
  <state>DF</state>
  <sales>665</sales>
</msgRoot>
```

- **Use Persistence Payload** — Check this box if the EMS payload does not contain the data-object field mapping information.

A Persistence Payload adheres to the `DataObjectOperations` XSD, and EMS does not perform XML parsing on the payload. Each payload is passed directly to BAM Persistence for processing. This means that the payload allows any combination of operations — like insert, update, upsert, or delete — against any set of data objects. Thus, that set of multiple data objects is updated all at once.

The Persistence payload can be in POJO (EJB,JMS), XML (EJB,JMS,HTTP/REST) or JSON (EJB,JMS,HTTP/REST) format. BAM Persistence processes the operations in this collection in the order they were added. You can send a JMS message to BAM Persistence containing any of the three formats as Object or Text messages.

The following XSD shows the format of the XML schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:complexType name="column" abstract="true">
    <xsd:sequence/>
    <xsd:attribute name="name" type="xsd:string"/>
    <xsd:attribute name="predicate" type="filterPredicate"/>
    <xsd:attribute name="action" type="updateAction"/>
  </xsd:complexType>
  <xsd:complexType name="DataObjectDeleteOperation">
    <xsd:complexContent>
      <xsd:extension base="dataObjectFilteredOperation">
        <xsd:sequence/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="dataObjectFilteredOperation" abstract="true">
    <xsd:complexContent>
      <xsd:extension base="dataObjectOperation">
        <xsd:sequence>
          <xsd:element name="filter-columns" minOccurs="0">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:choice maxOccurs="unbounded">
                  <xsd:element name="string" type="stringColumn"
minOccurs="0"/>
                  <xsd:element name="integer" type="integerColumn"
minOccurs="0"/>
                  <xsd:element name="float" type="floatColumn"
minOccurs="0"/>
                  <xsd:element name="decimal" type="decimalColumn"
minOccurs="0"/>
                  <xsd:element name="date" type="dateColumn"
minOccurs="0"/>
                </xsd:choice>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
```

```

        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="DataObjectInsertOperation">
    <xsd:complexContent>
        <xsd:extension base="dataObjectOperation">
            <xsd:sequence>
                <xsd:element name="insert-columns" minOccurs="0">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:choice maxOccurs="unbounded">
                                <xsd:element name="string" type="stringColumn"
minOccurs="0"/>
                                <xsd:element name="integer" type="integerColumn"
minOccurs="0"/>
                                <xsd:element name="float" type="floatColumn"
minOccurs="0"/>
                                <xsd:element name="decimal" type="decimalColumn"
minOccurs="0"/>
                                <xsd:element name="date" type="dateColumn"
minOccurs="0"/>
                            </xsd:choice>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="dataObjectOperation" abstract="true">
    <xsd:sequence/>
    <xsd:attribute name="dataobject" type="xsd:string"/>
    <xsd:attribute name="cdc" type="xsd:boolean" use="required"/>
</xsd:complexType>
<xsd:complexType name="DataObjectOperations">
    <xsd:sequence>
        <xsd:element name="operations" minOccurs="0">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:choice maxOccurs="unbounded">
                        <xsd:element name="insert" type="DataObjectInsertOperation"
minOccurs="0"/>
                        <xsd:element name="update" type="DataObjectUpdateOperation"
minOccurs="0"/>
                        <xsd:element name="upsert" type="DataObjectUpsertOperation"
minOccurs="0"/>
                        <xsd:element name="delete" type="DataObjectDeleteOperation"
minOccurs="0"/>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DataObjectUpdateOperation">
    <xsd:complexContent>
        <xsd:extension base="dataObjectFilteredOperation">
            <xsd:sequence>
                <xsd:element name="update-columns" minOccurs="0">
                    <xsd:complexType>
                        <xsd:sequence>

```

```

                                <xsd:choice maxOccurs="unbounded">
                                  <xsd:element name="string" type="stringColumn"
minOccurs="0"/>
                                  <xsd:element name="integer" type="integerColumn"
minOccurs="0"/>
                                  <xsd:element name="float" type="floatColumn"
minOccurs="0"/>
                                  <xsd:element name="decimal" type="decimalColumn"
minOccurs="0"/>
                                  <xsd:element name="date" type="dateColumn"
minOccurs="0"/>
                                </xsd:choice>
                              </xsd:sequence>
                            </xsd:complexType>
                          </xsd:element>
                        </xsd:sequence>
                      </xsd:extension>
                    </xsd:complexContent>
                  </xsd:complexType>
                <xsd:complexType name="DataObjectUpsertOperation">
                  <xsd:complexContent>
                    <xsd:extension base="dataObjectFilteredOperation">
                      <xsd:sequence>
                        <xsd:element name="update-columns" minOccurs="0">
                          <xsd:complexType>
                            <xsd:sequence>
                              <xsd:choice maxOccurs="unbounded">
                                <xsd:element name="string" type="stringColumn"
minOccurs="0"/>
                                <xsd:element name="integer" type="integerColumn"
minOccurs="0"/>
                                <xsd:element name="float" type="floatColumn"
minOccurs="0"/>
                                <xsd:element name="decimal" type="decimalColumn"
minOccurs="0"/>
                                <xsd:element name="date" type="dateColumn"
minOccurs="0"/>
                              </xsd:choice>
                            </xsd:sequence>
                          </xsd:complexType>
                        </xsd:element>
                      <xsd:element name="insert-columns" minOccurs="0">
                        <xsd:complexType>
                          <xsd:sequence>
                            <xsd:choice maxOccurs="unbounded">
                              <xsd:element name="string" type="stringColumn"
minOccurs="0"/>
                              <xsd:element name="integer" type="integerColumn"
minOccurs="0"/>
                              <xsd:element name="float" type="floatColumn"
minOccurs="0"/>
                              <xsd:element name="decimal" type="decimalColumn"
minOccurs="0"/>
                              <xsd:element name="date" type="dateColumn"
minOccurs="0"/>
                            </xsd:choice>
                          </xsd:sequence>
                        </xsd:complexType>
                      </xsd:element>
                    </xsd:sequence>
                  </xsd:extension>
                </xsd:complexContent>

```

```
</xsd:complexType>
<xsd:complexType name="dateColumn">
  <xsd:complexContent>
    <xsd:extension base="column">
      <xsd:sequence/>
      <xsd:attribute name="value" type="xsd:string"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="decimalColumn">
  <xsd:complexContent>
    <xsd:extension base="column">
      <xsd:sequence/>
      <xsd:attribute name="value" type="xsd:decimal"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="floatColumn">
  <xsd:complexContent>
    <xsd:extension base="column">
      <xsd:sequence/>
      <xsd:attribute name="value" type="xsd:double"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="integerColumn">
  <xsd:complexContent>
    <xsd:extension base="column">
      <xsd:sequence/>
      <xsd:attribute name="value" type="xsd:long"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="stringColumn">
  <xsd:complexContent>
    <xsd:extension base="column">
      <xsd:sequence/>
      <xsd:attribute name="value" type="xsd:string"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="dataObjectOperations" type="DataObjectOperations"/>
<xsd:simpleType name="filterPredicate">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="NULL"/>
    <xsd:enumeration value="NOTNULL"/>
    <xsd:enumeration value="LIKE"/>
    <xsd:enumeration value="NOTLIKE"/>
    <xsd:enumeration value="EQ"/>
    <xsd:enumeration value="NE"/>
    <xsd:enumeration value="GT"/>
    <xsd:enumeration value="GTE"/>
    <xsd:enumeration value="LT"/>
    <xsd:enumeration value="LTE"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="updateAction">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="REPLACE"/>
    <xsd:enumeration value="INCREMENT"/>
    <xsd:enumeration value="DECREMENT"/>
  </xsd:restriction>
</xsd:simpleType>
```

```
</xsd:simpleType>
</xsd:schema>
```

 **Note:**

The datetime specification and XML Message Formatting properties are not applicable when you use the Persistence Payload option.

- **Data Object Name** — Type the name or click **Browse** to select a data object.
  - **Operation** — Select Insert, Upsert, Update, or Delete.
  - **Key** — Check this box if the field is a key.
  - **Tag/Attr Name** — Type the source tag or attribute name for the corresponding **Data Object Field**.
9. Click **Save**.
- After you save the EMS, **Edit**, **Start**, **Stop**, **Metrics**, and **Copy EMS** buttons appear, and the EMS settings are no longer editable. See [Using Enterprise Message Sources](#) for more information.

## Configuring a DateTime Specification

You can enter a date and time pattern and provide locale information.

When including a timestamp field in an EMS payload, consider the following:

- Inserting null in a timestamp field might not be in the control of a client such as EMS.
- When no value is given for a timestamp field (as in Oracle BAM), EMS assigns the current date and time.
- When the incoming timestamp value does not adhere to the `xsd:dateTime` or the datetime format specified in the EMS, the current datetime is inserted.

Note that when **Use DateTime Specification** is disabled (not checked), the incoming value must be in `xsd:dateFormat`. That is, `xsd:dateFormat ([-]CCYY-MM-DDThh:mm:ss[Z](+|-)hh:mm)` is the default format when **DateTime Specification** is not configured.

Valid value patterns for `xsd:dateTime` include:

- 2001-10-26T21:32:52
- 2001-10-26T21:32:52+02:00
- 2001-10-26T19:32:52Z
- 2001-10-26T19:32:52+00:00
- -2001-10-26T21:32:52
- 2001-10-26T21:32:52.12679

**To configure a DateTime Specification:**

1. In an EMS that you are creating or editing, check **Use DateTime Specification**.  
A datetime format link that shows the default format, **<MM/dd/yyyy HH:mm:ss.SSS>**, appears.
2. Click the datetime format link.

The DateTime Specification dialog appears.

3. Select a datetime pattern from the **Pattern** drop-down list or type a pattern.

You must supply a valid date and time pattern that adheres to the Java SimpleDateFormat. [Table 15-1](#) provides the syntax elements for SimpleDateFormat, and [Table 15-2](#) provides some examples.

When you explicitly select the `HH:mm:ss` datetime format, the default value `1/1/1970` is inserted for the date, and the EMS ignores the date value.

When you explicitly select only the date (excluding the hour, minute, and second) as the datetime format, then the date is inserted with its hour, minute, and second set to `12:00:00 AM`. The EMS ignores the time value.

4. Optionally enter the locale information:
  - **Language** — This should be a lowercase two-letter ISO-639 code, for example, `en`, `fr`, `de`, `it`.
  - **Country** — This should be uppercase two-letter ISO-3166 code, for example, `CN`, `TW`, `FR`, `DE`.
  - **Variant** — This can be vendor or browser specific code. For example, use `WIN` for Windows or `MAC` for Macintosh. Where there are two variants, separate them with an underscore, and put the most important one first. For example, a Traditional Spanish collation might construct a locale with parameters for language, country, and variant as `es`, `ES`, `Traditional_WIN`.
5. Click **OK**.

**Table 15-1 Syntax Elements for SimpleDateFormat**

Symbol	Meaning	Presentation	Example
G	Era	Text	AD
y	Year	Number	2003
M	Month	Text or Number	July; Jul; 07
w	Week in year (1-53)	Number	27
W	Week in month (1-5)	Number	2
D	Day in year (1-365 or 1-364)	Number	189
d	Day in a month	Number	10
F	Day of week in month (1-5)	Number	2
E	Day in week	Text	Tuesday; Tue
a	AM/PM marker	Text	AM
H	Hour (0-23)	Number	0
k	Hour (1-24)	Number	24
K	Hour (0-11 AM/PM)	Number	0
h	Hour (1-12 AM/PM)	Number	12
m	Minute in an hour	Number	30
s	Second in a minute	Number	55
S	Millisecond (0-999)	Number	978
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00

**Table 15-1 (Cont.) Syntax Elements for SimpleDateFormat**

Symbol	Meaning	Presentation	Example
Z	Time zone	RFC 822 time zone	-0800
'	Escape for text	Delimiter	MMM '01 -> Jul '01

The examples in [Table 15-2](#) show how date and time patterns are interpreted in the United States locale. The date and time used in all of the examples are 2001-07-04 12:08:56 local time in the U.S. Pacific Time time zone.

**Table 15-2 Date and Time Pattern Examples**

Date and Time Pattern	Result
"yyyy.MM.dd G 'at' HH:mm:ss z"	2001.07.04 AD at 12:08:56 PDT
"EEE, MMM d, 'yy"	Wed, Jul 4, '01
"h:mm a"	12:08 PM
"hh 'o'clock' a, zzzz"	12 o'clock PM, Pacific Daylight Time
"K:mm a, z"	0:08 PM, PDT
"yyyyy.MMMMM.dd GGG hh:mm aaa"	02001.July.04 AD 12:08 PM
"EEE, d MMM yyyy HH:mm:ss Z"	Wed, 4 Jul 2001 12:08:56 -0700
"yyMMdHHmmssZ"	010704120856-0700
"yyyy-MM-dd'T'HH:mm:ss.SSSZ"	2001-07-04T12:08:56.235-0700

## Oracle AQ JMS DateTime Considerations

If you are sending datetime data to an Oracle BAM EMS through Oracle AQ JMS, the following must be considered while configuring a DateTime Specification.

The default datetime formats in Oracle database are specified either explicitly, with the NLS session parameters `NLS_DATE_FORMAT`, `NLS_TIMESTAMP_FORMAT`, and `NLS_TIMESTAMP_TZ_FORMAT`, or implicitly, with the NLS session parameter `NLS_TERRITORY`.

If trigger processing code (PL/SQL) does not override the date format with explicit formatting, the dates are formatted according to the format specified by NLS parameters for the database session, and sent accordingly to EMS. This situation means that the EMS DateTime Specification must have the equivalent format of NLS parameters to parse and interpret the incoming data.

However, problems arise on the EMS side if a database administrator changes the NLS parameters. It is always safe to use explicit formatting using the `to_char()` function, rather than rely on the default NLS parameters specified format.

A line such as this example in Trigger processing code

```
'<HIREDATE>' || :new.HIREDATE || '</HIREDATE>' ||
```

should be changed to something similar to

```
'<HIREDATE>' || to_char(:new.HIREDATE, 'MM/dd/yy HH24:MI:SS') || '</HIREDATE>' ||
```

The corresponding format selected from EMS DateTime Specification drop-down list is `MM/dd/yy H:mm:ss`.

Similarly, for timestamp data, if the format selected on the database side with the `to_char()` function is `MM/dd/yy HH24:MI:SS.FF`, the corresponding EMS DateTime Specification format is `MM/dd/yy H:mm:ss:SSS`.

## Using Advanced XML Formatting

The Advanced formatting options allow an EMS to contain a user-supplied XSL Transformation (XSLT) for each formatted field in the message.

Uses for XSLT include:

- Handling of hierarchical data. The Data Flow does not handle hierarchical data. The XSLT can flatten the received XML into a single record with repeating fields.
- Handling of message queues that contain messages of multiple types in a single queue. The Data Flow requires that all records from the Message Receiver be of the same schema. The EMS output can be defined as a combined superset of the message schemas that are received, and the XSL transformation can identify each message type and map it to the superset schema as appropriate.
- Handling of XML that, while not expressing hierarchical data, does contain needed data at multiple levels in the XML. EMS formatting can only read from one level with the XML. The XSL transformation can identify the data needed at various levels in the input XML and output it all in new XML that contains all of the data combined at one level.

**To specify an XSL transformation:**

1. In an EMS that you are creating or editing, check **XSL Formatting: Use Pre-Processing**.
2. Click **Advanced Formatting Options**.

The XSL Transformation dialog opens.

3. In the **Enter XSL Transformation Code** field, type or paste the XSL markup for the transformation. You can write the XSL markup in another editing tool and then copy and paste the code into this dialog.
4. In the **Sample XML to Transform** field, type sample XML to test the transformation against. The sample XML is not saved in this dialog and is not displayed if you close and reopen this dialog.
5. Click **Verify XSL Syntax** to validate the XSL syntax.
6. Click **Test transformation on sample XML** to test your transformation.

The results are displayed in the **Results** field, including any errors in the XSL syntax, the sample XML syntax, or the transformation.

7. Click **OK**.

## Configuring EMS Error Handling

Configure the **Error Handling** settings to specify how you want EMS payload errors to be handled. You can configure how to handle errors in EMS payloads in these ways: write the error message to a log file, insert the error message into a data object, or publish the error message to a JMS topic or queue.

Error handling occurs after rollback has occurred and after all JMS message retries have failed.

**To configure fault handling:**

1. In an EMS that you are creating or editing, check **Log faulted messages** to send error messages to the log, or check **Write faulted messages** to write error messages to an Oracle BAM data object or a JMS topic.  
You can select both options to log and write faulted messages.
2. If you selected **Log faulted messages**, you can select **Include payloads** to include the payloads with the error messages in the log.
3. If you selected **Write faulted messages**, select **To Data Objects** to insert the messages in a data object, or select **To JMS Queue/Topic** to write the messages to JMS. Then use [Table 15-3](#) to provide additional information for the option you selected.

**Table 15-3 Parameters for Write Faulted Messages Options**

Option	Parameter	Description
To Data Object	Error Data Object	The data object name to which to write the error message.
To Data Object	Data Object Field	The field in the data object to which to write the error message. Be sure to select a string field that is long enough to accommodate the longest error payload.
To JMS Queue/Topic	Initial Context Factory	The initial context factory to be used for looking up specified JMS connection factory or destination. For example: weblogic.jndi.WLInitialContextFactory
To JMS Queue/Topic	JNDI Service Provider URL	Configuration information for the service provider to use. Used to set <code>javax.naming.Context.PROVIDER_URL</code> property and passed as an argument to <code>initialContext()</code> . An incorrect provider URL is the most common cause of errors. For example: t3://localhost:7001
To JMS Queue/Topic	Topic/Queue Connection Factory	The name used in a JNDI lookup of a previously created JMS connection factory. For example: jms/QueueConnectionFactory
To JMS Queue/Topic	Topic/Queue Name	The Topic/Queue used to post the error message. For example: jms/demoQueue jms/demoTopic
To JMS Queue/Topic	JNDI Username	The identity of the principal for authenticating the JNDI service caller. This user must have RMI login permissions.
To JMS Queue/Topic	JNDI Password	The identity of the principal for authenticating the JNDI service caller.
To JMS Queue/Topic	JMS Username (Optional) JMS Password (Optional)	You can optionally provide this information when a new JMS connection is created by a connection factory. Used to authenticate a connection to a JMS provider for either application-managed or container-managed authentication.

The following example is the format of a published message:

```
EMS <<EMS Name>> failed to process the payload: <<payload>> with the following exception:
<<Exception details>>.
```

For example:

```
EMS MyInsertEMS, failed to process the payload: <testems><test>abcd</test1>  
with the following exception:  
The end-tag for element type "test" must end with a '>' delimiter.
```

Use caution while designing the fault handling when the error message is pushed to a JMS topic or queue. If this topic or queue is in turn configured for another (or the same) EMS, then that EMS pulls the same message again, which fails recursively. Although Oracle BAM has taken care of the message by encoding that message with a CDATA, there might be other issues such as SQL exceptions that might fail recursively.

## Using Enterprise Message Sources

You can modify EMS attributes to suit your organization's needs.

This section includes the following topics:

- [Editing an Enterprise Message Source](#)
- [Starting and Stopping an Enterprise Message Source](#)
- [Monitoring Enterprise Message Source Metrics](#)
- [Copying an Enterprise Message Source](#)
- [Securing an Enterprise Message Source](#)
- [Deleting an Enterprise Message Source](#)

### Editing an Enterprise Message Source

**To edit an EMS:**

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Enterprise Message Sources**.  
All EMS objects in the current project are displayed in a list.
3. Click the EMS name, or right-click the EMS name and select the **Edit** menu item.  
The EMS opens in a new tab.
4. If the EMS has been saved previously, click the **Edit** button to make the settings editable.
5. Make any desired changes and click **Save**.
6. Click the X on the EMS tab to close it.

### Starting and Stopping an Enterprise Message Source

To open the EMS, see [Editing an Enterprise Message Source](#).

Use the **Start** and **Stop** buttons on an EMS definition tab to start and stop the EMS. In the **Stopped** status, the consumer is inactive.

Starting an EMS subscribes it and stopping an EMS unsubscribes it.

By default, the EMS starts when the Oracle BAM Server is started.

Click **Edit** to change the **Auto-Start** setting.

**Note:**

The default value for EMS Auto-Start is **No**.

You can also start and stop an EMS using `bamcommand` like so:

```
bamcommand -cmd start -name <ems_name> -type ems
```

```
bamcommand -cmd stop -name <ems_name> -type ems
```

## Monitoring Enterprise Message Source Metrics

To open the EMS, see [Editing an Enterprise Message Source](#).

Use the **Metrics** button on an EMS definition tab to monitor selected EMS statistics. The Metrics dialog displays the **Server Name**, **Messages Sent**, **Messages Persisted**, and **Messages Lost** counters. These values are accumulated since last start or reset.

**Server Name** indicates the name of the server on which EMS is currently running. This facilitates the identification of the node at which EMS is running in cluster environments.

**Messages Lost** is calculated by subtracting Messages Persisted from Messages Sent.

Click **Refresh** to see these latest counter values.

Click **Reset** to set counter values to zero.

**Note:**

If you want better throughput and do not need Metrics data, you can set the JVM argument to `-Dems.metrics.required=false` at server start. After specifying the flag, you will not see the message count in Metrics and because the state is not maintained, you can expect a better performance.

## Copying an Enterprise Message Source

To copy an EMS:

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Enterprise Message Sources**.  
All EMS objects in the current project are displayed in a list.
3. Click the EMS name, or right-click the EMS name and select the **Edit** menu item.  
The EMS opens in a new tab.
4. Click **Copy EMS**.  
The Copy EMS dialog appears.
5. Type a **Name** and a **Display Name**.

The **Name** is case sensitive, must begin with a letter, and may consist only of letters, numbers, and the underscore character. It may have up to 128 characters. It cannot be changed after the EMS is created.

The **Display Name** is case sensitive and may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters. It can be changed at any time.

6. Click **OK**.

The EMS copy opens in a new tab.

7. Follow the steps in [Creating an Enterprise Message Source: Steps](#) beginning with step 5.

## Securing an Enterprise Message Source

### To change security settings for an EMS:

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Enterprise Message Sources**.  
All EMS objects in the current project are displayed in a list.
3. Right-click the EMS and select **Security** from the pop-up menu.  
The security tab for the EMS opens.
4. To add a role or group to whom you can explicitly grant or deny permissions, follow these steps:
  - a. Click the **Add** icon in the Grant Permissions or Deny Permissions table.  
The Add Application Roles, Groups, and Users dialog opens.  
See [Managing Oracle BAM Users](#) for information about how to add users to roles and groups.
  - b. Type a **Name** for the role or group you are adding.
  - c. Select from the drop-down **List**: Application Role or Group.
  - d. Click **Search** to populate the Available Members list.
  - e. To add a member to the Selected Members list, select the member and click the single right arrow.
  - f. To add all members to the Selected Members list, select the member and click the double right arrow.
  - g. To remove members from the Selected Members list, use the single and double left arrows.
  - h. When the Selected Members list is final, click **OK**.  
The Add Application Roles, Groups, and Users dialog closes, and the **Name** you specified appears in the table.
5. To remove a role or group, select the table row and click the **Remove** icon.
6. To grant permissions, select **Read**, **Write**, **Remove**, or **Security** for the users, roles, and groups listed in the Grant Permissions table.
7. To deny permissions, select **Read**, **Write**, **Remove**, or **Security** for the users, roles, and groups listed in the Deny Permissions table.
8. Click **Save**.

## Deleting an Enterprise Message Source

You must stop an EMS before you can delete it. See [Starting and Stopping an Enterprise Message Source](#).

### To delete an EMS:

1. Go to the **Administrator** page.
2. In the left navigation pane, click the arrow to the left of **Enterprise Message Sources**.  
All EMS objects in the current project are displayed in a list.
3. Click the EMS icon and click the **Delete** icon, or right-click the EMS name and select the **Delete** menu item.  
A dialog asks you to confirm the EMS deletion.
4. Click **OK**.  
The EMS disappears from the list.

## Creating an Outbound Connection JNDI

You configure an outbound connection JNDI in the Oracle WebLogic Server Administration Console.

### To create an outbound connection JNDI:

1. Access the Oracle WebLogic Server console using a URL that includes the Oracle WebLogic Server hostname and the console port:  

```
http://wls-hostname:console-port/console
```

  
For example:  

```
http://localhost:7011/console
```
2. Log in using the Oracle WebLogic Server admin username and password.
3. In the left pane of the Oracle WebLogic Server Administration Console, select **Deployments**.
4. On the Summary of Deployments page, Control tab, expand the **BamServer** node.
5. Click the **BeamAdapter.rar** link.
6. On the Settings for BeamAdapter.rar page, click the **Configuration** tab and the **Outbound Connection Pools** subtab.
7. Expand the **oracle.tip.adapter.jms.IJmsConnectionFactory** node.
8. Click **New**.
9. Select **oracle.tip.adapter.jms.IJmsConnectionFactory** and click **Next**.
10. Type the **JNDI Name** and click **Finish**.  
The JNDI name for an EMS must begin with `eis`. An example JNDI name is `eis/bam/wls/TestQueue`.
11. Click **OK** to save the outbound connection JNDI with a deployment plan.  
The default path is `FMW_HOME/soa/bam/Plan.xml`.

12. When the Settings for BeamAdapter.rar page reappears, click the **Configuration** tab and the **Outbound Connection Pools** subtab again.
13. Expand the **oracle.tip.adapter.jms.JmsConnectionFactory** node again.
14. Click the name of the outbound connection JNDI that you created.

The Outbound Connection Properties page appears.

15. Specify each of the following properties by clicking the **Property Value** table cell, typing the value, and pressing **Enter**.
  - **ConnectionFactoryLocation** — Type the value for your JMS provider. For Oracle WebLogic Server, enter `weblogic.jms.ConnectionFactory`, or your custom connection factory details.
  - **FactoryProperties** — Specify the following properties in the following format:

```
InitialContextFactory=value;JNDIServiceProviderURL=value;
JNDIUserName=value;JNDIPassword=value
```

For example:

```
java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory;
java.naming.provider.url=t3://www.example.com:7001;
java.naming.security.principal=weblogic;
java.naming.security.credentials=weblogic1
```

Property descriptions are as follows:

- **InitialContextFactory** — The initial context factory to be used for looking up specified JMS connection factory or destination.
  - **JNDIServiceProviderURL** — The configuration information for the service provider. Used to set the `javax.naming.Context.PROVIDER_URL` property and passed as an argument to `initialContext()`.
  - **JNDIUserName** — The identity of the principal for authenticating the JNDI service caller. This user must have RMI login permissions. Used to set `javax.naming.Context.SECURITY_PRINCIPAL` and passed to `initialContext()`.
  - **JNDIPassword** — The identity of the principal for authenticating the JNDI service caller. Used to set `javax.naming.Context.SECURITY_CREDENTIALS` and passed to `initialContext()`.
  - **IsTopic** — Type `true` for a Topic or `false` for a Queue.
  - **Password and Username** — Type the credentials used to authenticate a connection to a JMS provider for either application-managed or container-managed authentication.
16. Click **Save**.
  17. Re-start WLS and BAM servers.

 **Note:**

You must update the adapter deployment if you change Oracle BAM adapter properties. For more information, see [Updating the Oracle BAM Adapter Deployment](#).

## Updating the Oracle BAM Adapter Deployment

When you deploy a SOA composite using a new JNDI, its name must be specified in the JCA file and be referable on the weblogic server (WLS) before you deploy the composite so BAM Adapter properties are updated.

To update Oracle BAM Adapter deployments:

1. Log in to the administration console.
2. Click the **Lock and Edit** button.
3. In Deployments, check the box next to *bamjcaadapter*.
4. Click **Update**, then check the **Redeploy this application using the following deployment files:** button that appears in the next screen.
5. Click **Finish**.

## Using Foreign JMS Providers

Oracle WebLogic Server can integrate non-Oracle WebLogic Server (foreign) JMS providers with applications deployed in it. One such application is Oracle BAM.

Foreign JMS providers have their own JMS client and Java Naming and Directory Interface (JNDI) Client APIs. Some configuration must be done to identify these dependencies and make these APIs available on Oracle WebLogic Server so that JMS resources hosted on a remote provider can be looked up by applications deployed in Oracle WebLogic Server.

See "Oracle JCA Adapter for JMS" in *Understanding Technology Adapters* and "Configuring Foreign Server Resources to Access Third-Party JMS Providers" in *Administering JMS Resources for Oracle WebLogic Server* for more information.

The high level configuration steps are:

1. Make the JMS and JNDI client library of the foreign provider available to applications deployed on Oracle WebLogic Server.

Identify the JMS and JNDI client Java Archive (JAR) files of the foreign provider and place them in the `WLS_HOME/lib` directory.

2. Create a foreign server using Oracle WebLogic Server Administration Console.

Go to **JMS Modules** in Oracle WebLogic Server Administration Console, and create a new module.

Inside this module, click **New**, select **Foreign Server**, and create a new foreign server by navigating through all the pages.

Provide appropriate JNDI properties for the remote provider for the foreign server definition.

3. Create JMS resources (that is, connection factories and destinations) for the foreign JMS server.

Inside the **Foreign Server** link, select the **Destination** tab and create links for

- Remote ConnectionFactory
- Remote Destination (Queue/Topic)

Local JNDI names configured for these destinations must be used while configuring EMS to consume messages from these destinations.

4. Configure an EMS definition on the Administrator page of BAM Composer to consume messages from foreign destinations.

The whole process of accessing JMS resources hosted on foreign providers is transparent to Oracle BAM Server. After the previous steps have been followed correctly, remote destinations from foreign JMS providers are published on the local WL server JNDI tree, so that applications deployed on the server (like Oracle BAM EMS) can look them up, just like any other collocated Oracle WebLogic Server JMS resource. Oracle WebLogic Server takes care of communicating with the appropriate foreign JMS provider at runtime.

## Configuring the JCA Alerts Mechanism in EMS

An Oracle BAM EMS instance starts the JCA Service in a background thread. If JCA encounters any exception such as a Remote Queue failure, the JCA Service sends an alert exception to BAM EMS.

Oracle BAM now provides for an alert mechanism that can notify you about these JCA exceptions.

To address this, Oracle BAM ships with an `EMSHistory` data object. It is automatically created when you start the server. JCA service exceptions are captured by this data object, which keeps the EMS instance running. If you would like to make the EMS instance stop, you can configure it accordingly.

To configure a JCA alert mechanism:

1. In Oracle BAM Composer, create a new project or select an existing project. For more information on creating a project, see [Creating a Project](#).
2. Add the **EmsHistory** data object to the project.
3. Create an alert. See [Creating an Alert](#) for more information.
4. Select the **When a data field in a data object meets specified conditions** Event Trigger.
5. Click the **<this data field has a condition of x>** condition and select the **EmsHistory** data object.
6. Add a row filter with the conditions **EMS NAME** and **is not null** and click **OK**.
7. Click **<select user>** and assign an appropriate user role.
8. Click the **<add action>** link, select **Send a message via email**, then complete all required fields in the Message Editor.
9. Click **OK**.

# Monitoring and Managing Oracle BAM

This chapter describes how to monitor and manage Oracle Business Activity Monitoring (Oracle BAM) to keep the system running optimally.

This chapter includes the following sections:

- [Understanding Oracle BAM Monitoring and Management](#)
- [Monitoring Continuous Queries](#)
- [Monitoring Viewsets](#)
- [Monitoring the State of the BAM Server and Services](#)
- [Configuring BAM Server MBean Properties](#)
- [Using the BAM Diagnostic Framework](#)
- [Monitoring Oracle BAM Performance](#)
- [Tuning Oracle BAM Performance](#)
- [Monitoring Oracle BAM Logs](#)
- [Monitoring Oracle BAM Web Services](#)
- [Managing Oracle BAM Availability](#)
- [Configuring Oracle User Messaging Service](#)

## Understanding Oracle BAM Monitoring and Management

Oracle BAM is monitored using the BAM Composer, the Oracle Enterprise Manager Fusion Middleware Control, and Oracle User Messaging Service.

You monitor continuous queries and viewsets on the Administrator page in BAM Composer.

You monitor and manage several aspects of Oracle BAM using Oracle Enterprise Manager Fusion Middleware Control, including MBean properties, performance metrics, log files, web services, and the Oracle User Messaging Service.

You enable the BAM Diagnostic Framework by setting three MBean properties in Fusion Middleware Control. To view the diagnostic information, you use the diagnostic data objects as you would any BAM data objects.

## Monitoring Continuous Queries

Use the Continuous Query Monitoring page to monitor all continuous queries across different projects in a BAM server instance.

See [Creating Business Queries](#) for more information about continuous queries.

**To monitor continuous queries:**

1. Go to the **Administrator** page.

2. Click **Continuous Queries Monitoring**.

The Continuous Queries Monitoring tab opens.

3. Select a **Project** or accept the default of All to monitor queries for all projects.

4. Select a **Status** or accept the default of All to monitor queries with all statuses. Statuses are:

- Registering — A newly saved query is being registered.
- Deactivated — A query remains in BAM but does not fetch data.
- Activating — A newly saved or activated query is becoming active.
- Active — A query is fetching data.
- Deactivating — A query is becoming inactive.
- Dropping — A query is being dropped (deleted) from BAM.
- Pre-migrating — A query is being prepared for migration to another BAM server.
- Migrating — A query is being migrated to another BAM server.

5. Examine the information in the columns of the Query Details table:

- Project — The name of the BAM project to which the query belongs.
- Query — The name of the query.
- Status — The status of the query, such as Active, Deactivated, and so on.
- Server — The name of the BAM server on which the query runs.
- Statement — The SQL statement for the query.

6. To activate a deactivated query, check the box for the query row and click **Activate Query**. Continuous queries are active by default.

7. To deactivate an active query, check the box for the query row and click **Deactivate Query**.

8. To drop a query and delete it from the BAM project, check the box for the query row and click **Drop Query**.

9. To update the Query Details table, click **Refresh**.

10. Click the X on the Continuous Queries Monitoring tab to close it.

## Monitoring Viewsets

A master *viewset* holds the query results in a data snapshot and the changes since the last snapshot for business views in the same project that use the same query and row security filters.

The first view opened in a dashboard for a given query creates the master viewset. Opening other views that use the same query creates slave viewsets.

Viewsets minimize time-consuming query reexecution and ensure that views are incrementally updated using a push-based mechanism.

You should monitor the number of currently open master viewsets and the number of slave viewsets sharing each master. This can help you determine the current load on the Oracle BAM system.

**To monitor viewsets:**

1. Go to the **Administrator** page.
2. Click **Viewset Monitoring**.  
The Viewset Monitoring tab opens.
3. Select a **Project** or accept the default of All to monitor viewsets for all projects.
4. Examine the information in the columns of the Viewset Details table:
  - Project — The name of the BAM project to which the viewset belongs.
  - Query — The name of the query on which the viewset is based.
  - Viewset ID — The unique ID for the viewset.
  - CQL Query — The name of the CQL query on which the viewset is based.
  - Viewset Type — The viewset type: Active, Tactical KPI, or RealTime KPI.
  - Expiry Time (secs) — The time left before the viewset expires unless pinged by BAM data control.
  - Last Pinged — The last time BAM data control pinged this viewset.
  - Message Selector — The common message selector filter on the JMS topic to which the BAM report cache publishes and BAM data control subscribes, which identifies viewset IDs.
5. To close a viewset and delete it from the BAM project, check the box for the viewset row and click **Close Viewset**.
6. To update the Viewset Details table, click **Refresh**.
7. Click the X on the Viewset Monitoring tab to close it.

## Monitoring the State of the BAM Server and Services

The Oracle BAM Server and its services can be monitored to troubleshoot any operations issues in real time.

You can view the current state of the Oracle BAM server and its services at this URL:

```
http://host:port/oracle/bam/server
```

For example:

```
http://www.example.com:7001/oracle/bam/server
```

The default Oracle BAM server port is 7001. This may be different from the Oracle WebLogic Server administration server port.

The Description table displays the following information:

- Server Name — The name of the Oracle WebLogic Server managed server dedicated to Oracle BAM, for example `bam_server1`.
- Server State — The current state of the server, such as Running or Stopped.
- Server Version — The release version of the BAM server.
- Build Number — The build number of the BAM server.

- **Build Label** — The build label, which is the name of the deployment corresponding to the Oracle BAM application on the Oracle WebLogic Server managed server.
- **Build Date** — The build date of the BAM server.
- **Schema Version** — The schema version of the BAM server. The schema controls the structure of BAM data objects.

The Service Status table displays the states, such as Running or Stopped, of the Persistence, Report Cache, Alerts Engine, and Enterprise Message services.

Clicking the **Persistence** link displays the number of Inserts, Updates, and Deletes and the number of sent and received notifications for those operations. Click the browser **Back** button to return to the Description and Service Status tables.

 **Note:**

BAM Composer has limited functionality on session replication. For example, in a clustered server environment, if Server1 is responding to user X's requests, user X has opened several tabs and pop-ups. On failure of Server1, Server2 should respond to further requests from user X and entire session objects like http session, opened tabs, and pop-ups should work seamlessly. But due to limited functionality, only http session is replicated to Server2, opened tabs and popup are closed and application displays the Home page. Subsequent requests from user X are handled by Server2. On clicking any user interface elements during session replication time, the end user may get 500 or 503 errors.

Refresh the page to proceed further.

## Configuring BAM Server MBean Properties

Use Fusion Middleware Control to configure MBean properties for the Oracle BAM server.

**To set BAM MBean properties:**

1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the Business Activity Monitoring node.
3. Double-click the BamServer node under the Business Activity Monitoring node.

For example, the BamServer node might be `BamServer (bam_server1)`.

The BAM Properties page appears, displaying BAM System Settings and Alert Engine properties.

4. To display other BAM properties, click **More BAM Advanced Configuration Properties** at the bottom of the page.

The System MBean Browser opens, displaying BEAMServerConfig properties on the Attributes tab. All properties have descriptions and most have values.

5. Edit the **Value** of each property you wish to change, then click **Apply**.

To undo all unapplied changes, click **Revert**.

6. Restart the Oracle WebLogic Server admin server and managed servers. For more information, see Starting and Stopping Servers in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

## Using the BAM Diagnostic Framework

You enable the BAM Diagnostic Framework by setting three MBean properties in Fusion Middleware Control. To view the diagnostic information, you use the diagnostic data objects as you would any BAM data objects.

The following BAM MBean properties control the BAM diagnostic framework:

- **DiagnosticEnabled** — Enables the diagnostic framework if set to `true`. The default is `false`.
- **DiagnosticLevel** — Sets the diagnostic logging level to `INFO` or `DEBUG`. The `DEBUG` level gives granular diagnostic logs. The default is `INFO`.
- **DiagnosticComponents** — Specifies the BAM components for which diagnostic messages are logged. Specify a comma-separated list of any of the following components:
  - `ADF` — The Application Development Framework, which provides lower-level components upon which Oracle BAM is based.
  - `DC` — Data Control, which manages how data flows into data objects and queries.
  - `RC` — The Report Cache, which receives and caches query results, then pushes the results to dashboards and the alert service.
  - `CQS` — The Continuous Query Service, which delivers continuous query results.
  - `ALERTS_ENGINE` — The Alert Service, which compares query results to alert events and conditions and executes the corresponding alert actions and notifications.
  - `COMPOSER` — BAM Composer, the browser-based BAM interface.
  - `COMMON_QUERY` — The Common Query Service, which delivers non-continuous query results.

By default, all components are included.

To set these properties, see [Configuring BAM Server MBean Properties](#).

All diagnostic messages are saved to `BASE_METRICS`, a simple data object. Granular details pertaining to the Report Cache, the Alert Service, and the Continuous Query Service are persisted to the module-specific derived data objects `RC_METRICS`, `ALERTS_METRICS`, and `CQS_METRICS`. These data objects are created when the diagnostic framework is enabled.

You can use these diagnostic data objects in the same way you use other data objects. For example, you can create queries, views, alerts, and dashboards to monitor the diagnostic data.

## Monitoring Oracle BAM Performance

The performance of Oracle BAM is reflected in metrics and statistics. There are Performance Summary pages with appropriate metrics for Oracle BAM Server.

**To monitor BAM performance:**

1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the Business Activity Monitoring node.
3. Right-click the `BamServer` node under the Business Activity Monitoring node.

For example, the `BamServer` node might be `BamServer (bam_server1)`.

4. Select the **Monitoring > Performance Summary** menu item.  
The Performance Summary page appears.
5. Click **Show Metric Palette** to display the Metric Palette.
6. Click the arrow to the left of a metric folder to expand the folder. Keep expanding folders until you see individual metrics with check boxes.
7. Check each metric you wish to display. Charts for each selected metric appear in the Performance Summary pane.

Descriptions of each metric are available in the *Oracle Enterprise Manager Oracle Fusion Middleware Metric Reference Manual*.

## Tuning Oracle BAM Performance

If Oracle BAM is running more slowly than expected, you can tune it in these ways:

- If throughput of data into a data object from an Enterprise Message Source or other source is slow, create an index column for the primary key column. See [Adding Index Columns](#) for more information.
- If Oracle BAM is running slowly and out-of-memory exceptions occur, adjust `USER_MEM_ARGS` to increase the memory allocated. Use the following command (in csh) with the `-Xms2048m` and `-Xmx2048m` arguments:

```
setenv USER_MEM_ARGS "-Xms2g -Xmx2g -XX:NewSize=1g -XX:+CMSConcurrentMTEnabled -  
XX:+UseConcMarkSweepGC"
```

For more tips on how to tune Oracle BAM performance, see "Tuning Oracle Business Activity Monitoring" in *Tuning Performance*.

## Monitoring Oracle BAM Logs

Use Fusion Middleware Control to monitor Oracle BAM logs.

**To monitor BAM logs:**

1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the Business Activity Monitoring node.
3. Right-click the BamServer node under the Business Activity Monitoring node.  
For example, the BamServer node might be `BamServer (bam_server1)`.
4. Select the **Logs > View Log Messages** menu item.  
The Log Messages page appears.
5. To display a list of BAM log files, click **Target Log Files**.  
The Log Files page appears.
6. To View a log file, select the file and click **View Log File**.  
The View Log File page appears.
7. To view specific types of messages, specify search criteria and click **Search**.

## Monitoring Oracle BAM Web Services

Oracle BAM provides `DataObjectDefinition`, `DataObjectOperationsByName`, `DataObjectOperationsById`, `DataObjectOperations10131`, and `ManualRuleFire` web services.

For more information, see [Using Oracle BAM Web Services](#).

Use Fusion Middleware Control to monitor Oracle BAM web services.

### To monitor BAM web services:

1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the WebLogic Domain node.
3. Expand the node for the domain under the WebLogic Domain node.  
For example, the domain node might be `soainfra`.
4. Right-click the `BamServer` node under the domain node.  
For example, the `BamServer` node might be `bam_server1`.
5. Select the **Web Services** menu item.  
The Web Services page appears.
6. Select the **Java EE** tab.

You can see the following statistics in the Java EE Web Services table: **Invocation Count**, **Response Count**, **Response Error Count**, **Average Execution Time (ms)**, and **Average Response Time (ms)**.

## Managing Oracle BAM Availability

Oracle BAM high availability is based on Oracle WebLogic Server high availability.

### Note:

Make sure that times on all Oracle BAM Server instances in a cluster are synchronized to UTC. Time differences can result in event losses for continuous queries.

Oracle BAM Server instances can be started and stopped using Fusion Middleware Control.

Restarting is required for any configuration changes made in Fusion Middleware Control properties pages for Oracle BAM or its configuration files.

### To start or stop Oracle BAM servers in Fusion Middleware Control:

1. Log in to the Fusion Middleware Control console.
2. In the Target Navigation pane, expand the Business Activity Monitoring node.
3. Right-click the `BamServer` node under the Business Activity Monitoring node.  
For example, the `BamServer` node might be `BamServer (bam_server1)`.
4. Select the **Control** menu item.

**Start Up** starts the server.

**Shut Down** shuts down the server.

 **Note:**

Do not use **Start Up** alone to restart the component. To restart the component, you must stop the component using **Shut Down** before using **Start Up** to start the component or an error occurs.

## Configuring Oracle User Messaging Service

Oracle User Messaging Service (UMS) must be configured properly in Fusion Middleware Control to send email notifications for alerts.

The UMS email driver monitors the outbound email account configured for Oracle BAM Server for any delivery failures, including rejected emails. Email delivery failure notifications are asynchronous: there is no definite time within which email rejection notification is received.

To make UMS work with SSL, remove `javax.net.ssl.trustStore` from the JVM argument in the `setDomainEnv.sh` file for your Oracle WebLogic Server domain.

### To configure UMS:

1. Log in to the Fusion Middleware Control console.
2. Make sure that the BAM Server is running:
  - a. In the Target Navigation pane, expand the Business Activity Monitoring node.
  - b. Right-click the `BamServer` node under the Business Activity Monitoring node.  
For example, the `BamServer` node might be `BamServer (bam_server1)`.
  - c. Select `Control > Start Up` from the menu.  
If the BAM Server is not running, it is started. If it is already running, an `Application already running` error message appears.
3. In the Target Navigation pane, expand the User Messaging Service node.
4. Right-click the **usermessagingdriver-email (bam\_server1)** node.
5. Select the **Email Driver Properties** menu item.  
The Email Driver Properties page appears.
6. Set or verify the following properties. Email addresses and mail server names are example values.
  - Use Default Sender Address — `EMAIL:example@example.com`
  - E-mail Receiving Protocol — `IMAP`
  - Connection Retry Limit — `-1`
  - Message Cleanup Frequency — `600`
  - Auto Delete — `unchecked`
  - Debug Mode — `unchecked`
  - Message Retrieval Frequency — `30`

- Message Folder — INBOX
  - Outgoing Mail Server — `mailserver.example.com`
  - Outgoing Mail Server Port — 465
  - Outgoing Mail Server Security — SSL
  - Default From Address — `example@example.com`
  - Outgoing Username — `example@example.com`
  - Outgoing Password — select **clear text password** from the drop-down]
  - Incoming Mail Server — `mailserver.example.com`
  - Incoming Mail Server Port — 993
  - Enable SSL — checked
  - Incoming Mail IDs — `example@example.com`
  - Incoming User IDs — `example@example.com`
  - Incoming Passwords — select **clear text password** from the drop-down]
  - Message Processing Chunk S — 100
  - Disable Plain-Text Authentic — unchecked
7. Click **Apply** to save the property changes.
  8. In the Target Navigation pane, expand the Business Activity Monitoring node.
  9. Double-click the BamServer node under the Business Activity Monitoring node.  
For example, the BamServer node might be `BamServer (bam_server1)`.  
The BAM Properties page appears, displaying BAM System Settings and Alert Engine properties.
  10. Type the default sender address as the value for the **Sender Email ID** property. For example, type `example@example.com`.
  11. Click **Apply** to save the change.

# Securing Oracle BAM

This chapter describes how to create users, groups, and roles with different levels of access to Oracle Business Activity Monitoring (Oracle BAM). It also describes how to configure security for Oracle BAM in Oracle WebLogic Server.

This chapter includes the following sections:

- [Understanding Oracle BAM Security](#)
- [Understanding Oracle BAM Permissions](#)
- [Managing Oracle BAM Users](#)
- [Configuring BAM Server Security](#)

## Understanding Oracle BAM Security

Oracle BAM security depends on Oracle WebLogic Server security.

Oracle BAM users and groups are created in the Oracle WebLogic Server Administration Console. Roles that correspond to and grant default permissions to BAM groups are defined in Oracle Enterprise Manager Fusion Middleware Control.

## Understanding Oracle BAM Permissions

Each entity in Oracle BAM has permissions that can be assigned to BAM users. All entities that are part of a project inherit the project permissions by default. Data object security includes metadata security and data (row) security.

Permissions for BAM entities are as follows:

- **Read** — Permission to view the entity, or the metadata of a data object.
- **Write** — Permission to edit the entity, or the metadata of a data object. Write permission automatically includes Read permission.
- **Remove** — Permission to delete the entity.
- **Select** — Permission to select, or read, the data in a data object. Applies only to data objects.

This is the minimum permission needed for using a data object in business queries, business views, dashboards, and other Designer entities.

- **Delete** — Permission to delete data object rows. Applies only to data objects.
- **Update** — Permission to update or insert data object rows. Applies only to data objects.
- **Security** — Permission to set these permissions for other users of this entity.

Permissions can be granted or denied. All *granted* permissions are combined using OR logic during runtime authorization. For example, if a user has multiple roles, and if any of these roles is granted access permission to an entity, then the user can access the entity.

However, an Administrator user or the entity owner can explicitly deny access permission to a specific role. All *denied* permissions are combined using AND logic during runtime authorization, and denied permissions override granted permissions. If any role the user has is denied access to an entity, then the user can't access the entity, even if his other roles grant permission to access it.

For more information about permissions for specific BAM entities, see the following sections:

- [Securing a Project](#)
- [Securing a Business Query](#)
- [Securing a KPI](#)
- [Securing a Business View](#)
- [Securing a Dashboard](#)
- [Securing an Alert](#)
- [Securing a Parameter](#)
- [Securing a Data Object](#)
- [Securing an Enterprise Message Source](#)

## Managing Oracle BAM Users

Oracle BAM provides user roles to control visibility and access. In addition to managing predefined user roles, you can also create your own.

This section contains the following topics:

- [Predefined Groups and Roles](#)
- [Adding Users to Groups](#)
- [Creating Custom Groups](#)
- [Creating Custom Roles](#)

## Predefined Groups and Roles

[Table 17-1](#) describes the predefined groups and roles in Oracle BAM. All BAM users must be in the BAMUsers group in addition to any other groups.

**Table 17-1 Oracle BAM Predefined Security Groups and Roles**

Group	Role	Description
BAMContentViewer	BAMContentViewer	Can view dashboards and alert history on the Home page.
BAMContentCreator	BAMContentCreator	Can perform all BAMContentViewer tasks. Can view data objects and create projects, dashboards, alerts, and all components thereof on the Designer page.
BAMArchitect	BAMArchitect	Can create and modify data objects and Enterprise Message Sources on the Administrator page.
BAMAdministrator	BAMAdministrator	Can perform all tasks of other roles. Can create and modify users, groups, roles, and permissions. Can modify the Oracle BAM system configuration.
BAMUsers	(none)	Can log in to Oracle BAM and view the Home page.

By default, users who are only in the BAMUsers group can log in but have no permissions. Also by default, users in the BAMContentViewer group and role have Read permissions.

The DefaultDataAccess property changes some default permissions. If set to `off`, users in the BAMContentViewer and BAMContentCreator groups do not have Read, Select, or Update permissions for data objects by default. If set to `on`, these groups do have these permissions by default. The property default is `on`. To set this property, see [Configuring BAM Server MBean Properties](#).

## Adding Users to Groups

You configure Oracle BAM users and assign them to groups in the Oracle WebLogic Server Administration Console. Assigning users to groups automatically assigns them to the corresponding roles.

For complete details, see Create Users and Add Users to Groups in the *Oracle WebLogic Server Administration Console Online Help*.

### To add users and assign them to groups:

1. Access the Oracle WebLogic Server console using a URL that includes the Oracle WebLogic Server hostname and the console port:

```
http://wls-hostname:console-port/console
```

For example:

```
http://localhost:7011/console
```

2. Log in using the Oracle WebLogic Server admin username and password.
3. In the left pane of the Oracle WebLogic Server Administration Console, select **Security Realms**.
4. On the Summary of Security Realms page, select the name of the realm (for example, **myrealm**).
5. On the Settings for *Realm Name* page, select **Users and Groups > Users**.
6. Click **New**.
7. In the **Name** field of the Create New User page, enter a unique alphanumeric name for the user.
8. In the Description field, enter a description. The description might be the user's full name. This is optional.
9. In the **Provider** drop-down list, select **DefaultAuthenticator**.
10. In the **Password** field, enter a password for the user.

The minimum password length is 8 characters. Do not use the username/password combination `weblogic/welcome1` in production.

11. Re-enter the password for the user in the **Confirm Password** field.
12. Click **OK** to save your changes.
13. Click the name of the new user in the User table.
14. On the Settings for *User Name* page, select **Groups**.
15. Select the BAMUsers group and at least one other BAM group from the **Available** list box and move them to the **Chosen** list box. See [Table 17-1](#) for descriptions of the BAM groups.

16. Click **Save**.

## Creating Custom Groups

You can create custom groups for Oracle BAM in the Oracle WebLogic Server Administration Console. After you create a custom group, you can create a custom role and assign the group to the role.

### To create a new group:

1. Access the Oracle WebLogic Server console using a URL that includes the Oracle WebLogic Server hostname and the console port:  

```
http://wls-hostname:console-port/console
```

For example:

```
http://localhost:7011/console
```
2. Log in using the Oracle WebLogic Server admin username and password.
3. In the left pane of the Oracle WebLogic Server Administration Console, select **Security Realms**.
4. On the Summary of Security Realms page, select the name of the realm (for example, **myrealm**).
5. On the Settings for *Realm Name* page, select **Users and Groups**.
6. Select the **Groups** tab, and click **New**.
7. Enter the group name and description. Do not change the provider.
8. Click **OK**.

## Creating Custom Roles

You can create custom roles for custom Oracle BAM groups in Fusion Middleware Control. For more information, see Managing Application Roles in *Securing Applications with Oracle Platform Security Services*.

## Configuring Oracle BAM Server Security

Security is configured in Oracle WebLogic Server.

See Configuring Security for a WebLogic Domain in *Administering Security for Oracle WebLogic Server*. For more information on Oracle WebLogic Server security, see Overview of the WebLogic Security Service in *Understanding Security for Oracle WebLogic Server*

The following topics address security for Oracle BAM:

- [Configuring Credential Mapping](#)
- [Configuring Secure Socket Layer](#)

## Configuring Credential Mapping

Configure Oracle BAM Adapter to securely store user name and password properties using Oracle WebLogic Server credential mapping functionality.

 **Note:**

Use plain text user names and passwords only in nonproduction mode. Do not mix using credential mapping and plain text user information; apply one at a time to avoid confusion.

To configure credential mapping:

1. Using the Oracle WebLogic Server Administration Console, go to **Deployment > OracleBamAdapter > Security > Credential Mappings**.

 **Note:**

2. Click **New** and select an outbound connection pool instance for which to create a credential mapping entry, then click **Next** (By default, Oracle BAM is installed with two connection pools, **eis/bam/rmi** and **eis/bam/soap**).

 **Note:**

Ensure that a separate credential mapping entry for each outbound connection pool is created. If there are any additional outbound connection pools other than the defaults, you must create corresponding credential mappings for them as well.

3. Select **Unauthenticated WLS User**, and click **Next (Unauthenticated WLS User is similar to an anonymous user)**.
4. Specify a corresponding user name and password for connecting to Oracle BAM Server, then click **Finish** to complete the credential mapping.
5. Repeat these steps to create the credential mapping for the other Oracle BAM connection pool entries.

After you configure the credential mappings for each of the outbound connection pool entries, the mappings appear in the **Credential Mappings** table.

6. For these changes to take effect, Oracle WebLogic Server must be restarted.

## Configuring Secure Socket Layer

Oracle WebLogic Server provides the facilities needed to enable Secure Socket Layer (SSL) on any Oracle SOA Suite and Oracle WebCenter Portal connections into Oracle WebLogic Server.

Use the Java Development Kit (JDK) `keytool` utility to create and manage keystores and certificates, and use the Oracle WebLogic Server Administration Console to configure Oracle WebLogic Server listeners.

For more information, see *Configuring SSL in Administering Security for Oracle WebLogic Server*.

Oracle WebLogic Server lets you configure Oracle BAM to run on both SSL and non-SSL ports. However, when Oracle BAM is running on both SSL and non-SSL ports, it cannot send two Oracle BAM report URLs to the alert email receiver. It is the responsibility of the administrator to decide which mode of access (SSL or non-SSL) to use for the Oracle BAM report URL recipients.

To start Oracle BAM in SSL mode, and enable Oracle BAM clients (like BAMCommand) to invoke EJBs, JMS resources, and JDBC resources using SSL, you must configure the following Oracle BAM properties:

- **Communication\_Protocol in the BAMServerConfig.xml file, as follows:**  
Default value: `<Communication_Protocol>t3</Communication_Protocol>`  
For SSL using t3s: `<Communication_Protocol>t3s</Communication_Protocol>`
- **protocol and port in the BAMCommandConfig.xml file, as follows:**  
Default value: `<protocol>t3</protocol>`  
Default value: `<port>listenport</port>`  
For SSL using t3s: `<protocol>t3s</protocol>`
- **ListenPort in BAMCommonConfig.xml file, as follows:**  
Default value: `<ListenPort>ListenPort</ListenPort>`  
For SSL: `<ListenPort>SSLListenPort</ListenPort>`

# 18

## Using Commands with Oracle BAM

This chapter describes how to use the BAMCommand command-line utility to export, import, migrate, clear, and delete BAM metadata, data, and project artifacts. It also describes how to use the loadgen utility to simulate a data stream.

This chapter includes the following sections:

- [Using BAMCommand](#)
- [Using the Loadgen Utility](#)

### Using BAMCommand

This section describes how to run the BAMCommand command-line utility locally or remotely, and how to export, import, migrate, clear, and delete BAM metadata, data, and project artifacts.

This section includes the following topics:

- [Executing BAMCommand](#)
- [Using the BAMCommand Configuration File](#)
- [Running BAMCommand Remotely](#)
- [Specifying the Command and Parameter Syntax](#)
- [Summary of Individual Operations](#)
- [Detailed Operation Descriptions](#)
- [Command File Structure](#)
- [Log File Structure](#)
- [Export File Structure](#)
- [Migrate Considerations](#)
- [Upgrade Considerations](#)
- [Regular Expressions](#)

### Executing BAMCommand

BAMCommand can be executed from the `FMW_HOME\soa\bam\bin` directory on the Windows platform and from `FMW_HOME/soa/bam/bin` on UNIX platforms.

Entering `bamcommand` on the command line by itself lists the BAMCommand operations and parameters.

Before executing BAMCommand, you must set the `JAVA_HOME` environment variable to point to the root directory of the supported version of the Java Development Kit. See the Oracle BAM support matrix on the Oracle Technology Network web site for supported JDK versions.

## Using the BAMCommand Configuration File

Use the BAMCommand configuration file to preset BAMCommand parameters and not have to type them on the command line each time you use BAMCommand. Command-line parameters override parameters in the configuration file. This file is located at `FMW_HOME/soa/bam/bin/BAMCommandConfig.xml`.

Here is an example BAMCommandConfig.xml file:

```
<host>www.example.com</host>
<port>7001</port>
<username>weblogic</username>
<password>abcdefg1</password>
<dbusername>SOAINFRA</dbusername>
<dbpassword>abcdefg1</dbpassword>
<dburl>jdbc:oracle:thin:@localhost:1521:orcl</dburl>
```

## Running BAMCommand Remotely

You can run BAMCommand from a remote system (where Oracle BAM is installed) and execute the commands on a server located remotely. To run BAMCommand remotely, specify the properties `-host` and `-port` on the command line or in `FMW_HOME/soa/bam/bin/BAMCommandConfig.xml`, as shown below.

```
<host>www.example.com</host>
<port>7001</port>
```

The Oracle BAM version installed on the remote system should be same as the Oracle BAM Server version (that is, both servers should be from the same label).

## Specifying the Command and Parameter Syntax

The basic structure of the BAMCommand command-line entry is as follows:

```
bamcommand [-host host] [-port port] [-username username] [-logfile filename] [-logmode
[overwrite|append]] -cmd operation -param value...
```

All operation-specific parameters on the command line are in the following form:

```
-param value
```

The `param` portion is not case sensitive. If the `value` portion contains spaces or other special characters, it must be enclosed in double quotation marks. For example:

```
bamcommand -cmd export -name "/Samples/Call Center" -file "C:\CallCenter.zip"
```

This section includes the following topics:

- [Specifying General Parameters](#)
- [Specifying Security Credentials](#)
- [Specifying Object Names](#)

## Specifying General Parameters

The following parameters can be used in any BAMCommand commands:

- cmd

`-cmd operation`

Specifies a single operation to be executed. Any parameters needed for the operation must also be on the command line.

The `cmd` and `cmdfile` parameters are mutually exclusive. Exactly one of them must be present.

- cmdfile

`-cmdfile filename`

Specifies the name of an XML file that contains operations to be processed. The XML file extension is optional.

The `cmd` and `cmdfile` parameters are mutually exclusive. Exactly one of them must be present.

The following is an example command file:

```
<OracleBAMCommands continueonerror="1">
  <Export id="1" type="dataobject" name="D01" file="file1"/>
  <Export id="2" type="dataobject" name="D02" file="file2"/>
</OracleBAMCommands>
```

See [Command File Structure](#) for more information.

- host

`-host www.example.com`

Optional parameter that specifies the host name of the Oracle BAM server. The default host name is specified in the default BAMCommand configuration file. See [Running BAMCommand Remotely](#) for more information.

- port

`-port port`

Optional parameter that specifies the port of the Oracle BAM server. The default port is 7001, and is specified in the default BAMCommand configuration file. See [Running BAMCommand Remotely](#) for more information.

If the Oracle BAM Server port number is changed from the default during the setup and configuration of Oracle BAM, then you must manually change the port number from 7001 to the new port number in the BAMCommand configuration file `FMW_HOME/soa/bam/bin/BAMCommandConfig.xml`.

- username

`-username username`

Optional parameter that specifies the username that the command should run as. See [Specifying Security Credentials](#) for more information. BAMCommand prompts for the password unless it is configured in the `FMW_HOME/soa/bam/bin/BAMCommandConfig.xml` file.

- logfile

`-logfile filename`

Optional parameter that specifies the name of the file to which results and errors are logged. If the file does not exist, it is created. If the file does exist, any existing contents are

overwritten or appended to according to the `logmode` setting. If this parameter is not present, results and errors are output to the console.

See [Log File Structure](#) for more information.

- `logmode`  
`-logmode [overwrite|append]`

Optional parameter that indicates whether an existing log file is to be overwritten or appended to. If this parameter is not present, `overwrite` is the default.

## Specifying Security Credentials

BAMCommand requires users to provide security credentials when running operations. If no security credentials have been specified in the configuration file, BAMCommand securely prompts for a user name and password.

To use default credentials, add the `username` and `password` properties to the `FMW_HOME/soa/bam/bin/BAMCommandConfig.xml` file. For example:

```
<username>weblogic</username>  
<password>abcdefg1</password>
```

However, the `-username` command-line parameter always overrides the `username` property specified in the configuration file.

The user name and password for running BAMCommand operations can come from the configuration file or command-line prompts. You can also specify the user name as a command-line parameter. The options are as follows:

- If the user name and password are only specified in the configuration file (that is, `-username` parameter is not used in the command line), then the `username` and `password` values in the configuration file are used.
- If only the user name is specified in the configuration file and the password is not, then the user name value is used, and BAMCommand prompts the user for the password at the command line.
- If user name is specified on the command line, then that value is used, and BAMCommand prompts the user for a password. The password prompt occurs regardless of any properties specified in the configuration file. For example:

```
bamcommand -cmd export -name TestDO -file C:\TestDO.zip -username username
```

## Specifying Object Names

Whenever an object name is specified in a command, the following rules apply.

When specified on a command line, if the name contains spaces or characters that have special meaning to DOS or UNIX, the name must be quoted according to the rules for command lines.

Project artifacts must be prefixed by the project name. For example:

```
-name MyProject/MyAlert
```

The full internal name of data objects must be specified.

## Summary of Individual Operations

This section summarizes the parameters for each BAMCommand operation. You can display a summary of these operations in the command window by entering `bamcommand` (without any parameters) at the command prompt.

[Table 18-1](#) summarizes the operations available in BAMCommand.

**Table 18-1** BAMCommand Operation Summary

Operation	Parameters
clear	<code>-name <i>itemname</i></code> <code>[-type [<i>dataobject</i>]]</code> For more information about clear see <a href="#">Clear</a> .
delete	<code>[-all [0 1]]</code> <code>[-match <i>pattern</i>]</code> <code>-name "<i>projectname/itemname</i>"</code> <code>[-regex <i>regularexpression</i>]</code> <code>[-type [<i>dataobject project view businessquery dashboard alert kpi parameter ems all</i>]]</code> For more information about delete see <a href="#">Delete</a> .
export	<code>[-all [0 1]]</code> <code>[-append [0 1]]</code> <code>[-contents [0 1]]</code> <code>[-dependencies [0 1]]</code> <code>-file <i>filename</i></code> <code>[-layout [0 1]]</code> <code>[-match <i>pattern</i>]</code> <code>-name "<i>projectname/itemname</i>"</code> <code>[-owner [0 1]]</code> <code>[-regex <i>regularexpression</i>]</code> <code>[-type [<i>dataobject project view businessquery dashboard alert kpi parameter ems all</i>]]</code> For more information about export see <a href="#">Export</a> .

**Table 18-1 (Cont.) BAMCommand Operation Summary**

Operation	Parameters
import	<pre>[-batchsize <i>integer</i>] [-contents [0 1]] [-continueonerror [0 1]] [-datamode [error append overwrite update]]] [-datatruncate [0 1]] [-dbusername <i>dbusername</i>] [-dburl <i>dburl</i>] [-dbdriver <i>dbdriver</i>] -file <i>filename</i> [-migrate [0 1]] [-mode [error append overwrite update]]] -name "<i>projectname/itemname</i>" [-preserveowner [0 1]] -setcol "<i>columnname/overridevalue</i>" [-type [dataobject project view businessquery dashboard alert  kpi parameter ems all]] [-upgrade [0 1]] For more information about import see <a href="#">Import</a>.</pre>
rename	<pre>-name "<i>projectname/itemname</i>" -newdisplayname "<i>projectname/itemname</i>" [-type [dataobject project view businessquery dashboard alert  kpi parameter ems]] For more information about rename see <a href="#">Rename</a>.</pre>
start	<code>bamcommand -cmd start -name &lt;ems_name&gt; -type ems</code>
stop	<code>bamcommand -cmd stop -name &lt;ems_name&gt; -type ems</code>

## Detailed Operation Descriptions

This section details each of the BAMCommand operations, including their parameters, and gives examples. It includes the following topics:

- [Clear](#)
- [Delete](#)
- [Export](#)
- [Import](#)
- [Rename](#)

### Clear

Clears the contents of a data object from the Oracle BAM server.

**Table 18-2 Clear Operation Parameters**

Parameter	Description
<code>-name itemname</code>	The name of the item to be cleared. Required.
<code>-type itemtype</code>	The type of the item to be cleared. <code>dataobject</code> is the only allowed value and the default if this parameter is omitted. This is outlined through an example as follows.

Here is an example of how you can clear a data object:

```
bamcommand -cmd clear -name "Call Center" -type dataobject
```

## Delete

Deletes an item from the Oracle BAM server.

**Table 18-3 Delete Operation Parameters**

Parameter	Description
<code>-all [0 1]</code>	Controls whether all items of the specified type are deleted (This is illustrated in an example command statement after this table). A nonzero or omitted value means delete all items of the specified type, a zero (0) value means only delete the named (or matched) items. Zero is the default if this parameter is omitted.
<code>-match pattern</code>	A DOS-style pattern matching string, using the asterisk (*) and question mark (?) characters. The items whose names match the pattern are deleted. This is illustrated in an example command statement after this table.
<code>-name "projectname/itemname"</code>	The name of the item to be deleted. For project artifacts, the name of the project must be specified as a prefix.
<code>-regex regexexpr</code>	A regular expression pattern matching string. The items whose names match the pattern are deleted. See <a href="#">Regular Expressions</a> for more information.
<code>-type itemtype</code>	The type of the item to be deleted. The following are valid: <ul style="list-style-type: none"> <li><code>dataobject</code> (This is illustrated in an example command statement after this table)</li> <li><code>project</code></li> <li><code>view</code></li> <li><code>businessquery</code></li> <li><code>dashboard</code></li> <li><code>alert</code></li> <li><code>kpi</code></li> <li><code>parameter</code></li> <li><code>ems</code> (Enterprise Message Source) (This is illustrated in an example command statement after this table.)</li> <li><code>all</code></li> </ul> <code>dataobject</code> is the default if this parameter is omitted.

Here is an example of how you can delete a data object. This command deletes a data object named `TestDO`. Note that the `dataobject` type is assumed if the `-type` parameter is not specified.

```
bamcommand -cmd delete -name TestDO
```

The following command deletes all objects of type EMS:

```
bamcommand -cmd delete -type ems -all 1
```

You can also delete items that have a common prefix at one go. For example, this command deletes all items that have names beginning with ACME.

```
bamcommand -cmd delete -match "ACME*" -type all
```

## Export

Exports information about one or more objects in the Oracle BAM server to a ZIP file. See [Export File Structure](#) for an example of an export file.

**Table 18-4 Export Operation Parameters**

Parameter	Description
-all [0 1]	Controls whether all items of the specified type are exported. A nonzero or omitted value means export all items of the specified type, a zero value means only export the named (or matched) items. Zero (0) is assumed if this parameter is omitted. For projects and project artifacts, only the items owned by the user running BAMCommand are exported, unless the user running BAMCommand is an administrator. When an administrator runs BAMCommand, any user's items may be exported.
-append [0 1]	Controls whether exported information is appended to existing ZIP files. A nonzero or omitted value means append items of the specified type, a zero value means overwrite them. Zero (0) is assumed if this parameter is omitted.
-contents [0 1]	Controls whether content information (row, column values) is to be exported. Applies only to data objects. Cannot be used with the <code>-type all</code> parameter because the <code>contents</code> parameter is not available for other artifacts. A nonzero value means export the content information. Zero (0) means do not export the content information. A nonzero value is the default if this parameter is omitted. In addition, can only be used when: <ul style="list-style-type: none"> <li>• All the data objects are exported using the parameter <code>-all 1</code>.</li> <li>• A set of data objects is exported using the <code>regex</code> or <code>match</code> parameter.</li> <li>• A single data object is exported.</li> </ul>
-dependencies [0 1]	Controls whether dependent items are exported with the specified items. A nonzero or omitted value means include dependent items, a zero value means ignore them. Zero (0) is assumed if this parameter is omitted.
-file <i>filename</i>	The name of the ZIP file to export to. Required. If the file does not exist, it is created. If the file does exist, any contents are overwritten. The default extension is <code>.zip</code> .
-layout [0 1]	Controls whether data object layout (metadata) is exported. A nonzero value means export the layout, a zero value means do not. Zero (0) is assumed if this parameter is omitted.
-match <i>pattern</i>	A DOS-style pattern matching string, using the asterisk (*) and question mark (?) characters. The items whose names match the pattern are exported.
-name " <i>projectname/itemname</i> "	The internal name of the item to be exported. For project artifacts, the name of the project must be specified as a prefix.

**Table 18-4 (Cont.) Export Operation Parameters**

Parameter	Description
<code>-owner [0 1]</code>	Controls whether ownership information is exported. A nonzero value, such as <code>-owner nonzero</code> or <code>-owner</code> means export the ownership information, a zero value means do not. Zero (0) is assumed if this parameter is omitted, for example, <code>-owner 0</code> or without the <code>-owner</code> option including total name <code>-owner</code> and its value such as <code>bamcommand -cmd import -name DO</code> .
<code>-regex regexexpr</code>	A regular expression pattern matching string. The items whose names match the pattern are exported. See <a href="#">Regular Expressions</a> for more information.
<code>-type itemtype</code>	The type of the item to be exported. The following are valid: <ul style="list-style-type: none"> <li>• dataobject</li> <li>• project</li> <li>• view</li> <li>• businessquery</li> <li>• dashboard</li> <li>• alert</li> <li>• kpi</li> <li>• parameter</li> <li>• ems (Enterprise Message Source)</li> <li>• all</li> </ul> dataobject is the default if this parameter is omitted.

You can export a data object as follows:

```
bamcommand -cmd export -name "CallCenter" -file "C:\CallCenter.zip"
```

Note that the `-type` parameter was not included in this example. By default `dataobject` is the `-type` if it is not specified.

To export data object contents, use the following command format:

```
bamcommand -cmd export -name TestDataObject -contents 1 -file "C:/TestDataObjectData.zip"
```

Here is an example of how you can export data objects using the 'Match' parameter:

```
bamcommand -cmd export -match "M*" -file "C:/DostartingwithM.zip"
```

Here's an example of how you can export a project:

```
bamcommand -cmd export -name MyProject -type project -file "C:\MyProject.zip"
```

Here is an example of how you can export a business view:

```
bamcommand -cmd export -name "MyProject/BarChart" -type view -file "C:\BarChart.zip"
```

Note that the project name and a slash (/) precede the view name.

Here is an example of how you can export a business query:

```
bamcommand -cmd export -name "MyProject/CallsPerHour" -type businessquery -file "C:\CallsPerHour.zip"
```

Note that the project name and a slash (/) precede the query name.

Here is an example of how you can export a dashboard:

```
bamcommand -cmd export -name "MyProject/CallCenterDash" -type dashboard -file
"C:\CallCenterDash.zip"
```

Note that the project name and a slash (/) precede the dashboard name.

Here is an example of how you can export an alert:

```
bamcommand -cmd export -name "MyProject/Alert1" -type alert -file "C:\Alert1.zip"
```

Note that the project name and a slash (/) precede the alert name.

Here is an example of how you can export an Enterprise Manager source:

```
bamcommand -cmd export -type ems -name TestEMS -file "C:\TestEMS.zip"
```

Here is how you can export all dashboards in a project:

```
bamcommand -cmd export -all 1 -name MyProject -type dashboard -file "C:\AllDash.zip"
```

Note that the project name must be specified.

Here is how you can export all Oracle BAM objects in the system:

```
bamcommand -cmd export -type all -file "C:\temp\TestAll.zip"
```

## Import

Imports information from a previously exported ZIP file or a previous version of BAM to objects in the Oracle BAM server. The objects may be created, replaced, or updated. If an object does not exist, it is created if possible.

For data objects, the input file must contain the layout information necessary to create the data object. If the file contains no content information, an empty data object is created.

**Table 18-5 Import Operation Parameters**

Parameter	Description
<code>-batchsize</code> <i>integer</i>	Applies only if <code>-migrate</code> is used. Specifies the JDBC batch update size for data migration. The default is 10000.
<code>-contents</code> [0 1]	<p>Controls whether content information (row, column values) is to be imported. Applies only to data objects.</p> <p>A nonzero value means import the content information. Zero (0) means do not import the content information. A nonzero value is the default if this parameter is omitted.</p> <p>Cannot be used with the <code>-type all</code> parameter because the <code>contents</code> parameter is not available for other artifacts.</p> <p>Cannot be set to zero when used with the <code>migrate</code> parameter.</p> <p>In addition, can only be used when:</p> <ul style="list-style-type: none"> <li>• All the data objects are imported using the parameter <code>-all 1</code>.</li> <li>• A set of data objects is imported using the <code>regex</code> or <code>match</code> parameter.</li> <li>• A single data object is imported.</li> </ul>

Table 18-5 (Cont.) Import Operation Parameters

Parameter	Description
- continueonerror [0 1]	<p>Applies only if <code>-migrate</code> is used. Determines whether importing of data records continues if an error occurs.</p> <p>If an error occurs during migration of one data object, other data objects in the same <code>migrate</code> operation are not affected.</p> <p>A nonzero value means true (continue). Zero (0) means false (do not continue). Zero is the default if this parameter is omitted.</p> <p>If this parameter is true, rejected data records are written to a BAD file, which can be corrected and then imported in a separate operation. The naming convention for the BAD file is:</p> <pre>dataobjectname_BAD_timestamp.xml</pre>
-datamode [error append  overwrite  update]]	<p>Specifies how imported data interacts with existing data. The <code>BEAM_ID</code> column determines matches.</p> <ul style="list-style-type: none"> <li>• <code>error</code> (default) skips matches between existing and imported data records.</li> <li>• <code>append</code> inserts imported records without affecting existing data.</li> <li>• <code>overwrite</code> deletes all existing data before importing.</li> <li>• <code>update</code> appends records that do not already exist and updates records that do exist. If <code>BEAM_ID</code> is missing, imported data is appended.</li> </ul> <p>To specify how existing items are handled, use the <code>mode</code> parameter. To import the 11g data object schema, use the <code>upgrade</code> parameter.</p>
-datatruncate [0 1]	<p>Applies only if <code>-migrate</code> is used. Determines whether data values exceeding 2000 characters are truncated to 2000 characters.</p> <p>A nonzero value means true (truncate). Zero (0) means false (do not truncate). One (1) is the default if this parameter is omitted.</p> <p>If this parameter is false (0) and a data value exceeding 2000 characters is encountered, the <code>-continueonerror</code> parameter determines the action taken.</p>
-dbusername <i>dbusername</i> -dburl <i>dburl</i>	<p>These parameters apply only if <code>-migrate</code> is used. They specify the connection to the Oracle BAM 14c database. All are required.</p> <p>For convenience, you can specify these parameters, plus the <code>dbpassword</code>, in the <code>BAMCommandConfig.xml</code> file in addition to the parameters for the 14c database. See <a href="#">Using the BAMCommand Configuration File</a> for more information.</p>
-file <i>filename</i>	<p>The name of the file to import from. Required. This is usually a file created using the <code>export</code> operation. It can also be an XML or CSV file for a data object or enterprise message source from Oracle BAM 11g.</p>
-migrate [0 1]	<p>Applies only to data objects. Imports Oracle BAM 11g data into Oracle BAM 14c. A nonzero value means true (migrate). Zero (0) means false (do not migrate). Zero is the default if this parameter is omitted.</p> <p>When <code>-migrate</code> is used, the <code>dbusername</code> and <code>dburl</code> parameters are required and the <code>batchsize</code> parameter is optional.</p> <p>To import the 11g data object schema, use the <code>upgrade</code> parameter. See <a href="#">Migrate Considerations</a> for more information.</p>

**Table 18-5 (Cont.) Import Operation Parameters**

Parameter	Description
-mode [error append overwrite update]]	<p>Specifies how imported items interact with existing items having the same names:</p> <ul style="list-style-type: none"> <li>• <code>error</code> (default) disallows import if dependent objects of an artifact are already present in system. For example, if an existing project contains data objects, a project of the same name is not imported.</li> <li>• <code>append</code> allows import if the existing item has dependent items and appends to the existing item definition.</li> <li>• <code>overwrite</code> removes the existing item having the same name.</li> <li>• <code>update</code> allows import if the existing item has dependent items and updates the existing item definition.</li> </ul> <p>To specify how existing data is handled, use the <code>datamode</code> parameter. To import the 11g data object schema, use the <code>upgrade</code> parameter.</p>
-name "projectname/itemname"	The name of the item to be imported. For project artifacts, the name of the project must be specified as a prefix.
-preserveowner [0 1]	<p>Controls whether ownership information is imported.</p> <p>A nonzero or omitted value means import the ownership information. A zero value means set the ownership to the user running the import. Zero (0) is assumed if this parameter is omitted.</p>
-setcol "columnname/overridevalue"	Applies only to data objects. Allows override of column values from the command line during import. The override value can be <code>NULL</code> for no value, <code>NOW</code> for the current datetime, or another value.
-type itemtype	<p>The type of the item to be imported. The following are valid:</p> <ul style="list-style-type: none"> <li>• <code>dataobject</code></li> <li>• <code>project</code></li> <li>• <code>view</code></li> <li>• <code>businessquery</code></li> <li>• <code>dashboard</code></li> <li>• <code>alert</code></li> <li>• <code>kpi</code></li> <li>• <code>parameter</code></li> <li>• <code>ems</code> (Enterprise Message Source)</li> <li>• <code>all</code></li> </ul> <p><code>all</code> is the default if this parameter is omitted.</p>
-upgrade [0 1]	<p>Applies only to data objects and EMS objects. Imports Oracle BAM 11g artifacts into Oracle BAM 14c.</p> <p>A nonzero value means true (upgrade). Zero (0) means false (do not upgrade). Zero is the default if this parameter is omitted.</p> <p>To import 11g data object data, use the <code>migrate</code> parameter.</p> <p>See <a href="#">Upgrade Considerations</a> for more information.</p>

Here is how you can import a data object and override the schema:

```
bamcommand -cmd import -name TestDataObject -mode update -type dataobject -file
"C:\TestDataObject.zip"
```

Here is how you can import an 11g data object schema.

```
bamcommand -cmd import -nameOldDataObject -upgrade 1 -type dataobject -file
"C:\OldDataObject.xml"
```

Here is how you can import 11g data:

```
bamcommand -cmd import -nameOldDataObject -migrate 1 -type dataobject
-file "C:\OldDataObject.xml" -continueonerror 1
-mode update -dbusername SOAINFRA -dburl jdbc:oracle:thin:@localhost:1511:orcl
```

Note that the 11g database parameters are required, but the `-continueonerror` and `-mode` parameters are optional.

Here is how you can import a query from a project:

```
bamcommand -cmd import -name "MyProject/CallsPerHour" -type businessquery -file
"C:\CallsPerHour.zip"
```

Note that the project name and a slash (/) precede the query name.

Here is how you can import all objects in the file:

```
bamcommand -cmd import -file "C:\temp\TestAll.zip"
```

Note that `-type all` is the default and not needed.

Here is how you can import an Enterprise Message Source:

```
bamcommand -cmd import -file "C:\temp\TestEMS.zip" -type ems
```



#### Note:

During an EMS import using BAMCommand, regardless of the state of EMS (from when it was previously exported), the EMS will be in the 'STOP' state in the target machine.

## Rename

Changes the display name of an item from the Oracle BAM server. The internal name cannot be changed.

**Table 18-6 Rename Operation Parameters**

Parameter	Description
<code>-name</code> " <i>projectname/</i> <i>itemname</i> "	The internal name of the item to be renamed. For project artifacts, the name of the project must be specified as a prefix.
<code>-newdisplayname</code> " <i>projectname/</i> <i>itemname</i> "	The new display name of the item to be renamed. For project artifacts, the name of the project must be specified as a prefix.  The display name is case insensitive for data objects and case sensitive for all other types. It may contain any characters except the forward slash (/), which indicates a folder path. It may have up to 128 characters.

**Table 18-6 (Cont.) Rename Operation Parameters**

Parameter	Description
<code>-type <i>itemtype</i></code>	<p>The type of the item to be renamed. The following are valid:</p> <ul style="list-style-type: none"> <li>• <code>dataobject</code></li> <li>• <code>project</code></li> <li>• <code>view</code></li> <li>• <code>businessquery</code></li> <li>• <code>dashboard</code></li> <li>• <code>alert</code></li> <li>• <code>kpi</code></li> <li>• <code>parameter</code></li> <li>• <code>ems (Enterprise Message Source)</code></li> </ul> <p><code>dataobject</code> is the default if this parameter is omitted.</p>

To rename a data object, use the following command. This command gives the display name `MyDO` to a data object with the internal name of `TestDO`. Note that the `dataobject` type is assumed if the `-type` parameter is not specified.

```
bamcommand -cmd rename -name TestDO -newdisplayname MyDO
```

To rename a View, use the following command. This command gives the display name `MyView` to a business view with the internal name of `BarChart1`.

```
bamcommand -cmd rename -type view -name BarChart1 -newdisplayname MyView
```

## Command File Structure

This section contains the following topics:

- [Operation IDs](#)
- [Continue On Error](#)

The command file contains the root tag `OracleBAMCommands`.

Within the root tag is a tag for every operation to be executed. The tag name is the operation name, and the parameters for the operation are attributes.

Here is a sample command file:

```
<?xml version="1.0" encoding="utf-8"?>
<OracleBAMCommands continueonerror="1">
  <Export name="Samples/Media Sales" file="MediaSales.xml" contents="0" />
  <Rename name="Samples/Call Center" newdisplayname="Call Centre" />
  <Delete type="EMS" name="WebLog" />
  <Delete type="EMS" name="WebLog2" />
</OracleBAMCommands>
```

The output of this sample command file is shown in [Log File Structure](#).

## Operation IDs

This feature is only used when output is being sent to a log file. To make the parsing of log results easier, each operation can be given an ID. This ID is included in the Result or Error elements of any output related to that operation.

Here is a sample input:

```
<OracleBAMCommands continueonerror="1">
  <Delete id="1" type="dataobject" name="Data Object A"/>
  <Delete id="2" type="dataobject" name="Data Object B"/>
</OracleBAMCommands>
```

Here's what a sample output file would look like:

```
<?xml version="1.0"?>
<BAMCommandLog Login="weblogic">
  <Results Command="Delete" ID="1">Data Object &quot;/Data Object A&quot;
  deleted.</Results>
  <Error Command="Delete" ID="2">
    <![CDATA[BAM-02409: There is no Data Object named "Data Object B".
    [ErrorSource="BAMCommandEngine",ErrorID="BAMCommandEngine.DOEExist"]]]>
  </Error>
</BAMCommandLog>
```

## Continue On Error

Ordinarily, BAMCommand executes operations in a command file until a failure occurs, or until they all complete successfully. In other words, if a command file contains 20 operations, and the second operation fails for any reason, then no further operations are executed. This behavior can be changed by using the `continueonerror` attribute at either a global level or for each operation.

You can use the `continueonerror` attribute so that all operations are executed regardless of any failures. Here's how you can enable the Global `continueonerror` mode:

```
<OracleBAMCommands continueonerror="1">
  <Delete id="1" type="dataobject" name="Data Object A"/>
  <Delete id="2" type="dataobject" name="Data Object B"/>
</OracleBAMCommands>
```

In some cases, `continueonerror` only applies to the operation that deletes Data Object A. If this operation fails, then BAMCommand outputs the error and continues. But if any other operation fails, BAMCommand stops immediately. Here is how you can enable the operation-level `continueonerror` mode:

```
<OracleBAMCommands>
  <Delete id="1" type="dataobject" name="Data Object A" continueonerror="1"/>
  <Delete id="2" type="dataobject" name="Data Object B"/>
  <Delete id="3" type="dataobject" name="Data Object C"/>
  <Delete id="4" type="dataobject" name="Data Object D"/>
</OracleBAMCommands>
```

## Log File Structure

The log file contains the root tag `BAMCommandLog`.

Within the root tag is an entry for every error or informational message logged.

Errors are logged with the tag `Error`.

Informational messages are logged with the tag `Results`.

Both `Results` and `Error` tags optionally contain an attribute of the form `Command=cmdname`, if appropriate, that contains the name of the operation that generated the error or informational message.

The sample log file shown here is the output of the command file given in [Command File Structure](#):

```
<?xml version="1.0" encoding="utf-8"?>
<BAMCommandLog Login="user_name">
  <Results Command="Export">Data Object "/Samples/Media Sales" exported
  successfully (0 rows).</Results>
  <Results Command="Export">1 items exported successfully.</Results>
  <Results Command="Rename">Data Object "/Samples/Call Center" renamed to
  "/Samples/Call Centre".</Results>
  <Results Command="Delete">Enterprise Message Source "WebLog" deleted.</Results>
  <Error Command="Delete"><![CDATA[Error while processing command "Delete".
  [ErrorSource="BAMCommand", ErrorID="BAMCommand.Error"] There is no Enterprise Message
  Source named "WebLog2". [ErrorSource="BAMCommand",
  ErrorID="BAMCommand.EMSExist"]]]></Error>
</BAMCommandLog>
```

## Export File Structure

Here is an example of the structure of an exported ZIP file:

```
bam_export.zip:
-dataobject
  -do1
    -dataobject.xml
    -data.csv
-project
  -Project1
    -alert
      -alert1
    -view
      -view1
    -businessquery
      -query1
      -continuousquery
      -cquery1
  -project2
    -parameter
      -parameter2
    -view
      -view2
    -businessquery
      -query2
    -dashboard
      -dashboard1
  ...
-ems
  -ems1
```

Here is a sample XML schema file resulting from exporting a data object:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<ns0:simpleDataObject xmlns:ns0="http://xmlns.oracle.com/bam/bam12">
  <name>SPOTDO</name>
  <id>/dataobject/SPOTDO</id>
  <displayName>SPOTDO</displayName>
  <createdDate>2013-03-14T00:13:49.428</createdDate>
  <modifiedDate>2013-03-14T00:13:49.428</modifiedDate>
```

```
<flexTableConfigured>>false</flexTableConfigured>
<calculatedFields/>
<timeHierarchies/>
<cqType>RELATION</cqType>
<cqArchived>>true</cqArchived>
<cqSource>jms:topic/oracle.beam.server.event.dataobject</cqSource>
<cqStampType>SYSTEM</cqStampType>
<slowChangingDimension>>false</slowChangingDimension>
<physicalName>BEAM_VIEW_26</physicalName>
<eventId>BEAM_ID</eventId>
<replayAmount>0</replayAmount>
<hidden>>false</hidden>
<columns>
  <column>
    <name>BEAM_ID</name>
    <id>0aa861629bd445e6929c762bbd4b38ea</id>
    <displayName>BEAM_ID</displayName>
    <length>10</length>
    <nullable>>true</nullable>
    <unique>>false</unique>
    <precision>0</precision>
    <scale>0</scale>
    <dataType>INT</dataType>
    <columnType>ATTRIBUTE</columnType>
    <hidden>>false</hidden>
  </column>
  <column>
    <name>Sales</name>
    <id>c6e271cfef464fe6a7dd250b9ad62b3d</id>
    <length>10</length>
    <nullable>>true</nullable>
    <unique>>false</unique>
    <precision>0</precision>
    <scale>0</scale>
    <dataType>DECIMAL</dataType>
    <columnType>MEASURE</columnType>
    <hidden>>false</hidden>
  </column>
  <column>
    <name>Cost</name>
    <id>42df8b2bd35043cd82b12dd289279e81</id>
    <length>10</length>
    <nullable>>true</nullable>
    <unique>>false</unique>
    <precision>0</precision>
    <scale>0</scale>
    <dataType>DECIMAL</dataType>
    <columnType>MEASURE</columnType>
    <hidden>>false</hidden>
  </column>
  <column>
    <name>Category</name>
    <id>22d80831e48743e897814a23e3d7d549</id>
    <length>10</length>
    <nullable>>true</nullable>
    <unique>>false</unique>
    <precision>0</precision>
    <scale>0</scale>
    <dataType>VARCHAR</dataType>
    <columnType>DIMENSION</columnType>
    <hidden>>false</hidden>
  </column>
</columns>
```

```

<column>
  <name>Product</name>
  <id>75694c6184cc44199bcec91ee9d30736</id>
  <length>10</length>
  <nullable>true</nullable>
  <unique>false</unique>
  <precision>0</precision>
  <scale>0</scale>
  <dataType>VARCHAR</dataType>
  <columnType>DIMENSION</columnType>
  <hidden>false</hidden>
</column>
</columns>
<indexes/>
<hierarchies>
  <hierarchy>
    <name>Category</name>
    <id>523992a9e8634e1498e0987cd8adb00b</id>
    <columns/>
  </hierarchy>
  <hierarchy>
    <name>PYPDrill</name>
    <id>8dfe2fce91a740a0a4933b25386718b5</id>
    <columns>
      <columnRef>
        <dataObject>SPOTDO</dataObject>
        <column>Product</column>
      </columnRef>
    </columns>
  </hierarchy>
</hierarchies>
</ns0:simpleDataObject>

```

## Migrate Considerations

When you use the `import` operation with the `-migrate` parameter to bring Oracle BAM 12c data into Oracle BAM 14c, some information is modified.

The following considerations apply to data object migration:

- You can import from an 12c data object XML or CSV file.
- If an `auto-incr-integer` column is present in the 12c data object XML file, its values populate the `BEAM_ID` column in the 14c data object.
- If a `Timestamp` column is present in the 12c data object XML file, its values populate the `DATAOBJECT_CREATED` column in the 14c data object.
- If one of the batch inserts fails, and no further processing is done, then the data object state (data object name, batch size, batch number) is written to the `dostate-recovery.xml` file.

After you take corrective action and re-run the same command, the processing starts from the point where it failed.

After successful insertion of records, the corresponding sequence number (for the `FLEX` table) gets updated in the `BEAM_SEQUENCE` table.

- Data values exceeding 2000 characters are truncated to 2000 characters unless the `-datatruncate` parameter is set to zero, in which case the `-continueonerror` parameter determines the action taken.

## Upgrade Considerations

When you use the `import` operation with the `-upgrade` parameter to bring Oracle BAM 12c artifacts into Oracle BAM 14c, some information is modified.

The following considerations apply to data object upgrades:

- The 12c data object ID is converted to a 14c data object internal name. All special characters and embedded blanks are replaced with underscores and all leading underscores are removed.
- The 12c data object path and name are combined to form the 14c data object display name.
- The 12c column IDs are converted to 14c column internal names. All special characters and embedded blanks are replaced with underscores and all leading underscores are removed.
- If an `auto-incr-integer` column is present in the 12c data object XML file, its values populate the `BEAM_ID` column in the 14c data object.
- If a `Timestamp` column is present in the 12c data object XML file, its values populate the `DATAOBJECT_CREATED` column in the 14c data object.
- An 12c fact data object is converted into two 14c data objects:
  - A 14c simple data object consisting of the 12c non-lookup columns
  - A 14c logical data object consisting of the 12c lookup columnsThe lookup columns must be present in the 12c fact data object XML file or in the 14c schema.
- An 12c `String` column is converted to a 14c `Varchar` column with a width of no more than 2000 characters.

The following considerations apply to external data object upgrades:

- BAMCommand does not upgrade the external data source referenced by an 12c external data object. You must recreate the external data source before upgrading the 12c external data object, which is mapped to a 14c external data object.
- An 12c external data object referenced by a standard BAM 12c fact data object is mapped to a 14c simple data object, not to a 14c external data object. You must use a separate operation to migrate the data.

The following considerations apply to EMS upgrades:

- The 12c EMS Connection Factory details are not retained. The Outbound Connection JNDI is mapped to None. You must create these details manually and choose the 14c JNDI name.
- The 12c EMS field **Start when BAM Server starts** is not converted, and the **Auto-start** field is set to false in 14c. You must start the EMS manually.
- The 12c EMS definition XML file lacks the data object column data types. Therefore, you must specify data object and EMS definitions in the same XML file if the base data object is not present in the 14c database.

## Regular Expressions

The `export` and `delete` operations optionally accept a regular expression with the `-regex` parameter.

A regular expression is a pattern of text that consists of ordinary characters (for example, letters `a` through `z`) and special characters, known as *metacharacters*. The pattern describes one or more strings to match when searching for items by name.

### Note:

The behavior of BAMCommand `-regex` parameter is exactly like the `java.util.regex` package for matching character sequences against patterns specified by regular expressions.

Table 18-7 contains the complete list of metacharacters and their behavior in the context of regular expressions.

**Table 18-7 Metacharacters for Regular Expressions**

Character	Description
<code>\</code>	Marks the next character as a special character, a literal, a backreference, or an octal escape. For example, <code>'n'</code> matches the character "n". <code>'\n'</code> matches a newline character. The sequence <code>'\\'</code> matches <code>'\'</code> and <code>'\('</code> matches <code>'('</code> .
<code>^</code>	Matches the position at the beginning of the input string. If the <code>RegExp</code> object's <code>Multiline</code> property is set, <code>^</code> also matches the position following <code>'\n'</code> or <code>'\r'</code> .
<code>\$</code>	Matches the position at the end of the input string. If the <code>RegExp</code> object's <code>Multiline</code> property is set, <code>\$</code> also matches the position preceding <code>'\n'</code> or <code>'\r'</code> .
<code>*</code>	Matches the preceding character or subexpression zero or more times. For example, <code>zo*</code> matches <code>"z"</code> and <code>"zoo"</code> . <code>*</code> is equivalent to <code>{0,}</code> .
<code>+</code>	Matches the preceding character or subexpression one or more times. For example, <code>'zo+'</code> matches <code>"zo"</code> and <code>"zoo"</code> , but not <code>"z"</code> . <code>+</code> is equivalent to <code>{1,}</code> .
<code>?</code>	Matches the preceding character or subexpression zero or one time. For example, <code>"do(es)?"</code> matches the <code>"do"</code> in <code>"do"</code> or <code>"does"</code> . <code>?</code> is equivalent to <code>{0,1}</code> .
<code>{n}</code>	<code>n</code> is a nonnegative integer. Matches exactly <code>n</code> times. For example, <code>'o{2}'</code> does not match the <code>'o'</code> in <code>"Bob,"</code> but matches the two <code>o</code> 's in <code>"food"</code> .
<code>{n,}</code>	<code>n</code> is a nonnegative integer. Matches at least <code>n</code> times. For example, <code>'o{2,}'</code> does not match the <code>"o"</code> in <code>"Bob"</code> and matches all the <code>o</code> 's in <code>"foooooo"</code> . <code>'o{1,}'</code> is equivalent to <code>'o+'</code> . <code>'o{0,}'</code> is equivalent to <code>'o*'</code> .
<code>{n,m}</code>	<code>M</code> and <code>n</code> are nonnegative integers, where <code>n &lt;= m</code> . Matches at least <code>n</code> and at most <code>m</code> times. For example, <code>"o{1,3}"</code> matches the first three <code>o</code> 's in <code>"foooooo"</code> . <code>'o{0,1}'</code> is equivalent to <code>'o?'</code> . Note that you cannot put a space between the comma and the numbers.
<code>?</code>	When this character immediately follows any of the other quantifiers ( <code>*</code> , <code>+</code> , <code>?</code> , <code>{n}</code> , <code>{n,}</code> , <code>{n,m}</code> ), the matching pattern is non-greedy. A non-greedy pattern matches as little of the searched string as possible, whereas the default greedy pattern matches as much of the searched string as possible. For example, in the string <code>"oooo"</code> , <code>'o+?'</code> matches a single <code>"o"</code> , while <code>'o+'</code> matches all <code>'o</code> 's.

**Table 18-7 (Cont.) Metacharacters for Regular Expressions**

Character	Description
.	Matches any single character except "\n". To match any character including the '\n', use a pattern such as '[\s\S]'.
( <i>pattern</i> )	A subexpression that matches <i>pattern</i> and captures the match. The captured match can be retrieved from the resulting Matches collection using the \$0...\$9 properties. To match parentheses characters ( ), use '\(' or '\)'
(?: <i>pattern</i> )	A subexpression that matches <i>pattern</i> but does not capture the match, that is, it is a non-capturing match that is not stored for possible later use. This is useful for combining parts of a pattern with the "or" character ( ). For example, 'industr(?:y ies)' is a more economical expression than 'industry industries'.
(? <i>pattern</i> )	A subexpression that performs a positive lookahead search, which matches the string at any point where a string matching <i>pattern</i> begins. This is a non-capturing match, that is, the match is not captured for possible later use. For example 'Windows (?=95 98 NT 2000)' matches "Windows" in "Windows 2000" but not "Windows" in "Windows 3.1". Lookaheads do not consume characters, that is, after a match occurs, the search for the next match begins immediately following the last match, not after the characters that comprised the lookahead.
(?! <i>pattern</i> )	A subexpression that performs a negative lookahead search, which matches the search string at any point where a string not matching <i>pattern</i> begins. This is a non-capturing match, that is, the match is not captured for possible later use. For example 'Windows (?!95 98 NT 2000)' matches "Windows" in "Windows 3.1" but does not match "Windows" in "Windows 2000". Lookaheads do not consume characters, that is, after a match occurs, the search for the next match begins immediately following the last match, not after the characters that comprised the lookahead.
<i>x y</i>	Matches either <i>x</i> or <i>y</i> . For example, 'z food' matches "z" or "food". '(z f)ood' matches "zood" or "food".
[ <i>xyz</i> ]	A character set. Matches any of the enclosed characters. For example, '[abc]' matches the 'a' in "plain".
[^ <i>xyz</i> ]	A negative character set. Matches any character not enclosed. For example, '[^abc]' matches the 'p' in "plain".
[ <i>a-z</i> ]	A range of characters. Matches any character in the specified range. For example, '[a-z]' matches any lowercase alphabetic character in the range 'a' through 'z'.
[^ <i>a-z</i> ]	A negative range characters. Matches any character not in the specified range. For example, '[^a-z]' matches any character not in the range 'a' through 'z'.
\b	Matches a word boundary, that is, the position between a word and a space. For example, 'er\b' matches the 'er' in "never" but not the 'er' in "verb".
\B	Matches a nonword boundary. 'er\B' matches the 'er' in "verb" but not the 'er' in "never".
\cx	Matches the control character indicated by <i>x</i> . For example, \cM matches a Control-M or carriage return character. The value of <i>x</i> must be in the range of A-Z or a-z. If not, <i>c</i> is assumed to be a literal 'c' character.
\d	Matches a digit character. Equivalent to [0-9].
\D	Matches a nondigit character. Equivalent to [^0-9].
\f	Matches a form-feed character. Equivalent to \x0c and \cL.
\n	Matches a newline character. Equivalent to \x0a and \cJ.
\r	Matches a carriage return character. Equivalent to \x0d and \cM.
\s	Matches any white space character including space, tab, form-feed, and so on. Equivalent to [\f\n\r\t\v].
\S	Matches any non-white space character. Equivalent to [^\f\n\r\t\v].

**Table 18-7 (Cont.) Metacharacters for Regular Expressions**

Character	Description
\t	Matches a tab character. Equivalent to \x09 and \cl.
\v	Matches a vertical tab character. Equivalent to \x0b and \cK.
\w	Matches any word character including underscore. Equivalent to '[A-Za-z0-9_]'
\W	Matches any nonword character. Equivalent to '[^A-Za-z0-9_]'
\xn	Matches <i>n</i> , where <i>n</i> is a hexadecimal escape value. Hexadecimal escape values must be exactly two digits long. For example, '\x41' matches "A". '\x041' is equivalent to '\x04' & "1". Allows ASCII codes to be used in regular expressions.
\num	Matches <i>num</i> , where <i>num</i> is a positive integer. A reference back to captured matches. For example, '(.)\1' matches two consecutive identical characters.
\n	Identifies either an octal escape value or a backreference. If <i>n</i> is preceded by at least <i>n</i> captured subexpressions, <i>n</i> is a backreference. Otherwise, <i>n</i> is an octal escape value if <i>n</i> is an octal digit (0-7).
\nm	Identifies either an octal escape value or a backreference. If <i>nm</i> is preceded by at least <i>nm</i> captured subexpressions, <i>nm</i> is a backreference. If <i>nm</i> is preceded by at least <i>n</i> captures, <i>n</i> is a backreference followed by literal <i>m</i> . If neither of the preceding conditions exists, <i>nm</i> matches octal escape value <i>nm</i> when <i>n</i> and <i>m</i> are octal digits (0-7).
\nml	Matches octal escape value <i>nml</i> when <i>n</i> is an octal digit (0-3) and <i>m</i> and <i>l</i> are octal digits (0-7).
\un	Matches <i>n</i> , where <i>n</i> is a Unicode character expressed as four hexadecimal digits. For example, \u00A9 matches the copyright symbol (©).

## Using the Loadgen Utility

This section describes how to use the Loadgen Utility included with the Oracle BAM 14c release.

It includes the following subsections:

- [Overview of Loadgen](#)
- [Configuration and Execution](#)
- [Usage Examples](#)

## Overview of Loadgen

The Loadgen Utility, or load generator, is a simple utility provided by installing the Oracle BAM product and used to simulate a data feed into the BAM Server. It specifically acts on the Persistence Layer using the Enterprise Java Bean (EJB) interface. It is useful for testing the Event Processing Language (EPL) rules in an application without needing to connect to a real-world data feed. It can generate loads for both stream and archived relation data objects (DOs).

The load generator reads one or more XML files that contain the sample data feed information and sends each data item to the configured port. It reads items from the sample data file in order and inserts them into the stream, looping around to the beginning of the data file when it reaches the end; this ensures that a continuous stream of data is available, regardless of the

number of data items in the file. One can configure the rate of sent data, from the rate at which it starts, the final rate, and how long it takes the load generator to ramp up to the final rate.

## Loadgen Features

The load generator is expected to be used during Dashboard testing. This is particularly useful for functional and load testing to simulate production like load scenarios.

The utility supports the following use-case features:

- Simulating a data feed into the system to support functional testing for the data accuracy of the system. For example, one can use it to feed data to test CQL templates, such as a pattern match with a stream DO or a dashboard with an archived relation DO.
- Specifying a looping construct to generate ongoing loads.
- Specifying insert, delete and update events for an archived relation DO, and insert events for a stream DO.
- Specifying a filter condition to update only certain events.
- Specifying an initial injection rate and the final injection rate. The period between the initial rate to the final rate is called the *ramp up period*. After it reaches the final rate state, it injects data at a constant rate until the time period expires.
- Specifying multiple consumer threads to achieve a high throughput rate.
- Specifying input data with one or more XML files. The XML files are processed in order. There must be a target DO associated with each event.
- Specifying batch operations to group multiple events into a single operation.
- Running Loadgen from a command line in a plain text control file as described in [Running Loadgen With a Command Line in a Control File](#).
- Specifying the frequency to inject using a fixed period thus accommodating *bursts* of loads that simulate production deployments, as fully described in [Usage Examples](#).

## Configuration and Execution

The `bamloadgen` command can be executed from the `FMW_HOME\soa\bam\bin` directory on the Windows platform and from `FMW_HOME/soa/bam/bin` on UNIX platforms.

Running Loadgen involves the following configuration and execution concerns:

- Run Loadgen with a command line that calls `bamloadgen`. The command can be run from an optional executable text file we term a *control file*, or one can run it directly from a command prompt.
- The Oracle BAM servers must be running and accessible.
- The related DOs must be pre-existing in the system and accessible.
- The system displays runtime messages for the following conditions:
  - The normal processing status, such as the current injection rate.
  - Any exceptions occurring during processing, such as if server is not available, or there is a data mismatch or invalid stream DO in XML file.
- For an Archived Relation DO update, delete and insert operations are supported, while for a Stream DO only insert operations are supported.

- Input XML file in XSD format: One can either specify `$currentTime` as a date value for the system to capture its locale time as input, or one can specify an exact date value, such as "2013-04-29T12:45:00.000-8:00". For XML examples see [CRUD XML Insert\\_ Update\\_ and Delete Examples](#).

## Command-Line Syntax

The command-line calling `bamloadgen` can be executed directly from a command prompt or it can be run from an executable plain text control file. The command-line syntax supports the generation and control of data flow for any of the following: the specific initial rate, the final rate, ramp-up time, duration and the number of threads. The data source can come from one or more XML files. Syntax examples are as follows for each type of run mode with each variable option value identified in brackets (<...>) which are omitted in the actual command line. See [Table 18-8](#) for a complete descriptive list of available options.

- **Frequency mode without rate ramp-up:**

```
bamloadgen -XMLFile XMLFile -consumerThread_num thread_num
          -batch_size batch_size -frequency frequency -duration duration
```

- **Ramp-up mode without frequency:**

```
bamloadgen -XMLFile XMLFile -consumerThread_num thread_num
          -state_interval state_interval -ramp_up ramp_up -initial_rate
          initial_rate -final_rate final_rate -duration duration
```

**Table 18-8 Available Command-Line Options**

Option	Description	Data Type	Required
XMLFile	The XML files to read into the data object, separate multiple XML files by commas.	String	Yes
batch_size	The number of events grouped in a single operation, default value = 100.	Integer	No
consumerThread_num	The number of threads for each process operation, default value = 1.	Integer	No
queue_size	The size of queue, default value = 100.	Integer	No
frequency	The interval to inject data once in frequency mode, cannot be used together with ramp_up, state_interval, initial_rate, final_rate, default value = 1.	Integer	No
ramp_up	The number in seconds for the load generation to increase from the initial to the final rate, default value = 1.	Integer	No
state_interval	The time for the rate to ramp up from one state to another state, required with ramp_up mode, unit: seconds, default value = 1.	Integer	No
initial_rate	The initial rate of operation, unit: # of event/second, required with ramp_up mode.	Integer	No
final_rate	The final rate of operation, unit: # of event/second, required with ramp_up mode.	Integer	No
duration	Length of time in seconds load generation should continue to run, use 0 to run it indefinitely.	Integer	Yes
username	Specifies the name of the user to be used when logging into the BAM server.	String	Yes
password	Specifies the password for the user specified by -username.	String	Yes

**Table 18-8 (Cont.) Available Command-Line Options**

Option	Description	Data Type	Required
host	Specifies the host name or IP address of the target BAM server. Default: localhost.	String	No
port	Specifies the port number of the target BAM server. Default: 7001.	Integer	No
protocol	Specifies the communication protocol to be used. Can be t3 or t3s for SSL. Default: t3.	String	No
url	Specifies the URL of the target BAM server as an alternative to using <code>-host</code> , <code>-port</code> and <code>-protocol</code> . Default: <code>t3://localhost:7001</code> .	String	No
file	Specifies a control file containing command line arguments to run <code>bamloadgen</code> . See <a href="#">Running Loadgen With a Command Line in a Control File</a> .	String	No
retry_count	The number of retry attempts that should be made to re-establish a failed session. Default: 30.	Integer	No
retry_interval	The number of milliseconds to wait between retry attempts to re-establish a failed session. Default: 1000.	Integer	No

## Usage Examples

This sections shows examples of various usage scenarios.

### Setting the `-frequency` to Inject Data in Fixed Periods at a Stable Rate

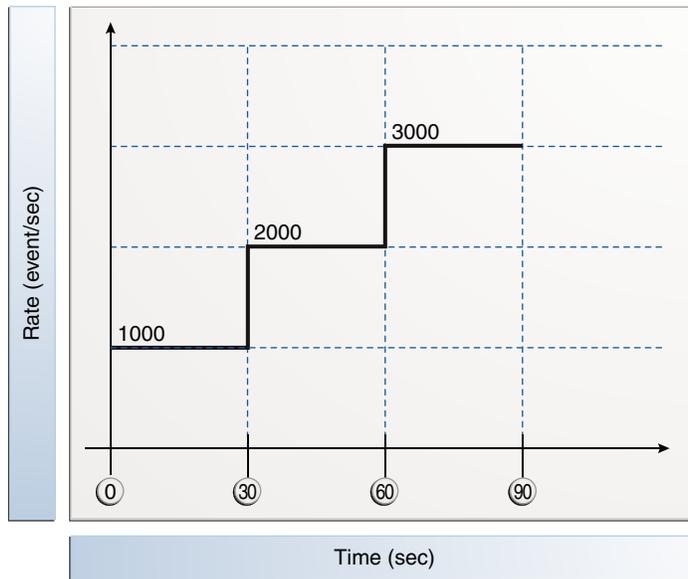
This command injects data once every 2 seconds (`-frequency 2`), grouping 10 records per injection (`-batch_size 10`) for a total of 20 seconds (`-duration 20`). The entire process computes to these records:  $(20/2)*10 = 100$  records inserted.

```
bamloadgen -XMLFile CALLCENTER_FACT_export.xml -frequency 2 -batch_size 10 -duration 20
```

### Inject at a Rate From Initial Rate to Final Rate with a Single File

```
bamloadgen -XMLFile example -batch_size 100 -state_interval 30 -ramp_up 60  
-initial_rate 1000 -final_rate 3000 -duration 90
```

This command line reads in the data specified by `example1.xml` by a speed starting from 1000 records/sec and reaches a final rate of 3000 records/sec within 60 seconds. The period between different rates is 30 seconds (per `-state_interval 30`), so it takes 2 steps to reach the final rate (`-ramp_up` or `-state_interval = 2`). After reaching the final rate, loadgen keeps running at the final rate until it ends at the 90th second (`-duration 90`). [Figure 18-1](#) depicts this.

**Figure 18-1 Example Injection Rate Changes, Initial to Final With a Single File**

## Multiple Threads for Higher Throughput

Specify `-consumerThread_num 10` to have 10 threads to process the operation together.

### Note:

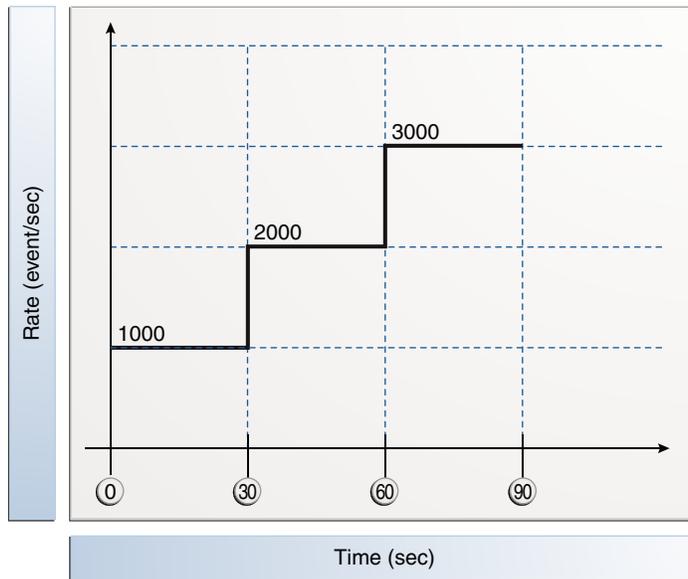
It might not necessarily achieve a higher rate depending on the local system environment. In testing, `-consumerThread_num 20` achieved the best injection rate, and changing to 30 had the same rate as with 20.

```
bamloadgen -XMLFile example1.xml,example2.xml -batch_size 100
-consumerThread_num 20 -state_interval 30 -ramp_up 60
-initial_rate 2000 -final_rate 3000 -duration 90 -username weblogic
```

## Multiple XML Input Files

```
bamloadgen -XMLFile example1.xml,example2.xml -batch_size 100 -thread_num 10
-state_interval 30 -ramp_up 60 -initial_rate 1000 -final_rate 3000 -duration 90"
```

This command line reads in the data specified by `example1.xml` and `example2.xml` generating two different producer threads and injects the data to the server. Because there are two producer threads injecting data, each thread runs at  $\frac{3}{4}$  of the total rate, so it goes from 500 -> 1000 -> 1500 (event/sec). After injecting the data, the back-end uses 10 consumer threads to process the data operations. [Figure 18-2](#) depicts this.

**Figure 18-2 Example Injection Rate Changes With Multiple Input Files**

## Batching Operations to Group Multiple Events Into a Single Operation

One can specify `-batch_size 100` to process 100 data entries in one operation. So if the XML file has 500 data entries then each thread will process 100 in its operation.

```
bamloadgen -XMLFile example_export.xml -batch_size 100 - consumerThread_num 10
-state_interval 30 -ramp_up 60 -initial_rate 1000 -final_rate 3000 -duration 90
```

## Simulate Data for Functional Testing

This command line feeds data to test CQL templates, like a pattern match for a stream DO. If the stream DO includes delete or update, then an exception is thrown.

```
bamloadgen -XMLFile example1.xml -batch_size 20 -thread_num 10 -state_interval 30
-ramp_up 60 -initial_rate 50 -final_rate 100 -duration 0
```



### Note:

One can specify `-duration 0` in the command line to make the data feed loop run indefinitely.

## Operating on an Archive Relation DO and a Stream DO

This example specifies insert, delete and update events for an Archived Relation DO and an insert event for a Stream DO. For the update events, one can specify a filter condition for updating certain events. The following command-line runs this example:

```
bamloadgen -XMLFile example1.xml -batch_size 20 -thread_num 10 -state_interval 30
-ramp_up 60 -initial_rate 50 -final_rate 100 -duration 90
```

The following code shows the XML file for Loadgen to update DO xyz, insert and delete DO abc. Both the update and delete data entries have a `filter-columns` element to process the operations for the corresponding data entry into the database. To operate on an Archive Relation and Stream DO, use the following XML.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dataObjectOperations>
  <operations>
    <update dataobject="xyz">
      <filter-columns>
        <string value="STRING_82" predicate="EQ" name="String_Column"/>
      </filter-columns>
      <update-columns>
        <string value="UPDATED" action="REPLACE" name="String_Column"/>
        <integer value="111" action="REPLACE" name="Integer_Column"/>
        <decimal value="111.10" action="REPLACE" name="Decimal_Column"/>
        <float value="111.11" action="REPLACE" name="Float_Column"/>
        <date value="2011-05-15T09:40:02.111Z" action="REPLACE"
          name="Date_Column"/>
      </update-columns>
    </update>
    <insert dataobject="abc">
      <insert-columns>
        <date value="2011-05-15T09:40:02.111Z" name="Date_Column"/>
        <decimal value="670.05" name="Decimal_Column"/>
        <float value="52.107746" name="Float_Column"/>
        <integer value="91" name="Integer_Column"/>
        <string value="STRING_36" name="String_Column"/>
      </insert-columns>
    </insert>
    <delete dataobject="xyz">
      <filter-columns>
        <string value="STRING_61" predicate="EQ" name="String_Column"/>
      </filter-columns>
    </delete>
  </operations>
</dataObjectOperations>
```

## Running Loadgen With a Command Line in a Control File

Run the following at a command line to run a control file named `controlFileExample.txt`:

```
./bamloadgen -file controlFileExample.txt
```

The control file might then simply contain the following content depending on the intention:

```
bamloadgen -XMLFile example_export.xml -batch_size 100 - consumerThread_num 10
-state_interval 30 -ramp_up_60 -initial_rate 1000 -final_rate 3000 -duration 90
```

## CRUD XML Insert, Update, and Delete Examples

See the following examples:

- The CRUD XML for inserting data

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dataObjectOperations>
  <operations>
    <insert dataobject="PATTERNMATCH_FACT">
      <insert-columns>
        <date value="$currentTime" name="callCreatedTime"/>
      </insert-columns>
    </insert>
  </operations>
</dataObjectOperations>
```

```

        <date value="$currentTime" name="callClosedTime"/>
        <integer value="1" name="callPriority"/>
        <string value="2" name="productId"/>
        <integer value="100" name="callWaitTime"/>
        <integer value="60" name="callProcessingTime"/>
        <string value="US" name="customerLocationId"/>
        <string value="CLOSED" name="callStatus"/>
        <string value="OPEN" name="customerStatus"/>
    </insert-columns>
</insert>
<insert dataobject="PATTERNMATCH_FACT">
    <insert-columns>
        <date value="$currentTime" name="callCreatedTime"/>
        <date value="$currentTime" name="callClosedTime"/>
        <integer value="1" name="callPriority"/>
        <string value="2" name="productId"/>
        <integer value="200" name="callWaitTime"/>
        <integer value="120" name="callProcessingTime"/>
        <string value="CA" name="customerLocationId"/>
        <string value="WAIT" name="callStatus"/>
        <string value="OPEN" name="customerStatus"/>
    </insert-columns>
</insert>
</operations>
</dataObjectOperations>

```

- Here is the CRUD XML for updating data:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dataObjectOperations>
    <operations>
        <update dataobject="xyz">
            <filter-columns>
                <string value="STRING_82" predicate="EQ" name="String_Column"/>
            </filter-columns>
            <update-columns>
                <string value="UPDATED" action="REPLACE" name="String_Column"/>
                <integer value="111" action="REPLACE" name="Integer_Column"/>
                <decimal value="111.10" action="REPLACE" name="Decimal_Column"/>
                <float value="111.11" action="REPLACE" name="Float_Column"/>
                <date value="1901-01-01T01:11:11.111-8:00" action="REPLACE"
                    name="Date_Column"/>
            </update-columns>
        </update>
    </operations>
</dataObjectOperations>

```

- Here is the CRUD XML for deleting data:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dataObjectOperations>
    <operations>
        <delete dataobject="xyz">
            <filter-columns>
                <string value="STRING_1" predicate="EQ" name="String_Column1"/>
            </filter-columns>
        </delete>
        <delete dataobject="xyz">
            <filter-columns>
                <string value="STRING_2" predicate="EQ" name="String_Column2"/>
            </filter-columns>
        </delete>
    </operations>
</dataObjectOperations>

```

```
</operations>  
</dataObjectOperations>
```

# Using Oracle BAM Web Services

This chapter describes how to use Oracle Business Activity Monitoring (Oracle BAM) web services such as `DOOperations`, `DODefinition`, and `ManualRuleFire` to build applications that publish data to the Oracle BAM Server for use in dashboards.

This chapter includes the following sections:

- [Understanding Oracle BAM Web Services](#)
- [Using the Legacy Data Object Web Services](#)
- [Using the Data Object Web Services](#)
- [Using the ManualRuleFire Web Service](#)

## Understanding Oracle BAM Web Services

The Oracle BAM web services allow users to build applications that publish data to the Oracle BAM Server for use in dashboards. Any client that can talk to standard web services can use these APIs to publish data to Oracle BAM.

The Oracle BAM web services interfaces allow integration of Oracle BAM with other components such as Oracle BPEL Process Manager and Oracle Mediator, and they facilitate SOA composite application development.



### Note:

This option cannot be used for complex processing of messages, performing lookups in Oracle BAM to augment the data, or initial bulk uploads to set up a star schema.

The data objects in the Oracle BAM Server are available using the Oracle BAM web services.

External web services can be called by an Oracle BAM alert rule. See [Creating an Alert](#) for more information.

Oracle BAM provides the following static untyped web service APIs:

- **DataObjectOperationsByName** allows developers to interact with data objects by their display names. For example, Call Center performs Delete, Get, Insert, Update, or Upsert operations on rows in data objects migrated from BAM 12c within a BAM 14c server.
- **DataObjectOperationsByID** allows developers to interact with data objects by their IDs. For example, Call\_Center performs Batch, Delete, Insert, Update, or Upsert operations on rows in data objects migrated from BAM 12c within a BAM 14c server.
- **DOOperations** performs Batch, Delete, Get, Insert, Update, or Upsert operations on rows in BAM 14c data objects.
- **DODefinition** performs Create, Delete, Get, or Update operations on definitions of BAM 14c data objects.

- **ManualRuleFire** can be used by other Oracle BAM services to launch rules created in Oracle BAM Composer.

These services can be discovered within an Oracle BAM Server using a WSIL interface.

In addition, you can view definitions (metadata) for all BAM data objects using the following WSIL interface:

```
http://host:port/OracleBAMWS/WebServices/Wsil/baminspection.wsil
```

## Using the Legacy Data Object Web Services

The legacy web services allow users to manipulate BAM 12c Data Objects in the Oracle BAM 14c Server by inserting, updating, deleting, and upserting rows into the Data Objects.

The following operations are supported by these web service interfaces.

- **Batch** performs batch operations on a data object. Batch is not available in the `DataObjectOperationsByName` web service.
- **Delete** removes a row from the data object.
- **Get** fetches the details from a data object per the specifications in the XML payload. Get is not available in the `DataObjectOperationsByID` web service.
- **Insert** adds a row to the data object.
- **Update** inserts new data into an existing row in a data object.
- **Upsert** inserts new data into an existing row in a data object if the row exists. If the row does not exist, a new row is created.

The request and response messages vary depending on the operation used. See [DataObjectOperationsByName](#) and [DataObjectOperationsByID](#) for information about using the operations supported by each of the web services.

## How to Use the Legacy Data Object Web Services

To use these web services, create a web service proxy in your application in Oracle JDeveloper.

The Web Services Description Language (WSDL) files for the `DataObjectOperations` web services are available at the following URLs on the system where Oracle BAM web services are installed.

```
http://host:port/OracleBAMWS/WebServices/DataObjectOperationsByName?WSDL
```

```
http://host:port/OracleBAMWS/WebServices/DataObjectOperationsByID?WSDL
```

### Note:

The default port for Oracle BAM web services on the Administration Server is 7001. On managed servers the default port number is 9001. You can specify any port number while installing Oracle BAM.

## DataObjectOperationsByName

The following operations are supported by the DataObjectOperationsByName web service for BAM 11g Data Objects.

- [Delete](#)
- [Get](#)
- [Insert](#)
- [Update](#)
- [Upsert](#)

### Delete

Delete removes a row from the data object.

### Request Message

The request message contains the following parameters.

**keysCSV (xsd:string)**

Comma separated column IDs that must be used as keys, for example:

```
Sales Number, Sales Area
```

**xmlPayload (xsd:string)**

Payload for the where clause to delete rows in a data object. Here is a sample Delete payload:

```
<DataObject Name="Employees" Path="/Samples">
  <Contents>
    <Row>
      <Column Name="Salesperson" Value="Greg Guan" />
    </Row>
  </Contents>
</DataObject>
```

### Get

Get fetches the details from a data object per the specifications in the XML payload

### Request Message

The request message contains the following parameters.

**keysCSV (xsd:string)**

Comma separated column IDs that must be used as keys, for example:

```
Sales Number, Sales Area
```

**xmlPayload (xsd:string)**

The payload specifies what to get from the data object.

For the `DataObjectOperationsByName` web service the data object name is specified in the payload. Here is a sample Get payload:

```
<DataObject Name="Employees" Path="/Samples">
  <Contents>
    <Row>
      <Column Name="Salesperson" Value="Greg Masters"/>
    </Row>
  </Contents>
</DataObject>
```

## Insert

Insert adds rows to the specified data object.

## Request Message

The request message contains the following parameters.

### **xmlPayload (xsd:string)**

The payload is specific to each data object. Here is a sample Insert payload.

```
<DataObject Name="Employees" Path="/Samples">
  <Contents>
    <Row>
      <Column Name="Salesperson" Value="Greg Guan" />
      <Column Name="Sales Area" Value="Northeast" />
      <column Name="Sales Number" Value="5671" />
    </Row>
  </Contents>
</DataObject>
```

## Update

Update operation updates existing data with new data in a data object.

## Request Message

The request message contains the following parameters.

### **keysCSV (xsd:string)**

Comma separated column IDs that must be used as keys, for example:

```
Sales Number, Sales Area
```

### **xmlPayload (xsd:string)**

Payload for the update statement and where clause to update rows in a data object. Here is a sample of an Update payload:

```
<DataObject Name="Employees" Path="/Samples">
  <Contents>
    <Row>
      <Column Name="Salesperson" Value="Greg Guan" />
    </Row>
  </Contents>
</DataObject>
```

## Upsert

Upsert operation updates existing data with new data in an existing row in a data object. If the row does not exist a new row is created.

### Request Message

The request message contains the following parameters.

**keysCSV (xsd:string)**

Comma separated column IDs that must be used as keys, for example:

```
Sales Number, Sales Area
```

**xmlPayload (xsd:string)**

Payload for the insert or update statement and where clause to upsert rows in a data object. Here is a sample of an Upsert payload:

```
<DataObject Name="Employees" Path="/Samples">
  <Contents>
    <Row>
      <Column Name="Salesperson" Value="Greg Guan" />
      <Column Name="Sales Area" Value="Northeast" />
      <column Name="Sales Number" Value="5671" />
    </Row>
  </Contents>
</DataObject>
```

## DataObjectOperationsByID

The following operations are supported by the DataObjectOperationsByID web service for BAM 11g Data Objects.

- [Batch](#)
- [Delete](#)
- [Insert](#)
- [Update](#)
- [Upsert](#)

### Batch

Batch performs batch operations on a data object.

### Request Message

The request message contains the following parameters.

**dataObject (xsd:string)**

Full relative path and name of the data object, for example:

```
/Samples/Employees
```

**xmlPayload (xsd:string)**

Contains the batch payload for any operations to be performed. Here is a sample Batch payload:

```
<payload>
  <_Employees operation="insert">
    <_Salesperson>Tim Bray</_Salesperson>
    <_Sales_Area>EMEA</_Sales_Area>
    <_Sales_Number>12345</_Sales_Number>
  </_Employees>
  <_Employees operation="update" keys="_Sales_Number">
    <_Salesperson>Tim Bray</_Salesperson>
    <_Sales_Area>EMEA</_Sales_Area>
    <_Sales_Number>12345</_Sales_Number>
  </_Employees>
</payload>
```

## Delete

Delete removes a row from the data object.

## Request Message

The request message contains the following parameters.

**dataObject (xsd:string)**

This parameter is not required by the DataObjectOperationsByName web service because the data object name and path are part of the payload.

Full relative path and name of the data object, for example:

```
/Samples/Employees
```

**keysCSV (xsd:string)**

Comma separated column IDs that must be used as keys, for example:

```
_Sales_Number,_Sales_Area
```

**xmlPayload (xsd:string)**

Payload for the where clause to delete rows in a data object. For example:

```
<_Employees>
  <_Sales_Number>12345</_Sales_Number>
</_Employees>
```

## Insert

Insert adds rows to the specified data object.

## Request Message

The request message contains the following parameters.

**dataObject (xsd:string)**

Full relative path and name of the data object, for example:

```
/Samples/Employees
```

**xmlPayload (xsd:string)**

The payload is specific to each data object.

For the DataObjectOperationsByName web service the data object name is specified in the payload, for example:

```
<_Employees>  
  <_Salesperson>Time Bray</_Salesperson>  
  <_Sales_Area>EMEA</_Sales_Area>  
  <_Sales_Number>12345</_Sales_Number>  
</_Employees>
```

## Update

Update operation updates existing data with new data in a data object.

### Request Message

The request message contains the following parameters.

**dataObject (xsd:string)**

Full relative path and name of the data object, for example:

```
/Samples/Employees
```

**keysCSV (xsd:string)**

Comma separated column IDs that must be used as keys, for example:

```
_Sales_Number, _Sales_Area
```

**xmlPayload (xsd:string)**

Payload for the update statement and where clause to update rows in a data object. For example:

```
<_Employees>  
  <_Sales_Area>Asia Pacific</_Sales_Area>  
  <_Sales_Number>12345</_Sales_Number>  
</_Employees>
```

## Upsert

Upsert operation updates existing data with new data in an existing row in a data object. If the row does not exist a new row is created.

### Request Message

The request message contains the following parameters.

**dataObject (xsd:string)**

Full relative path and name of the data object, for example:

```
/Samples/Employees
```

**keysCSV (xsd:string)**

Comma separated column IDs that must be used as keys, for example:

```
_Sales_Number, _Sales_Area
```

**xmlPayload (xsd:string)**

Payload for the insert or update statement and where clause to upsert rows in a data object. For example:

```
<_Employees>  
  <_Salesperson>Time Bray</_Salesperson>  
  <_Sales_Area>EMEA</_Sales_Area>  
  <_Sales_Number>12345</_Sales_Number>  
</_Employees>
```

## Using the Data Object Web Services

The Data Object web services allow a web service client to create, update, delete, and get BAM 14c data object rows and definitions.

The following operations are supported by these web services.

- **Batch** performs batch operations on a data object. Batch is not available in the DODefinition web service.
- **Create** creates a data object. Create is only available in the DODefinition web service.
- **Delete** removes a data object row or definition.
- **Get** fetches the details from a data object per the specifications in the XML payload.
- **Insert** adds a row to the data object. Insert is not available in the DODefinition web service.
- **Update** inserts new data into an existing row in a data object or updates a data object definition.
- **Upsert** inserts new data into an existing row in a data object if the row exists. If the row does not exist, a new row is created. Upsert is not available in the DODefinition web service.

The request and response messages vary depending on the operation used. See [DOOperations](#) and [DODefinition](#) for more information.

## How to Use the Data Object Web Services

To use these web services, you create a web service proxy in your application in Oracle JDeveloper.

The WSDL files for the DOOperations and DODefinition web services are available at the following URLs on the system where Oracle BAM web services are installed.

```
http://host:port/OracleBAMWS/WebServices/DOOperations?WSDL
```

```
http://host:port/OracleBAMWS/WebServices/DODefinition?WSDL
```

**Note:**

The default port for Oracle BAM web services on the Administration Server is 7001. On managed servers the default port number is 9001.

## DOOperations

The following operations are supported by DOOperations web service.

- [Batch](#)
- [Delete](#)
- [Get](#)
- [Insert](#)
- [Update](#)
- [Upsert](#)

## Batch

Batch performs batch operations on a data object.

## Request Message

The request message contains the following parameters.

### **dataObject (xsd:string)**

Full relative path and name of the data object, for example:

```
/Samples/Employees
```

### **xmlPayload (xsd:string)**

Contains the batch payload for any operations to be performed. Here is a sample DOOperations Batch payload.

```
<dataObjectOperations>
  <operations>
    <upsert dataobject="Sales">
      <filter-columns>
        <string value="Greg Guan" name="Salesperson" predicate="EQ"/>
      </filter-columns>
      <update-columns>
        <string value="Hanumesh" name="Salesperson"/>
        <string value="Northeast" name="SalesArea"/>
        <integer value="70000" name="SalesNumber"/>
      </update-columns>
      <insert-columns>
        <string value="Greg Guan" name="Salesperson"/>
        <string value="Northeast" name="SalesArea"/>
        <integer value="50000" name="SalesNumber"/>
      </insert-columns>
    </upsert>
    <insert dataobject="Sales">
      <insert-columns>
```

```

        <string value="Greg Guan1" name="Salesperson"/>
        <string value="Northeast" name="SalesArea"/>
        <integer value="50000" name="SalesNumber"/>
    </insert-columns>
</insert>
</operations>
</dataObjectOperations>

```

## Delete

Delete removes a row from the data object.

## Request Message

The request message contains the following parameter.

### xmlPayload (xsd:string)

Payload for the where clause to delete rows in a data object. Here is a sample Delete DOOperations payload:

```

<dataObjectOperations>
  <operations>
    <delete dataobject="Sales">
      <filter-columns>
        <string value="Hanumesh1" name="Salesperson" predicate="EQ"/>
      </filter-columns>
    </delete>
  </operations>
</dataObjectOperations>

```

## Get

Get fetches the details from a data object per the specifications in the XML payload.

## Request Message

The request message contains the following parameters.

### keysCSV (xsd:string)

Comma-separated column IDs must be used as keys, for example:

```
/Salesperson,Sales
```

### xmlPayload (xsd:string)

The payload specifies what to get from the data object.

For the DOOperations web service the data object name is specified in the payload, for example:

```

<DataObject Name="Sales">
  <Contents>
    <Row>
      <Column Name="Salesperson" Value="Hanumesh"/>
    </Row>
  </Contents>
</DataObject>

```

## Insert

Insert adds rows to the specified data object.

### Request Message

The request message contains the following parameter.

#### **xmlPayload (xsd:string)**

The payload is specific to each data object. Here is a sample DOOperations Insert Payload:

```
<dataObjectOperations>
  <operations>
    <insert dataobject="Sales">
      <insert-columns>
        <string value="Greg Guan" name="Salesperson"/>
        <string value="Northeast" name="SalesArea"/>
        <integer value="50000" name="SalesNumber"/>
      </insert-columns>
    </insert>
  </operations>
</dataObjectOperations>
```

## Update

Update updates existing data with new data in a data object.

### Request Message

The request message contains the following parameter.

#### **xmlPayload (xsd:string)**

The payload for the update statement includes a where clause to update the rows in a data object. For example:

```
<dataObjectOperations>
  <operations>
    <update dataobject="Sales">
      <filter-columns>
        <string value="Greg Guan" name="Salesperson" predicate="EQ"/>
      </filter-columns>
      <update-columns>
        <string value="Hanumesh" name="Salesperson"/>
        <string value="Northeast" name="SalesArea"/>
        <integer value="60000" name="SalesNumber"/>
      </update-columns>
    </update>
  </operations>
</dataObjectOperations>
```

## Upsert

Upsert updates existing data with new data in an existing row in a data object. If the row does not exist, a new row is created.

## Request Message

The request message contains the following parameter.

### **xmlPayload (xsd:string)**

Payload for the insert or update statement and a where clause to upsert rows in a data object. For example:

```

<dataObjectOperations>
  <operations>
    <upsert dataobject="Sales">
      <filter-columns>
        <string value="Greg Guan" name="Salesperson" predicate="EQ"/>
      </filter-columns>
      <update-columns>
        <string value="Hanumesh" name="Salesperson"/>
        <string value="Northeast" name="SalesArea"/>
        <integer value="70000" name="SalesNumber"/>
      </update-columns>
      <insert-columns>
        <string value="Greg Guan" name="Salesperson"/>
        <string value="Northeast" name="SalesArea"/>
        <integer value="50000" name="SalesNumber"/>
      </insert-columns>
    </upsert>
  </operations>
</dataObjectOperations>

```

## DODefinition

The following operations are supported by DODefinition web service.

- [Create](#)
- [Delete](#)
- [Get](#)
- [Update](#)

## Create

Create creates a new data object. You can create all four data object types: simple, derived, logical, and external.

## Request Message

The request message contains the following parameter.

### **xmlPayload (xsd:string)**

Contains the payload to create a data object. For example:

```

<ns0:simpleDataObject xmlns:ns0="http://xmlns.oracle.com/bam/bam12">
  <name>Sales</name>
  <displayName>Sales</displayName>
  <cqType>RELATION</cqType>
  <cqArchived>true</cqArchived>
  <hidden>>false</hidden>

```

```

<columns>
  <column>
    <name>Salesperson</name>
    <id>1621481da151473caedc1a66c19dc2e6</id>
    <displayName>Salesperson</displayName>
    <length>2000</length>
    <nullable>true</nullable>
    <unique>false</unique>
    <dataType>VARCHAR</dataType>
    <columnType>ATTRIBUTE</columnType>
    <hidden>false</hidden>
    <editable>true</editable>
  </column>
</columns>
</ns0:simpleDataObject>

```

Here is the xmlPayload to create a Derived Data Object:

```

<ns0:derivedDataObject xmlns:ns0="http://xmlns.oracle.com/bam/bam12">
  <name>DSales</name>
  <displayName>DSales</displayName>
  <internal>false</internal>
  <cqType>RELATION</cqType>
  <cqArchived>true</cqArchived>
  <hidden>false</hidden>
  <columns>
    <column>
      <name>Discount</name>
      <id>20a032f502dc4d38b9889a57d081be15</id>
      <displayName>Discount</displayName>
      <internal>false</internal>
      <length>2000</length>
      <nullable>true</nullable>
      <unique>false</unique>
      <precision>38</precision>
      <scale>10</scale>
      <dataType>DECIMAL</dataType>
      <columnType>ATTRIBUTE</columnType>
      <hidden>false</hidden>
      <editable>true</editable>
    </column>
  </columns>
  <parent type="dataObjectRef">Sales</parent>
</ns0:derivedDataObject>

```

Here is the xmlPayload to create a Logical Data Object:

```

<ns0:logicalDataObject xmlns:ns0="http://xmlns.oracle.com/bam/bam12">
  <name>LSales</name>
  <displayName>LSales</displayName>
  <timeHierarchies/>
  <dataObjects>
    <dataObject type="dataObjectRef">DSales</dataObject>
    <dataObject type="dataObjectRef">Sales</dataObject>
  </dataObjects>
  <columns>
    <column>
      <name>Salesperson</name>
      <displayName>Salesperson</displayName>
      <internal>false</internal>
      <columnRef>

```

```

        <dataObject>Sales</dataObject>
        <column>Salesperson</column>
    </columnRef>
    <hidden>false</hidden>
    <columnType>ATTRIBUTE</columnType>
</column>
<column>
    <name>Discount</name>
    <displayName>Discount</displayName>
    <internal>false</internal>
    <columnRef>
        <dataObject>DSales</dataObject>
        <column>Discount</column>
    </columnRef>
    <hidden>false</hidden>
    <columnType>ATTRIBUTE</columnType>
</column>
</columns>
<joins>
    <join>
        <id>b2d4e97ba8be400a847aaa354fc7b1a0</id>
        <internal>false</internal>
        <leftColumns>
            <column>
                <dataObject>Sales</dataObject>
                <column>BEAM_ID</column>
            </column>
        </leftColumns>
        <rightColumns>
            <column>
                <dataObject>DSales</dataObject>
                <column>BEAM_ID</column>
            </column>
        </rightColumns>
    </join>
</joins>
</ns0:logicalDataObject>

```

Here is the xmlPayload to create an External Data Object:

```

<ns0:externalDataObject xmlns:ns0="http://xmlns.oracle.com/bam/bam12">
    <name>ESales</name>
    <displayName>ESales</displayName>
    <internal>false</internal>
    <cqType>RELATION</cqType>
    <cqArchived>true</cqArchived>
    <hidden>false</hidden>
    <columns>
        <column>
            <name>FULLNAME</name>
            <internal>false</internal>
            <length>2000</length>
            <nullable>true</nullable>
            <unique>false</unique>
            <dataType>VARCHAR</dataType>
            <columnType>ATTRIBUTE</columnType>
            <label>label_FULLNAME</label>
            <hidden>false</hidden>
            <editable>true</editable>
            <physicalName>FULLNAME</physicalName>
        </column>
    </columns>

```

```

    <schema>DEV12_SOAINFRA</schema>
    <sourceName>BamDataSource</sourceName>
    <jndiName>jdbc/BeamDataSource</jndiName>
    <factDO>>true</factDO>
  </ns0:externalDataObject>

```

## Delete

Delete removes a data object definition and its contents.

## Request Message

The request message contains the following parameter.

### **dataObjectFullName (xsd:string)**

Full relative path and name of the data object to be deleted. For example:

```
/Samples/Employees
```

## Get

Get retrieves an existing data object definition.

## Request Message

The request message contains the following parameters.

### **dataObjectFullName (xsd:string)**

Full relative path and name of the data object, for example:

```
/Samples/Sales
```

## Response Message

The response message contains the following parameter.

### **xmlPayload (xsd:string)**

An XML description of the data object is returned. The schema used is the same definition as described for the Create and Update operations. You can use this operation to find the ID values of the data object and any columns. Here is the xmlPayload for the Get operation:

```

<ns0:simpleDataObject xmlns:ns0="http://xmlns.oracle.com/bam/bam12">
  <name>Sales</name>
  <displayName>Sales</displayName>
  <internal>false</internal>
  <createdBy>weblogic</createdBy>
  <modifiedBy>weblogic</modifiedBy>
  <cqType>RELATION</cqType>
  <cqArchived>true</cqArchived>
  <slowChangingDimension>false</slowChangingDimension>
  <hidden>false</hidden>
  <columns>
    <column>
      <name>Salesperson</name>
      <displayName>Salesperson</displayName>
      <internal>false</internal>
      <length>2000</length>
    </column>
  </columns>
</ns0:simpleDataObject>

```

```

        <nullable>true</nullable>
        <unique>>false</unique>
        <precision>38</precision>
        <scale>10</scale>
        <dataType>VARCHAR</dataType>
        <columnType>ATTRIBUTE</columnType>
        <hidden>>false</hidden>
        <editable>true</editable>
    </column>
</columns>
<indexes/>
<hierarchies/>
</ns0:simpleDataObject>

```

## Update

Update updates the definition of an existing data object. If a specified column exists in the original definition, the new column definition overwrites the old one. If columns in the existing definition are not specified in the new definition, their definitions are removed. The data object index definition can be updated as well.

## Request Message

The request message contains the following parameter.

### xmlPayload (xsd:string)

The payload for the Update operation is similar to the Create payload. For example:

```

<ns0:simpleDataObject xmlns:ns0="http://xmlns.oracle.com/bam/bam12">
  <name>Sales</name>
  <displayName>Sales</displayName>
  <cqType>RELATION</cqType>
  <cqArchived>true</cqArchived>
  <hidden>>false</hidden>
  <columns>
    <column>
      <name>SalesArea</name>
      <id>1621481da151473caedc1a66c19dc2e6</id>
      <displayName>Salesperson</displayName>
      <length>2000</length>
      <nullable>true</nullable>
      <unique>>false</unique>
      <dataType>VARCHAR</dataType>
      <columnType>ATTRIBUTE</columnType>
      <hidden>>false</hidden>
      <editable>true</editable>
    </column>
  </columns>
</ns0:simpleDataObject>

```

## Using the ManualRuleFire Web Service

The ManualRuleFire web service allows users to launch alerts in the Oracle BAM Server. FireRuleByName is the available operation.

See [ManualRuleFire Operations](#) for details.

For more information about alerts, see [Creating Alerts](#). For a ManualRuleFire example, see [Alert and Web Service Example](#).

## How to Use the ManualRuleFire Web Service

To use the ManualRuleFire web service, you create a web service proxy in your application in Oracle JDeveloper.

The WSDL file for the ManualRuleFire web service is available at the following URL on the system where Oracle BAM web services are installed.

`http://host:port/OracleBAMWS/WebServices/ManualRuleFire?WSDL`



### Note:

The default port for Oracle BAM web services on the Administration Server is 7001. On managed servers the default port number is 9001.

When the web service proxy is created, you see it in the Application Navigator under the Application Sources folder in your project.

## ManualRuleFire Operations

The following operation is supported by ManualRuleFire web service.

- [FireRuleByName](#)

## FireRuleByName

Use this operation to manually launch an alert.

This web service takes a string parameter, which must have the project internal name, followed by a period, followed by the alert internal name, as follows:

```
projectname.alertname
```

The period is the separator between the project name and the alert name. The web service always treats last period in the string as the separator, allowing the project name to contain periods. For example:

```
project.nemo.alert1
```

It follows that you cannot use the ManualRuleFire web service with an alert containing a period in its internal name. Since you cannot change the internal name, you must recreate the alert with an internal name that does not contain any periods.

## Request Message

The request message contains the following parameter.

### xmlPayload (xsd:string)

An example:

```
<FireRuleByName xmlns="http://xmlns.oracle.com/bam">  
  <strRuleName>string</strRuleName>  
</FireRuleByName>
```

## Response Message

### Returns (xsd:string)

```
<FireRuleByNameResponse xmlns="http://xmlns.oracle.com/bam">  
  <FireRuleByNameResult>string</FireRuleByNameResult>  
</FireRuleByNameResponse>
```