Adobe®

Using Business Activity Monitoring Workbench

Adobe® LiveCycle® ES

Update 1

July 2008

Adobe LiveCycle ES Update 1 Using Business Activity Monitoring Workbench

345 Park Avenue, San Jose, CA 95110
Printed in the United States of America

555 Twin Dolphin Drive, Redwood City, California 94065 U.S.A.

DISCLAIMER: Adobe Systems and Cognos Incorporated make no representations or warranties with respect to the contents or use of this publication. Further, Adobe Systems reserves the right to revise this publication and to make changes in its contents at any time, without obligation to notify any person or entity of such revisions or changes.

Contents

1 Getting Started

Adobe® LiveCycle® ES (Enterprise Suite) Update 1 (8.2) Business Activity Monitoring is a collection of client applications that run on Microsoft® Internet Explorer browsers, and which connect to servers that collect, store, and aggregate event and context information.

➤ **In this Chapter:**

Initial Logging In

The first time you log in to BAM Workbench, you are required to set several systems properties before continuing.

All five settings are located in the Systems Settings dialog box, accessible from the Administration Console in BAM Workbench.

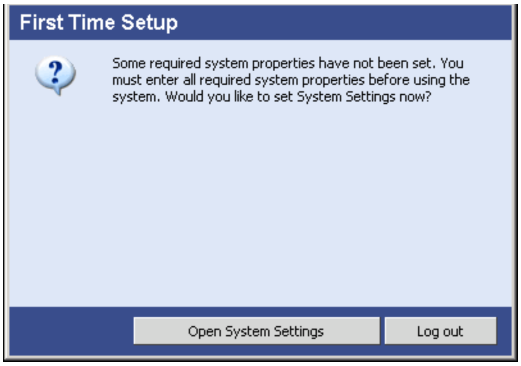
Setting	Location of setting
Recovery Log Directory	Checkpoint Configuration tab pane
Logging Directory	Logging tab pane
SMTP Host	SMTP Configuration tab pane
SMTP From Address	SMTP Configuration tab pane
SMTP User	SMTP Configuration tab pane

➤ To set the required system properties:

1. Log in to BAM Workbench:

   `http://hostname:port/bam/workbench`

   The following alert displays:

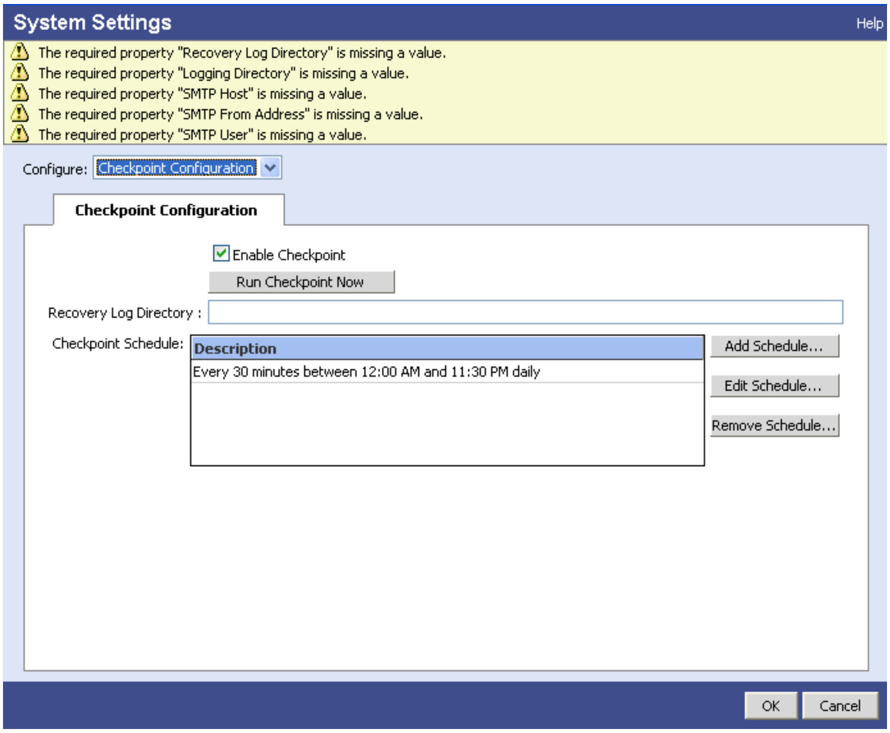


First Time Setup

Some required system properties have not been set. You must enter all required system properties before using the system. Would you like to set System Settings now?

2. Click the **Open System Settings** button.

   The System Settings dialog box opens by default to the Checkpoint Configuration tab pane of the System Settings dialog box. The settings that need to be configured are listed in a panel at the top of the dialog box.



⚠ The required property "Recovery Log Directory" is missing a value.
⚠ The required property "Logging Directory" is missing a value.
⚠ The required property "SMTP Host" is missing a value.
⚠ The required property "SMTP From Address" is missing a value.
⚠ The required property "SMTP User" is missing a value.

Configure: Checkpoint Configuration

☑ Enable Checkpoint

Recovery Log Directory :

Checkpoint Schedule:

Description
Every 30 minutes between 12:00 AM and 11:30 PM daily

3. Complete the settings in the locations indicated in the table at the start of these procedures.

4. Click **OK**.

   Note: You will not be able to exit the System Settings dialog until all five settings have been completed.

About this Document

This document describes Business Activity Monitoring features and how to use them. Specifically, it covers the four areas of Business Activity Monitoring:

- "Modeling Business Data" on page 30
- "Enabling Support for Business Views" on page 55
- "Managing System Administration" on page 92

The following steps describe the order of activities for using Business Activity Monitoring:

Note: Once the system is up and running, and once events and context have been identified, most users can begin with step 7.

1. Your systems administrator must first install, configure, and start the servers.

   To receive events, or to retrieve context data, the servers need to be configured to locate and collect the data. For example, to access a DBMS to retrieve context data, the servers will need to be configured with the access name and password for the database. See the Release Notes for detailed instructions about this task.

2. All administrator and developer access to Business Activity Monitoring is through a Microsoft Internet Explorer browser using one of the Business Activity Monitoring applications at an address provided by your site administrator:

   BAM Workbench is where developers create the views, rules, and alerts that model your business. It is also where administrators manage users and the application servers. Access the workbench with an address similar to this:

   ```
   http://hostname:port/bam
   ```

   BAM Dashboard is where users view the alerts, charts, and tables that present your business metrics. Access the dashboard with an address like this:

   ```
   http://hostname:port/bam/dashboard/
   ```

3. Everyone who uses BAM Workbench or BAM Dashboard needs a *user account*. Users can view their account information by clicking Account Settings in the application.

   System administrators can add and modify user accounts with the Administration Console through the Managing Users. (See on "Managing Users" on page 93.)

4. Events and context data are collected by agents that know how to get the data and make it available to Business Activity Monitoring.

   System administrators create and modify agents with the Administration Console. (See "Viewing the Agents List" on page 95 for details.)

5. Agents insert the collected event and context data into tables that are the source of the business views that present and aggregate the information.

   You can create the event and context tables with the BAM Workbench. (See "Enabling Support for Business Views" on page 55.)

6. Business views are the objects that contain, aggregate, and provide event and context information.

   You can create complex business views with the BAM Workbench. You can also create business views with the Scenario Modeler. (See "Working with Views" on page 64.)

7. At the heart of Business Activity Monitoring is the ability to identify exceptional business events and notify users about the activity.

   You do this by creating scenarios, rules, and alerts in the Scenario Modeler. You can also use the Scenario Modeler to define reportlets that provide the metrics associated with the events. (See "Modeling Business Data" on page 30.)

8. Every user can see the alerts, reportlets, and metrics that have been sent to them with the BAM Dashboard.

   This application presents the details about the alerts and metrics, and allows users to subscribe to new alerts. (See "Alert Messages" in *Using Business Activity Monitoring Dashboard*.)

   Note: Before starting BAM Dashboard, turn off any pop-up blockers in your Web browser.

2 Setting Up Context Data Sources

This chapter shows you how to set up context data sources. Context data supports complements event data by providing additional, meaningful information. For example, an event instance may include a product number but not a product description. Context data sources make this additional information available for inclusion in views. Context data sources have two parts: the agent, which references a JDBC connection or Web service invocation, and the context definition, which configures the input and output for the retrieved data.

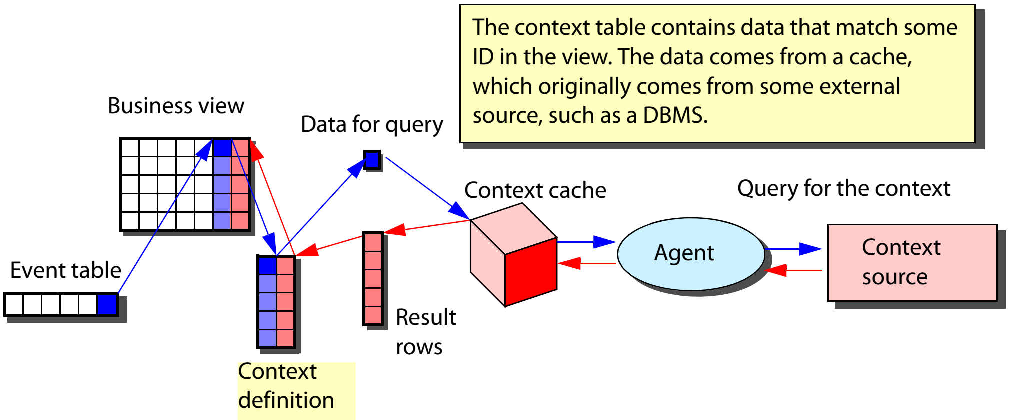
➤ **In this Chapter:**

What is Context Data?

When an event instance is processed in BAM, the data available from the event source may be limited. For example, a view that is processing a purchase order might have received a product identification number along with the event data. By joining information from complementary context data sources, the view can include additional information, such as product description, manufacturer, and so on. In the view definition, a WHERE clause would join the context definition to the event, similar to this:

```
WHERE event.product_id = context.suppliers_of_product_id
```

When the view performs this join, it passes the ID from the event to the context definition. If the matching supplier data are already in the context cache, the definition uses that data and passes it to the business view. If the data are not already in memory, the ID is passed to an agent (either as an SQL query or by value for a stored procedure), which sends data to the DBMS or Web service for processing. The result of the query is then loaded into the context definition, and subsequently included in the business view.


The context table contains data that match some ID in the view. The data comes from a cache, which originally comes from some external source, such as a DBMS.
Business view
Data for query
Context cache
Query for the context
Event table
Agent
Context source
Result rows
Context definition

The context source may be databases accessed through a JDBC, or business applications accessed through Web Services. For details about these type of sources, see “Creating a JDBC Context Data Source Definition” on page 20 or “Creating a Web Service Context Data Source Definition” on page 22.

Procedural Summary

This section briefly summarizes the procedures described in this chapter.

- "Working with Agents for Context Data Sources" on page 14

  The first step in creating a context data source is the creation of an agent in BAM Workbench. While BAM enables you to create many types of agents, for context data sources you will use either a JDBC or Web service agent.

  For JDBC agents, you will require the following information:

  - Database type

    BAM supports DB2, MS SQLServer, MySQL, Oracle and Sybase for context data sources.

  - Database URL or JNDI
  - Valid database user name and password

  For Web service agents, you will require the following information:

  - WSDL address
  - User name and password (optional/as required)

- "Working with Context Data Source Definitions" on page 20

  After creating the agent for the context data source definition, you create the data source definition:

  - For JDBC context data sources, you will specify either a query or a stored procedure. For stored procedures, you have the added task of defining inputs and outputs.
  - For Web service context data sources, you have the option of hiding or showing the returned XPath values.
  - For both JDBC and Web service data sources, you can define a range of data caching parameters, as described in "Working with Data Caching" on page 26.

- "Working with Data Caching" on page 26

  Caching allows you to store the results of context queries in memory, thereby improving performance. You can also set invalidation schedules that purge the cache on a predetermined basis.

Working with Agents for Context Data Sources

The first step in setting up a context data source is the creation of an agent that implements the connection to the data source. While BAM enables you to create several types of agents, for context data sources, you are limited to three types:

- JDBC, which enables you to retrieve context data from a database such as Oracle, MS SQLServer, DB2, Sybase, and others.
- Web Service, which enable you to retrieve context data from a Web service invocation.
- Salesforce, which allows you to retrieve context data from Salesforce tables.

Note: In both cases, the query or invocation is configured in the context definition, as described in "Working with Context Data Source Definitions" on page 20.

This section describes the following procedures:

- Creating a New Agent
- Editing Agents

Creating a New Agent

These steps summarize how to create a new agent for both JDBC and Web service data sources. Before you create an agent, your user configuration must have the necessary Create permission for agents and the connection specifications for the specific agent type.

Note: You must configure a corresponding JNDI definition in the application server configuration. For more information, see the documentation for installation of BAM.

➤ **To create a new agent:**

1. Log in to BAM Workbench.
2. Open the Administration Console tab.
3. Click on **Agents** in the left column.

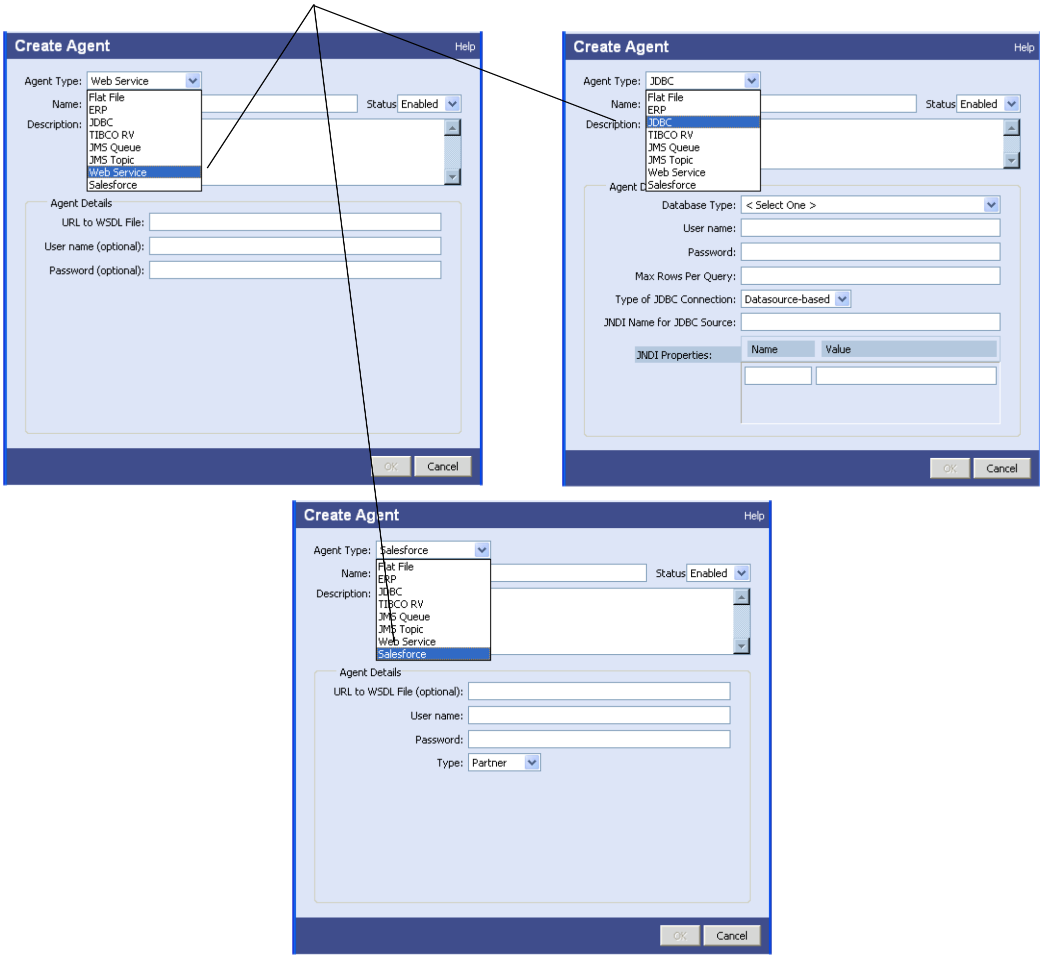
   The page redisplays to show the Agents tab.
4. Click **New Agent**.

   The Create Agent dialog box displays.

5. From the Agent Type list, select JDBC, Web Service, or Salesforce as desired.

   After selection, the dialog box redisplays to show settings appropriate to the agent type.

Create Agent settings for JDBC and Web Service



Create Agent

Agent Type: Web Service

Name:

Status Enabled

Description:

Agent Details

URL to WSDL File:

User name (optional):

Password (optional):

Create Agent

Agent Type: JDBC

Name:

Status Enabled

Description:

Agent D

Database Type: < Select One >

User name:

Password:

Max Rows Per Query:

Type of JDBC Connection: Datasource-based

JNDI Name for JDBC Source:

JNDI Properties:

Name	Value

Create Agent

Agent Type: Salesforce

Name:

Status Enabled

Description:

Agent Details

URL to WSDL File (optional):

User name:

Password:

Type: Partner

6. Complete the settings as appropriate.

   Attribute description for each are described in the following tables:

   - Table 1, “JDBC Agent Attributes,” on page 16
   - Table 2, “Web Service Agent Attributes,” on page 18

- Table 3, “Salesforce Agent Attributes,” on page 18

Table 1: JDBC Agent Attributes

Attribute	Description
Name	Identifies the agent. This name must be unique among agents. See “Object Namespace” on page 248 for details.
Description	Optional description that may contain any text characters.
Status	Whether or not the agent is enabled or disabled.
Database type	Identifies the DBMS vendor as MySQL, PostgreSQL, Oracle, Sybase, SQL Server, or DB2.
User name	User name to use to connect to the DBMS. Must have *query* access rights.
Password	Password for the User name. If you omit this option, the agent uses the password specified in the JDBC Source configuration definition in the application server.
Max rows per query	Maximum count of rows to return as the query result. Useful to keep users from returning exceptionally large results that impact the DBMS.
Type of JDBC connection	How to connect to the JDBC in the application server. • Datasource-based: Connects to a JDBC database pool through a JNDI connection. Do not use for IBM Websphere. • URL-based: Connects to a JDBC source via a URL.
JNDI name for JDBC Source (JNDI only)	Name of the data source to use as a connection to the database. The name is in JNDI form, such as “com.celequest.products.ProductSource”. A source gets its connection from a pool of connections maintained by the application server. That server keeps the connections open to reduce delays when establishing a connection. **Note:** The connection pool must be configured as a non-transactional pool; non-TxT on WebLogic.

Table 1: JDBC Agent Attributes (Continued)

Attribute	Description
JNDI properties (JNDI only)	Optional and additional Java naming and directory interface (JNDI) properties necessary to make or maintain the agent to the JDBC source. These name/value pairs allow you to specify JDBC properties. The names are either one of the short cuts listed below, or a JNDI recognized property. The agent recognizes the following names as short cuts to JNDI properties: • factory maps to INITIAL_CONTEXT_FACTORY. • provider maps to PROVIDER_URL. • security_credentials maps to SECURITY_CREDENTIALS. • security_principal maps to SECURITY_PRINCIPAL.
JDBC URL (URL only)	URL that maps to the JDBC connection configured in the application server running Business Activity Monitoring. Please refer to your JDBC driver documentation for the proper format of this URL.
JDBC driver class (URL only)	JDBC driver to use. This driver must reside in the classpath of the application server running Business Activity Monitoring. Please refer to your JDBC driver documentation for the proper classname of the JDBC driver.

Table 2: Web Service Agent Attributes

Attribute	Description
Name	Identifies the agent. This name must be unique among agents. See "Object Namespace" on page 248 for details.
Description	Optional description that may contain any text characters.
Status	Whether or not the agent is enabled, or disabled.
URL	HTTP location of the Web service definition language file (WSDL) that describes the service, the data it provides, and how to exchange data with the service. Note that returned data must be a SOAP doc-style message. RPC binding is not supported.
User name	(optional) User name to use when connecting to the service. This parameter is passed to the server when the server requires a user name.
Password	(optional) User password to use when connecting to the service. This parameter is passed to the server when the server requires a password.

Table 3: Salesforce Agent Attributes

Attribute	Description
Name	Identifies the agent. This name must be unique among agents. See "Object Namespace" on page 248 for details.
Description	Optional description that may contain any text characters.
Status	Whether or not the agent is enabled, or disabled.
URL	HTTP location of the Web service definition language file (WSDL) that describes the service, the data it provides, and how to exchange data with the service. Note that returned data must be a SOAP doc-style message. RPC binding is not supported.
User name	User name to use when connecting to the service. This parameter is passed to the server when the server requires a user name.
Password	User password to use when connecting to the service. This parameter is passed to the server when the server requires a password.

7. Set the Status value to Enabled.

8. Click OK.

   The Create dialog box closes and the new agent is now listed in the Agents tab. The agent is immediately available to receive context data.

Editing Agents

These steps summarize how to modify an existing an agent. For details about agents attributes, see the description of the specific source.

Note: Before you create an agent, you must have Read and Write permission for the agent. For more information, see the "Permissions" in the *Business Activity Monitoring Server Reference*.

➤ **To edit or modify an existing agent definition:**

1. Log in to BAM Workbench.
2. Open the Administration Console tab.
3. Click on **Agents** in the left column.

   The page redisplays to show the Agents tab.
4. Double-click the agent to alter.

   The Edit Agent dialog box displays.
5. Modify the attributes for the specific source type, as desired.
6. Click **OK**.

   The Create dialog box closes and the modified agent is now listed in the Agents tab.

Working with Context Data Source Definitions

This section describes the following procedures related to context definitions:

- Creating a JDBC Context Data Source Definition
- Creating a Web Service Context Data Source Definition
- Editing Context Data Source Definitions
- A Note on Context Column Limitations
- Mapping JDBC data types

Note: You cannot create a context data source definition without first creating an agent, as described in "Working with Agents for Context Data Sources" on page 14.

Creating a JDBC Context Data Source Definition

This section shows how to create a new JDBC context data source. A JDBC context data source definition can take the form of either an SQL query or the invocation of a stored procedure.

Both forms are described below.

➤ **To create a JDBC context data source definition:**

1. Log in to BAM Workbench.

2. Open the Application Workbench tab.

3. Click on **Contexts** in the left column.

   The page redisplays to show the Contexts tab.

4. Click **New Context**.

   The New Context dialog box displays.

5. Select JDBC and click **Continue**.

   The New Context dialog box redisplays to show the connection options.

6. From the JDBC Connection dropdown list, select the desired source.

   This list dynamically shows the JDBC agents you previously configured, as described in "Creating a JDBC Context Data Source Definition" on page 20.

7. To create a query-based source:

   - Select the Query radio button.
   - Enter a SQL query string in the text field. For example, `SELECT * FROM PRODUCTS`.
   - Click **Continue**.

     If the Query is not valid, an error message will appear. Otherwise, the Configure Context - JDBC Query dialog box displays. The results of the query are displayed in the lower portion of the dialog box in the Field Information tab pane.
   - Enter a name for the context in the Name field.
   - In the Status field, select Enabled.

- Select index for those columns you will be using in join conditions. (Index have no affect if prefetch is not enabled.)

  It is critical that you select the right index in order to have good performance when prefetch is enabled for data caching.

- Click the Data Caching tab if you wish to modify the caching parameters.

  The default is ten result sets.

- When finished, click the Save Context button.

  The dialog box closes and the BAM Workbench now displays tabs for the new context's columns (as read from the query), dependencies, and join relationships.

8. To create a stored procedure-based source:

   - Select the Stored Procedure radio button.
   - Click **Continue**.

     The Configure Context - Stored Procedure dialog box displays.

   - Enter a name for the context in the Name field.
   - In the Status field, select Enabled.
   - Enter the name of the stored procedure in the Procedure Name field.

     This value must match the name of a stored procedure in the database referenced by the agent specified as the database connection, as indicated in step 6 above.

   - In the Fields tab pane, create inputs and outputs, as described in Table 4: JDBC Stored Procedure Attributes.
   - Click the Data Caching tab if you wish to modify the caching parameters.

     The default is ten result sets.

   - When finished, click the Save Context button.

     The dialog box closes, and the BAM Workbench now displays tabs for the new context's columns (as read from the query), dependencies, and join relationships.

   The source will immediately be ready to receive context data.

Table 4: JDBC Stored Procedure Attributes

Attribute	Description
Outputs	Columns in the context definition, their data types, and optional formatting. The procedure returns a result set whose values map to the columns in the order they appear in this list. The data type identifies the type of the column in the definition and will automatically be converted from the JDBC type as defined in "Mapping JDBC data types" on page 25.
Inputs	Parameters passed to the stored procedure and their data types. Inputs are passed as arguments to the procedure in the order they appear in the list. The data type identifies Business Activity Monitoring data type of the value being passed to the procedure. See "Mapping JDBC data types" on page 25 for details.

Creating a Web Service Context Data Source Definition

This section describes how to create a Web service context source.

➤ **To create a Web Service context data source definition:**

1. Log in to BAM Workbench.

2. Open the Application Workbench tab.

3. Click on **Contexts** in the left column.

   The page redisplays to show the Contexts tab.

4. Click **New Context**.

   The New Context dialog box displays.

5. Select Web service and click **Continue**.

   The New Web Services Context dialog box displays.

6. Select the agent (as configured in "Working with Agents for Context Data Sources" on page 14) that represents the Web service you wish to use as the context source.

7. Click **Continue**.

   The New Web Services Context dialog box redisplays to show the methods/operations available in the Web service.

8. Select the desired method and click **Continue**.

   The Configure Context - Web Services dialog box displays. The input and outputs, as read from the WSDL file referenced by the Web service agent configuration, are displayed in the Field Information tab pane at the bottom of the dialog box.

   Note: Depending on its relevance, you can show or hide the contents as desired with the Hide/Show button.

9. Enter a name for the context in the Name field.

10. In the Status field, select Enabled.

11. Click the Data Caching tab if you wish to modify the caching parameters.

    The default is ten result sets.

12. When finished, click the Save Context button.

    The dialog box closes and the BAM Workbench now displays tabs for the new context's columns (as read from the query), dependencies, and join relationships.

    The source will immediately be ready to receive context data.

Creating a Salesforce Context Data Source Definition

When you import Salesforce tables with the Import SalesForce Object Wizard (see "Salesforce" on page 305), the wizard creates a context object from the imported table, appending _context to the name of the table to create the context name. For example, a table named Account imported for a context is named Account_context.

Use the following procedure to create a Salesforce context source, if you want to create additional contexts from your imported Salesforce tables.

➤ **To create a Salesforce context data source definition:**

1. Log in to BAM Workbench.
2. Open the Application Workbench tab.
3. Click on **Contexts** in the left column.

   The page redisplays to show the Contexts tab.
4. Click **New Context**.

   The New Context dialog box displays.
5. Select Salesforce and click **Continue**.

   The New Context dialog box redisplays to show the connection options.
6. From the Salesforce Connection dropdown list, select the desired source.

   This list dynamically shows the Salesforce agents you previously configured, as described in "Creating a Salesforce Context Data Source Definition" on page 22.
7. Enter a SQL query string in the Salesforce Query text field. For example, `SELECT * FROM ACCOUNT`.
8. Click **Continue**.

   If the Query is not valid, an error message appears. Otherwise, the Configure Context - Salesforce Query dialog box displays. The results of the query are displayed in the lower portion of the dialog box in the Field Information tab pane.

   - Enter a name for the context in the Name field.
   - In the Status field, select Enabled.
   - From the Salesforce Agent drop down select a Salesforce agent.
   - Click the Data Caching tab if you wish to modify the caching parameters.

     The default is ten result sets.
   - When finished, click the Save Context button.

     The dialog box closes and the BAM Workbench now displays tabs for the new context's columns (as read from the query), dependencies, and join relationships.

   The source will immediately be ready to receive context data.

Editing Context Data Source Definitions

These steps summarize how to modify an existing context data source definition.

➤ **To edit or modify an existing context data source definition:**

1. Log in to BAM Workbench.
2. Open the Application Workbench tab.

3. Click on **Contexts** in the left column.

   The page redisplays to show the Contexts tab.

4. Double-click the context you wish to alter.

   The details page of the context definition displays.

5. Click the **Edit This Context** button in the upper right portion of the window.

   The Configure Context dialog box displays.

6. Modify the definition settings as desired.

7. Click **OK**.

   The Create dialog box closes and the modified agent is now listed in the Agents tab.

A Note on Context Column Limitations

Context can be retrieved with *no limitations from a JDBC query source*. However, the following limitations apply when retrieving context from a JDBC stored procedure source or a Web service source:

- When the context column is referenced as part of a query, somewhere in the WHERE or FROM clause it *must* appear in an equality expression and then only as an atomic predicate (no other operators on the same side of the equal sign). For example, this is permitted:

  ```
  WHERE context_column = 10*event_column
  ```

  But the following is *not permitted* because the left-side predicate, which contains the context column, is an expression that includes an operator (/):

  ```
  WHERE context_column/10 = event_column
  ```

- The required atomic reference *may not* appear in a disjunct (OR) expression. This fails:

  ```
  WHERE (context_column = event_column OR A > B)
  ```

  However, it *may* appear in a conjunct (AND) expression:

  ```
  WHERE (context_column = event_column AND A > B)
  ```

- Once there is at least one equality reference in the query, you may use the column in any other way. For example, these two are permitted:

  ```
  WHERE (context_column = event_column AND
         context_column/10 = other_event_column)
  ```

  ```
  WHERE ((context_column = event_column AND
          context_column >= other_event_colum) OR
         (A > B))
  ```

  But the following *fails* because there is no equality reference in the query:

  ```
    WHERE context_column >= event_column
  ```

- The required equality expression *may not* reference another context column in the same table. For example:

  ```
    t1.context_column = t1.other_context_column
  ```

- However, the equality expression *may* reference a context column in another table, for example:

  ```
    t1.context_column = t2.other_context_column
  ```

Sybase Limitations

When making a query to a Sybase database, be aware of the following limitations:

- All names, including tables and columns, are case-sensitive.
- All queries must be in the form `SELECT * FROM table` only; you cannot include any SELECT clauses. To filter the results, load them into a business view, then filter that view.

Mapping JDBC data types

The data types of the DBMS columns are displayed as JDBC data types and map to Business Activity Monitoring Data Types as follows.

JDBC data type	Character	Integer	Double	Decimal	Timestamp	Boolean
CHAR	yes	—	—	—	—	as literal
VARCHAR	yes	—	—	—	—	as literal
LONGVARCHAR	yes	—	—	—	—	as literal
NUMERIC	—	yes	yes	yes	—	yes
DECIMAL	—	yes	yes	yes	—	yes
BIT	—	yes	yes	yes	—	yes
TINYINT	—	yes	yes	yes	—	yes
SMALLINT	—	yes	yes	yes	—	yes
INTEGER	—	yes	yes	yes	—	yes
BIGINT	—	yes	yes	yes	—	yes
REAL	—	yes	yes	yes	—	yes
FLOAT	—	yes	yes	yes	—	yes
DOUBLE	—	yes	yes	yes	—	yes
BINARY	—	—	—	—	—	—
VARBINARY	—	—	—	—	—	—
DATE	—	—	—	—	yes	—
TIME	—	—	—	—	yes	—
TIMESTAMP	—	—	—	—	yes	—

Note: Do not use Business Activity Monitoring Boolean data type in a WHERE predicate passed to the JDBC source. Boolean values may be included in the Select list.

Working with Data Caching

Caching allows you to store the results of context queries in memory. Subsequent requests for the same information are then retrieved from memory instead of impacting the DBMS with a redundant query. When caching is active and a view requests context, it searches the cache first.

This section describes:

- How Caching is Determined
- Setting Caching Parameters
- Setting Invalidation Schedules

How Caching is Determined

Business Activity Monitoring allows you to enable two different methods of data caching (see "Setting Caching Parameters" on page 27 for information on how to enable the caches):

- On-demand
- Prefetch

Using On-Demand Caching

On-demand caching occurs when you have selected "Cache data for this context" on the data caching tab and have **not** selected "Enable prefetch" (see "Setting Caching Parameters" on page 27). With on-demand caching, the recent query cache is created that maintains results on a least recently used (LRU) basis. This cache keeps track of when each result set was last requested. When the cache is full, it keeps the most recently accessed rows and discards those that have not been accessed in the longest period of time. If data is requested that is not in the recent query cache, the data is retrieved from the external data source.

> **Note:** Rows containing frequently requested data will remain in the recently query cache the longest to reduce impact on the database. However, if details about the information can change often, define an invalidation schedule to account for the changes and thereby invalidate the cache. (See "Setting Invalidation Schedules" on page 28 for information about invalidation schedules.)

When a scheduled invalidation occurs, the recent query cache is cleared and updated on subsequent queries. If context data is not rapidly changing, it is best to invalidate the recent query cache less often. For example, if the context is fairly static, you might want to invalidate the cache weekly or monthly. However, if the context database is updated nightly, you might want to invalidate the recent query cache nightly as well to ensure the latest data.

On demand caching can deliver better performance than if no caching is used. However, you might be able to improve performance further by using prefetch caching.

Using Prefetch Caching

When you enabling the prefetch cache (see "Setting Caching Parameters" on page 27), the entire external context table is cached into memory from the external data source and becomes a replacement for the external query source. The recent query cache functions as described in the previous section, "Using On-Demand Caching"; however, when data is requested that is not in the recent query cache, the data is retrieved from the prefetch cache instead of the external data source. When an invalidation occurs according to your specified invalidation schedule, the recent query cache is cleared and a query is issued to the external data source to update the in-memory copy of the context table. The recent query cache is updated from the prefetch cache on subsequent queries.

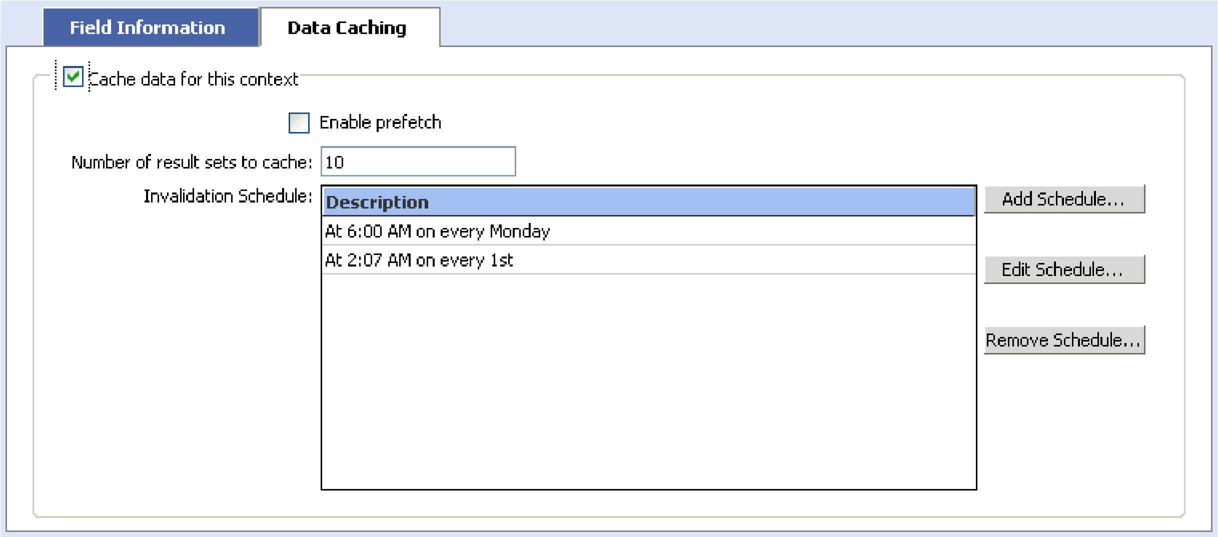
When enabling the prefetch data cache, you should consider the following:

- More memory is used because the information from the external data source is stored in memory, and an index is built for each column.
- The prefetch cache is only updated at the scheduled invalidation time. If data is not in the prefetch cache, no data is returned. Updates to the external data source are not reflected in the prefetch cache until after the next scheduled invalidation.

The prefetch cache retries up to 5 times to refresh on the invalidation schedule if an exception occurs. If it cannot refresh from the external data source, the recent query cache is then used until the next scheduled refresh.

Setting Caching Parameters

The Data Caching tab pane is located in the Configure Context dialog box for both JDBC and Web service data context sources, as described in "Working with Context Data Source Definitions" on page 20.



☑ Cache data for this context

□ Enable prefetch

Number of result sets to cache: 10

Invalidation Schedule:

Description
At 6:00 AM on every Monday
At 2:07 AM on every 1st

The Data Caching tab pane enables you to set the following parameters:

- **Cache data for this context** check box

  Either cache or do not cache context query result sets in the recent query cache. This box must be checked to enable the remaining parameters to be set. You must also add at least one invalidation schedule when selecting this parameter.

  Note: If not selected, the default is 10 result sets.

- **Enable prefetch** check box

  Either enable or disable context prefetch. Selecting this parameter creates the prefetch cache.

  Note: The prefetch cache is only supported for JDBC and Salesforce web services.

- Number of results to cache

  Specifies the count of result sets to cache in memory. Each set of results may contain one or more rows of context related to the event. For example, if 3 queries are made and each result set contains 5 rows, 15 rows would be stored in the cache.

- **Add Schedule** button

  This button opens a dialog box that enables you to set an invalidation schedule that specifies when to invalidate the caches and discard all information currently in the caches. For more information, see "Setting Invalidation Schedules" on page 28.

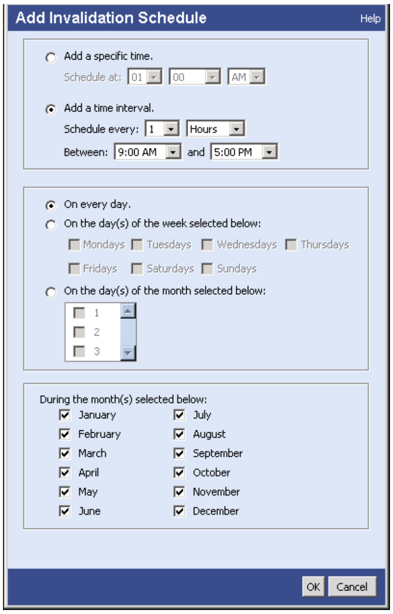
Setting Invalidation Schedules

You can create multiple invalidation schedules that determine when data can be purged from the cache.

➤ To create an invalidation schedule:

1. From the Configure Context dialog box, click the Data Caching tab.
2. In the resulting tab pane, click the **Add Schedule** button.

   The Add Invalidation Schedule dialog box displays.This dialog enables you to set a variety of scheduling options, including day, time, day of the week, and so on.



Add Invalidation Schedule

○ Add a specific time.
Schedule at: 01 00 AM

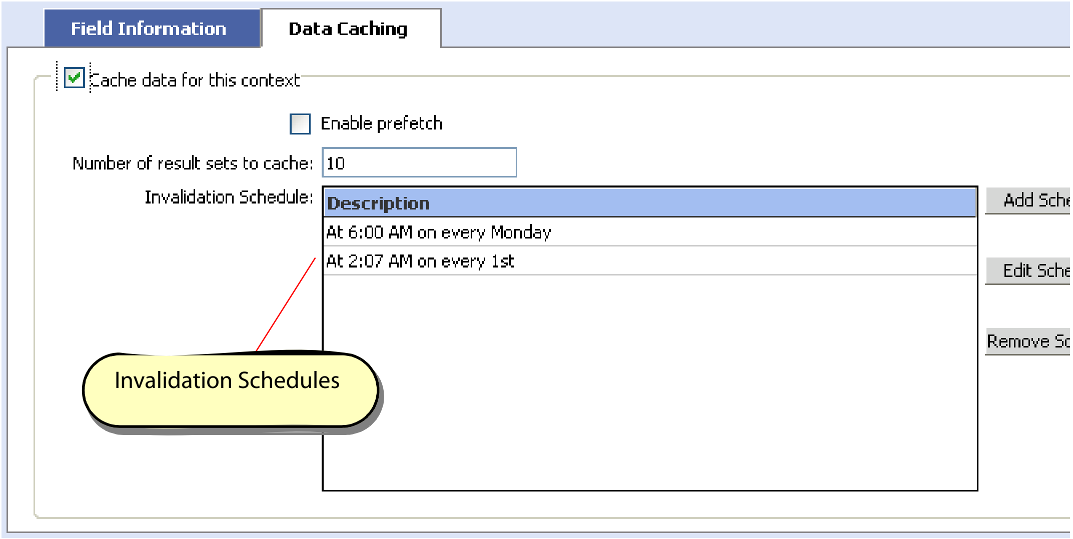
● Add a time interval.
Schedule every: 1 Hours
Between: 9:00 AM and 5:00 PM

● On every day.
○ On the day(s) of the week selected below:
□ Mondays □ Tuesdays □ Wednesdays □ Thursdays
□ Fridays □ Saturdays □ Sundays
○ On the day(s) of the month selected below:
□ 1
□ 2
□ 3

During the month(s) selected below:
☑ January ☑ July
☑ February ☑ August
☑ March ☑ September
☑ April ☑ October
☑ May ☑ November
☑ June ☑ December

3. Set the desired parameters and click **OK**.

   The Add Invalidation Schedule dialog box closes and the new schedule appears in the Data Caching tab pane.



☑ Cache data for this context

□ Enable prefetch

Number of result sets to cache: 10

Invalidation Schedule:

Description
At 6:00 AM on every Monday
At 2:07 AM on every 1st

Invalidation Schedules

Note: You can add multiple schedules for each context.

➤ To edit a validation schedule:

1. From the Configure Context dialog box, click the Data Caching tab.
2. In the resulting tab pane, select the schedule you wish to modify from the list in the Invalidation Schedule Description field.
3. Click the **Edit Schedule** button.

   The Add Invalidation Schedule dialog box displays.
4. Modify the schedule parameter as desired.
5. Click OK to save the changes and close the dialog box.

➤ To remove a validation schedule:

1. From the Configure Context dialog box, click the Data Caching tab.
2. In the resulting tab pane, select the schedule you wish to remove from the list in the Invalidation Schedule Description field.
3. Click the **Remove Schedule** button.

   The listing no longer appears.

3 Modeling Business Data

Business data modeling is a technique for describing the events, context, and rules that depict how your business functions. Modeling in Business Activity Monitoring is done by combining event streams and context sources into *business views*: models that provide a picture of a business activity. As new events enter the system, the views immediately update to reflect the current details about the activity.

In addition to the views, another part of modeling is the ability to create and test *scenarios*. Scenarios allow you to test for expected or possible outcomes, and to identify exceptional business conditions. Each scenario contains rules that identify specific possible conditions, and alerts and reportlets to send to key personnel when the condition is found to exist.

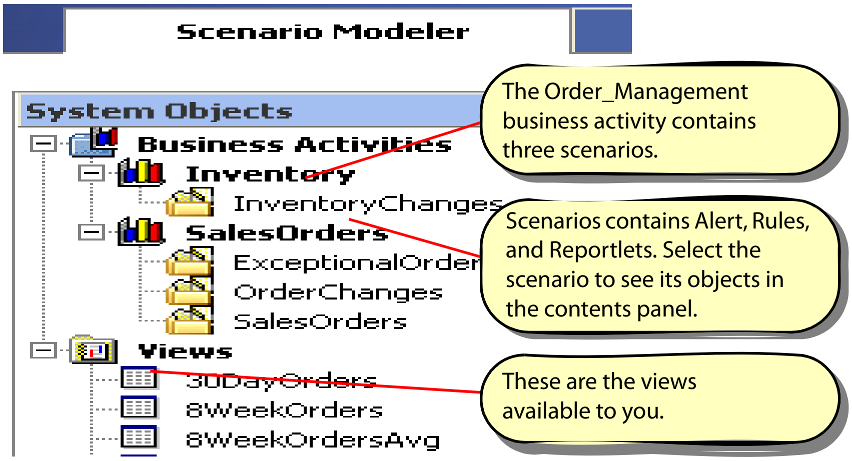
In BAM Workbench, all data modeling is performed with the Scenario Modeler.

➤ **In this Chapter:**

Working with the Scenario Modeler

The Scenario Modeler is where you define scenarios and their rules, alerts, and reportlets, and create views for modeling your business data.

On the left side of the Scenario Modeler is a hierarchical tree that displays all of the objects that you have defined, or which were defined by others and which you are allowed to see. Below the business activity objects is a list of the views that you can use to create your models.



The Order_Management business activity contains three scenarios.

Scenarios contains Alert, Rules, and Reportlets. Select the scenario to see its objects in the contents panel.

These are the views available to you.

When you select an object in the tree, the panel on the right side of the Scenario Modeler displays details about the object, and any objects that folder contains.

The following topics describe each of the objects in greater detail.

- "Working with Business Views" on page 33
- "Working with Business Activities and Scenarios" on page 40
- "Working with Rules and Alerts" on page 41
- "Working with Reportlets" on page 54

Viewing Object Status and Validity

Every object has a status that indicates its ability to receive and process new data:

- Running (or "enabled")

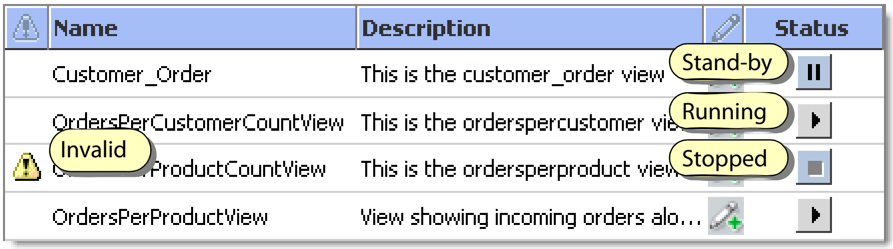
  The object is accepting new data and is processing them. All objects are created in this state.

- Stand-by (or "disabled")

  The object is not accepting new data. Disabling an object does not affect the definition or existence of that object; rather, it just keeps new data from flowing into the object and to all objects that rely on the target object. All dependant objects, however, are stopped.

- Stopped (or "disabled dependant")

  The object is not accepting new data and cannot be re-enabled until an object it depends on is first re-enabled. See "Viewing Dependencies" on page 57 for information about determining which objects are derived from or depend on a particular object.



⚠	Name	Description	✎	Status
	Customer_Order	This is the customer_order view		⏸
	OrdersPerCustomerCountView	This is the orderspercustomer vi		▶
⚠	OrdersPerProductCountView	This is the ordersperproduct view		■
	OrdersPerProductView	View showing incoming orders alo…	✎	▶

Note: See "Object Status" in *Server Reference* for more detailed information about object states and the effects of changing a state.

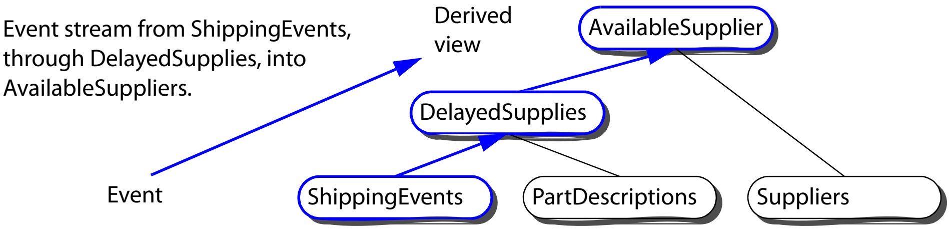
Furthermore, if an object has a reference to another object which cannot be satisfied, that object is *invalid*. A reference can be invalid because an object does not exist or because some attribute of an object does not match the requirements of the dependent (such as a data type mismatch), not because the dependent is disabled. Note that all objects that depend on an invalid object are also invalid.

Working with Business Views

Business views are data models that provide a real-time picture of a business activity. As changes and transactions occur in your business, they generate events that are sent to Business Activity Monitoring. Each new event drives an immediate update of the views — the business models — which in turn provides a real-time, updated view of the business metrics.

New events entering the system become part of a stream that flows from the event source to a business view, possibly passing through other business views along the way. As such, every business view depends on an event or another business view, and every view optionally joins the event information with context information used to provide a meaningful picture of the activity.

This illustration shows the view *AvailableSuppliers* as part of an event stream starting with *ShippingEvents* and passing through *DelayedSupplies*. *PartDescription* and *Supplies* are each joined context sources.



In the Scenario Modeler, you can create business views using events or other business views as the source of the event stream that drives the new view. You can also create views that join context to the event stream, *provided that a join relationship has been defined* between the source event or view and the context table. Use the BAM Workbench to create join relationships (see “Viewing Join Relationship Definitions” on page 58 for details).

A key power of business views is the ability to aggregate event and context information; that is, to detect, analyze, and combine the information into meaningful business metrics. For example, views can perform simple aggregations on the event stream, such as keeping a running total of all the events received, in whole or over a period of time.

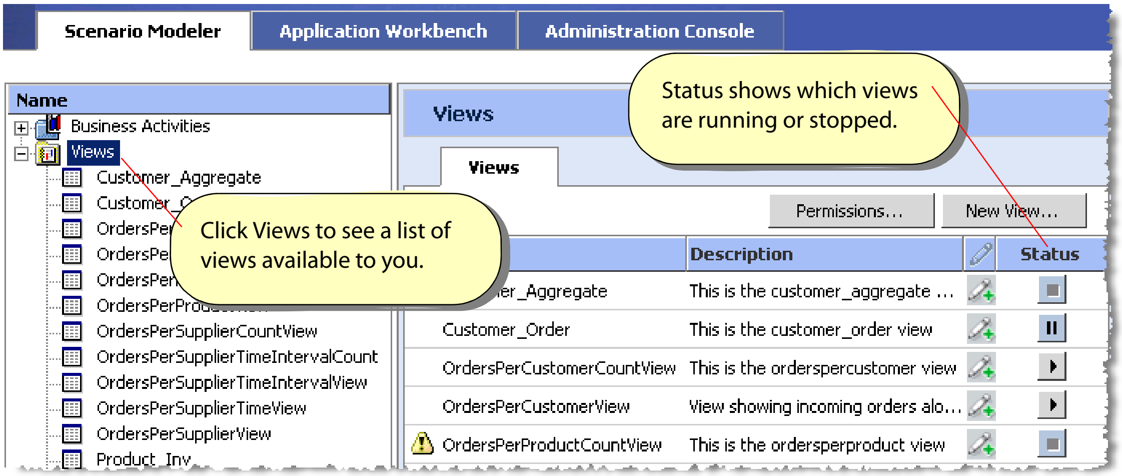
By combining aggregate data with other context, you can create more meaningful metrics for various scenarios. For example, a view might track the performance of sales representatives to identify which are on target to meet their individual quotas for the fiscal quarter. Each sale is recorded as an event, included in a total sales calculation, and compared against quota information retrieved as context. Those current performance results can further be combined to compare the performance of different sales regions. Another view might collect the performance of all sales regions and compare those metrics with current manufacturing projections to identify possible over-production situations.

Aggregation is performed by formulas that display their results in fields in the view. See “Working with View Fields” on page 37 for information about creating formulas in fields.

The rest of this section discusses how to create and work with views in the Scenario Modeler. For a discussion of creating more complex views, see “Working with Views” on page 64.

Views in the Scenario Modeler

To see and work with the views that you have access to, or to create new views, open the Views list in the Scenario Modeler by opening Object Browser and then opening Views in the object tree.



Status shows which views are running or stopped.

Click Views to see a list of views available to you.

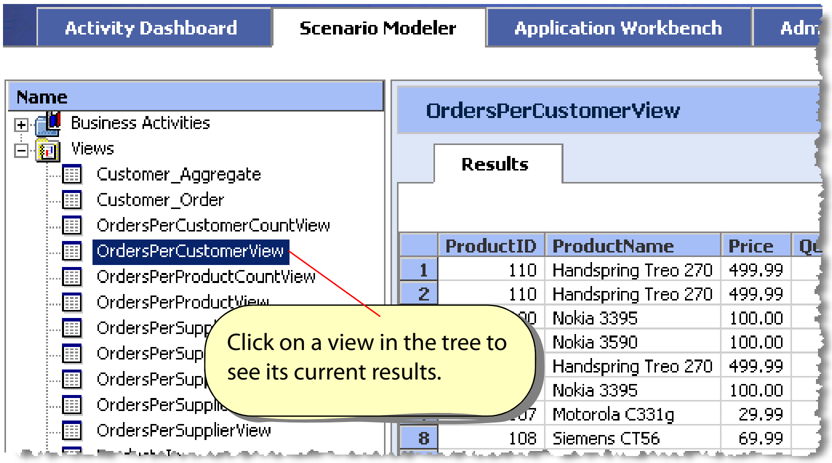
	Description		Status
er_Aggregate	This is the customer_aggregate ...		■
Customer_Order	This is the customer_order view		⏸
OrdersPerCustomerCountView	This is the orderspercustomer view		▶
OrdersPerCustomerView	View showing incoming orders alo...		▶
⚠ OrdersPerProductCountView	This is the ordersperproduct view		■

The Views list identifies each view available to you, shows the view's valid/invalid state, and shows its running or stopped status. When a view is stopped, it is not receiving events, nor is it passing them to any views, rules, alerts, or reportlets that depend on that view. See "Viewing Object Status and Validity" on page 32 for information about stopping and starting views, and for details about invalid views.

Note: To create a view in the Scenario Modeler, see "Creating Views" on page 34.

Viewing Results

To see the current values in the view (if any), click the view in the object tree.



Click on a view in the tree to see its current results.

	ProductID	ProductName	Price	Qu
1	110	Handspring Treo 270	499.99	
2	110	Handspring Treo 270	499.99	
	0	Nokia 3395	100.00	
		Nokia 3590	100.00	
		Handspring Treo 270	499.99	
		Nokia 3395	100.00	
	07	Motorola C331g	29.99	
8	108	Siemens CT56	69.99	

Creating Views

Before you can create a view in the Scenario Modeler, you will need either an existing event table or business view on which to base your new view. You can create event tables and base views in the BAM Workbench. See "Enabling Support for Business Views" on page 55 for details.

One of the powerful features of a business view is the ability to join event information with context. To create such a view, you will first need an existing context table, and at least one *join relationship* to that table. See "“Understanding Relationships” on page 36 for details. If the event or base view does not have a join relationship, you can still create a view in the Scenario Modeler; however, it cannot be joined with context information.

➤ To create a view in the Scenario Modeler:

1. Select **Views** in the object tree and click **New View** above the Views list.

2. Select the base event or view that will supply events to your new view.

   If this base event or view is related to any context tables, you may choose to include information from those tables as well. This allows you to create a view that joins context to the event information. Note that the relationships must first be defined in the BAM Workbench. See “Understanding Relationships” on page 36 for an explanation of this feature.

   If any join relationships are defined for this base event or view, a dialog lists the available relationships.

   - Select the related views and context tables to join with the event information in your view.
   - When there are multiple possible relationships, a dialog prompts you to pick the one that makes the join you desire.

3. Define the view attributes in the View Editor.

Details about the editor are discussed in following sections:

- “Understanding Relationships” on page 36
- “Creating Views” on page 34
- “Affecting the Results” on page 38

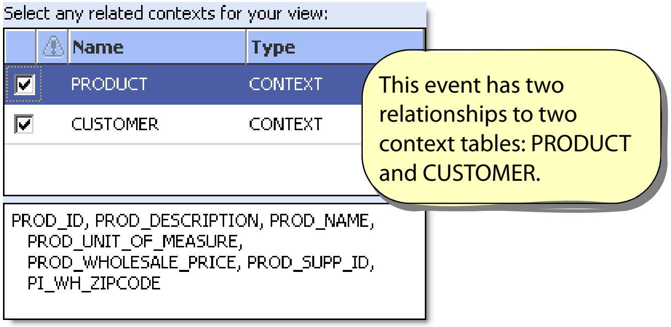
Save the view in a running state (*enabled*) and it will immediately be ready to receive events. For a discussion about the view states, see “Viewing Object Status and Validity” on page 32.

Understanding Relationships

Join relationships define the connection between tables or views and context tables. In the Scenario Modeler you use join relationships to create views that join events and context.

> **Note:** Define join relationships in the BAM Workbench. See "Viewing Join Relationship Definitions" on page 58 for details.

When creating a view in the Scenario Modeler and relationships are available for the base event or view, a dialog prompts you to pick the tables to join.



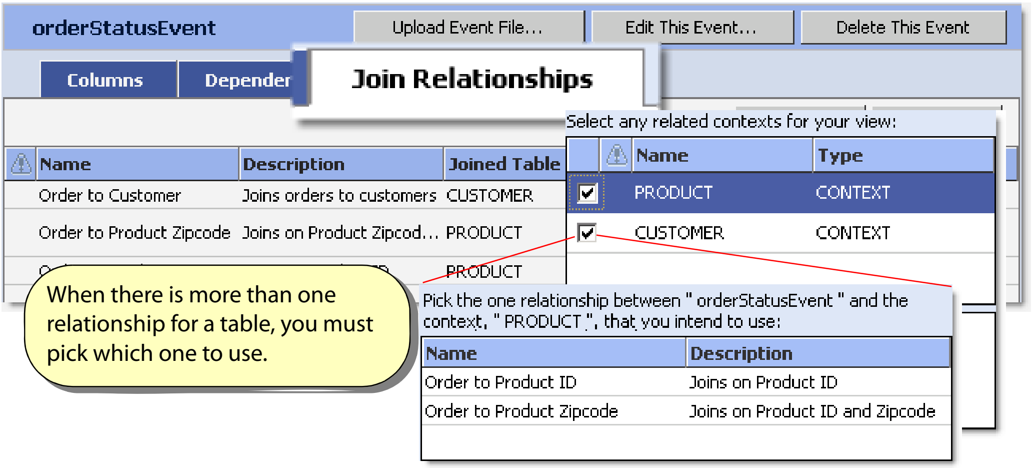
Select any related contexts for your view:

	Name	Type
☑	PRODUCT	CONTEXT
☑	CUSTOMER	CONTEXT

PROD_ID, PROD_DESCRIPTION, PROD_NAME, PROD_UNIT_OF_MEASURE, PROD_WHOLESALE_PRICE, PROD_SUPP_ID, PI_WH_ZIPCODE

This event has two relationships to two context tables: PRODUCT and CUSTOMER.

When there are multiple relationships to the selected table, a subsequent dialog prompts you to choose the one you want to use. For example, the illustration below shows the relationships defined in the BAM Workbench, then the two dialogs in the Scenario Modeler that prompt for the relationships to use:



Name	Description	Joined Table
Order to Customer	Joins orders to customers	CUSTOMER
Order to Product Zipcode	Joins on Product Zipcod...	PRODUCT
		PRODUCT

Select any related contexts for your view:

	Name	Type
☑	PRODUCT	CONTEXT
☑	CUSTOMER	CONTEXT

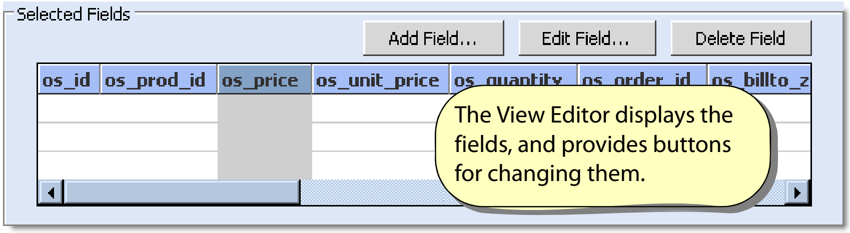
Pick the one relationship between " orderStatusEvent " and the context, " PRODUCT ", that you intend to use:

Name	Description
Order to Product ID	Joins on Product ID
Order to Product Zipcode	Joins on Product ID and Zipcode

When there is more than one relationship for a table, you must pick which one to use.

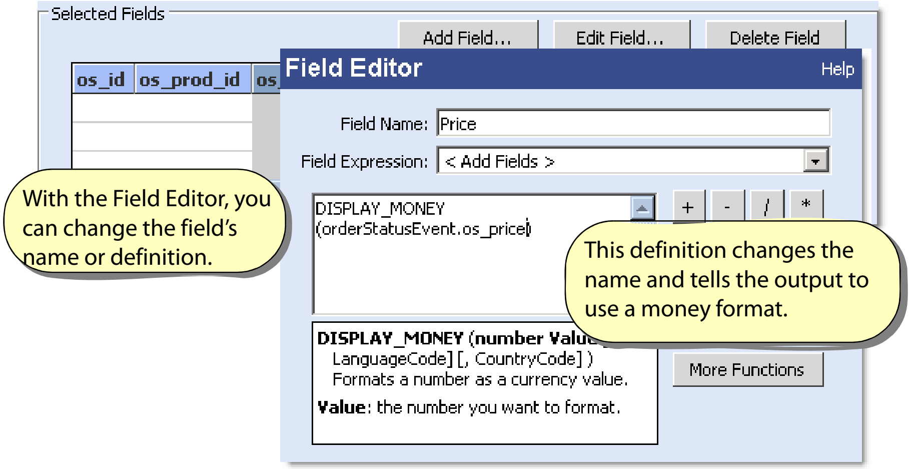
Working with View Fields

When creating a view in the Scenario Modeler, and after selecting the source tables and views, all fields appear in the view's field list. In the view editor, you can add or deleted the selected fields, change their field names, or edit their definitions.



The View Editor displays the fields, and provides buttons for changing them.

To change a field's name, or to change the definition of the field, click **Edit Field** and use the Field Editor.



Field Name: Price

Field Expression: < Add Fields >

```
DISPLAY_MONEY
(orderStatusEvent.os_price)
```

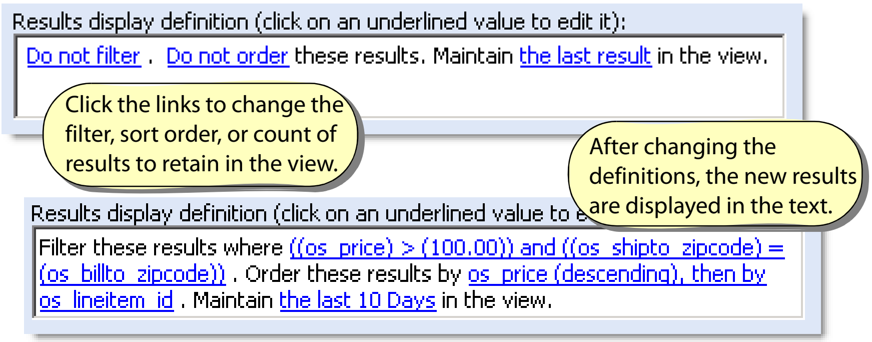
DISPLAY_MONEY (number Value
LanguageCode] [, CountryCode])
Formats a number as a currency value.
Value: the number you want to format.

With the Field Editor, you can change the field's name or definition.

This definition changes the name and tells the output to use a money format.

Affecting the Results

Three options allow you to affect the results that appear in the view: filtering, ordering, and maintaining results. The text field in the view editor displays the definitions of the affects. You can change any of them by clicking the link in the description.



Results display definition (click on an underlined value to edit it):

Do not filter . Do not order these results. Maintain the last result in the view.

Click the links to change the filter, sort order, or count of results to retain in the view.

After changing the definitions, the new results are displayed in the text.

Results display definition (click on an underlined value to e

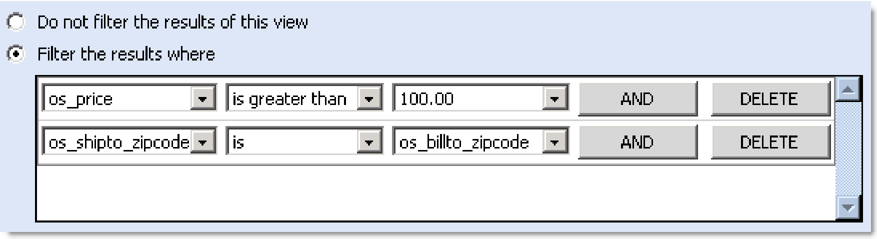
Filter these results where ((os_price) > (100.00)) and ((os_shipto_zipcode) = (os_billto_zipcode)) . Order these results by os_price (descending), then by os_lineitem_id . Maintain the last 10 Days in the view.

The following sections describe these effects in detail:

- “Filtering Results” on page 38
- “Ordering Results” on page 39
- “Maintaining Results” on page 39

Filtering Results

Filters allow you to restrict the view results to include only those rows that meet the specified condition. The condition can be simple; for example, *only those products whose price is greater than 100.00*. Filters can also be complex and with multiple conditions, all of which must be met before the result is allowed into the view. The example in illustration below has two conditions. The second compares the values from two tables and only allows results where the zip code value is the same in both tables.



○ Do not filter the results of this view

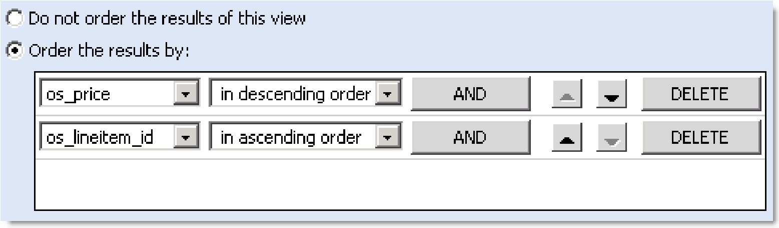
● Filter the results where

os_price	is greater than	100.00	AND	DELETE
os_shipto_zipcode	is	os_billto_zipcode	AND	DELETE

To create more complex filters, use the BAM Workbench and define the filter in the view’s Where Clause. See “Working with Clauses” on page 70 for more information.

Ordering Results

Ordering sorts the view results based on the values in one or more fields. In the Order Results dialog you pick the fields to sort, and whether to sort the results in ascending or descending order. When you choose to order on multiple fields, the results are sorted by the top field first, then in order from there. For example, in illustration below, the results are sorted by price, and items with the same price are sorted by ID:



○ Do not order the results of this view
● Order the results by:

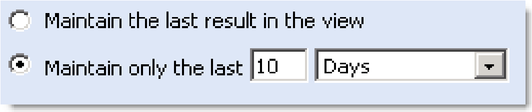
os_price	in descending order	AND	▲	▼	DELETE
os_lineitem_id	in ascending order	AND	▲	▼	DELETE

Note: The Order Results option is only available for stateful views; stateless views cannot be ordered. For a detailed explanation, see “Stateless and Stateful Views” in the *Business Activity Monitoring Server Reference*.

Maintaining Results

By default, a stateless view contains only rows representing the last event that satisfied the view condition; rows from previous events are discarded. Maintaining results allows you to specify sets of recent event information to keep around. This allows you to see the results from more than one event when you look at the view results. It also allows external applications to perform trend or historical analysis, provided that they receive the view as a real-time data feed (*metrics*).

When you choose to maintain results, you specify a count of events or of time-span units. The illustration below specifies keep 10 days worth of events:



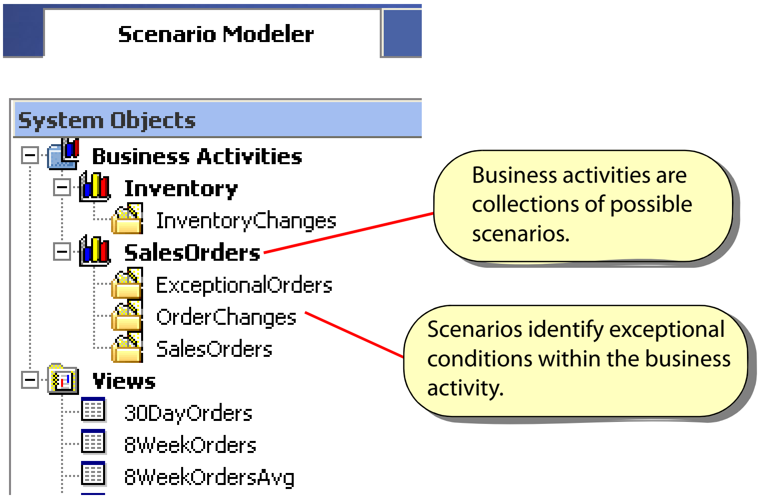
○ Maintain the last result in the view
● Maintain only the last 10 Days

The number specifies the maximum number of events to maintain. For example, when maintaining 10 events, when the eleventh event arrives, it is kept, but the oldest is discarded.

When using a time interval, the actual count of events in the view varies depending on the number of events in the interval when the view updates. The set of events is determined *when the last event was inserted into the view*, not at the current time. For example, an interval of one hour shows all the events that arrived in the view for the hour previous to the last update. If no events were inserted in the last day, the view might still show an hour’s worth of events from the previous day. However, as soon as new event arrives at the view, all those events are discarded.

Working with Business Activities and Scenarios

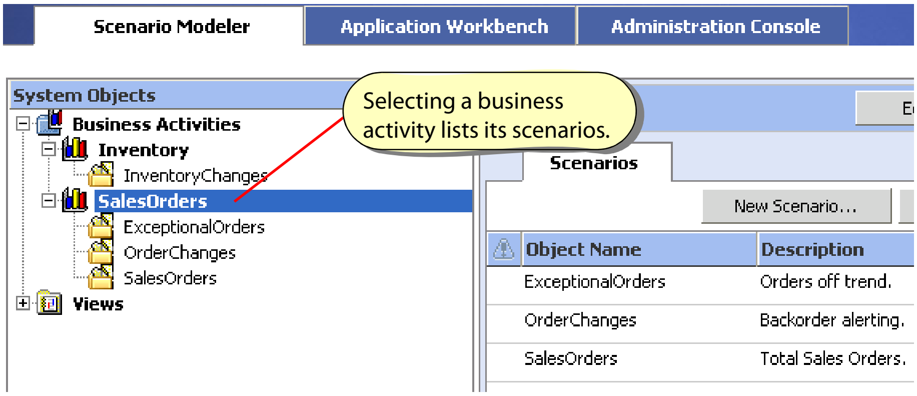
The business activities folders organize your scenarios. You must have at least one folder, but you can have as many as you need. Scenarios are a bit more restrictive. Every scenario has an associated business view, and all the rules, alerts, and reportlets within the scenario must be based on that view, or a view derived from that view.



Business activities are collections of possible scenarios.

Scenarios identify exceptional conditions within the business activity.

Selecting a business activity in the object tree lists that folder's scenarios in the contents panel. Similarly, selecting a scenario lists its rules. You can edit any of the contained objects by double-clicking them in the contents panel, or by selecting them and choosing Edit.



Selecting a business activity lists its scenarios.

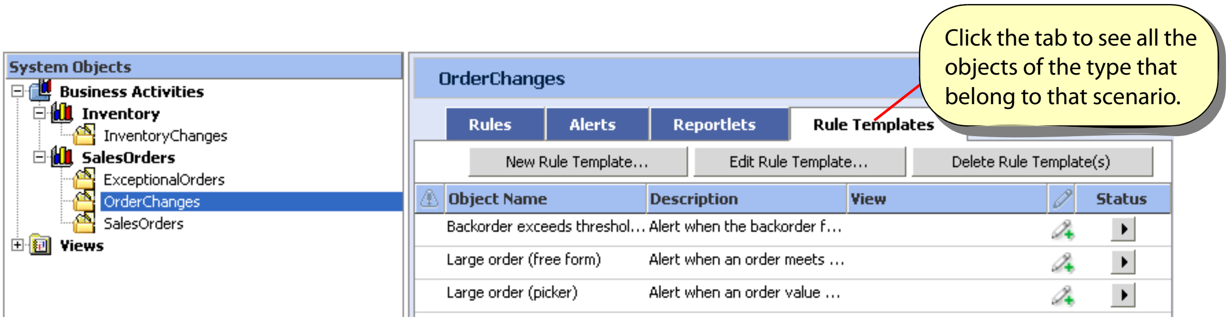
Object Name	Description
ExceptionalOrders	Orders off trend.
OrderChanges	Backorder alerting.
SalesOrders	Total Sales Orders.

The Status column in the contents panel indicates whether the object is receiving and processing new data. See "Viewing Object Status and Validity" on page 32 for details.

Note: For detailed information about each of these objects, see sections "Business Activities" and "Scenarios" in the *Business Activity Monitoring Server Reference*.

Working with Rules and Alerts

All rules and alerts belong to scenarios. Scenario provide a base business view that its rules monitor and from which its alerts and reportlets draw event and context details.



Object Name	Description	View		Status
Backorder exceeds threshol...	Alert when the backorder f...			
Large order (free form)	Alert when an order meets ...			
Large order (picker)	Alert when an order value ...			

Click the tab to see all the objects of the type that belong to that scenario.

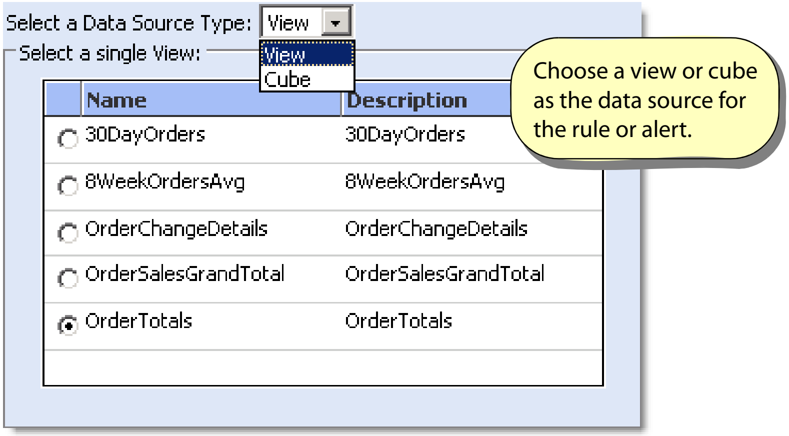
➤ **To access a rule, alert or reportlet:**

1. Open the scenario folder that contains the object.
2. To see a list of the objects, click on the **Alerts**, **Rules**, or **Reportlets** tab.
3. To edit the object, select it and click the **Edit** button.
4. To create a new object, click the **New** button.

   Note: You can create an alert without a rule, but every rule needs an associate alert as its rule-action.

Data Source

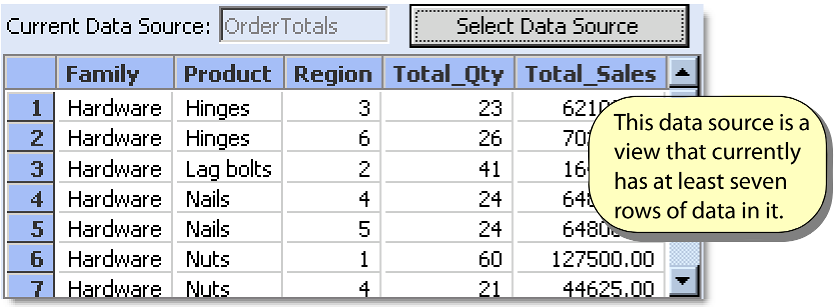
Rules monitor views or cubes (the *data source*) looking for specific conditions, and alerts retrieve event and context details from data source. The data source must be either the view associated with the scenario when you created the scenario, or a view or cube derived from the associated view. When you create a new rule, the system assumes the associated view unless you choose another by clicking **Select Data Source**.



Name	Description
○ 30DayOrders	30DayOrders
○ 8WeekOrdersAvg	8WeekOrdersAvg
○ OrderChangeDetails	OrderChangeDetails
○ OrderSalesGrandTotal	OrderSalesGrandTotal
● OrderTotals	OrderTotals

Choose a view or cube as the data source for the rule or alert.

If the data source currently has any results in it, those values are shown to assist you as you create the rule or alert.



Current Data Source: OrderTotals

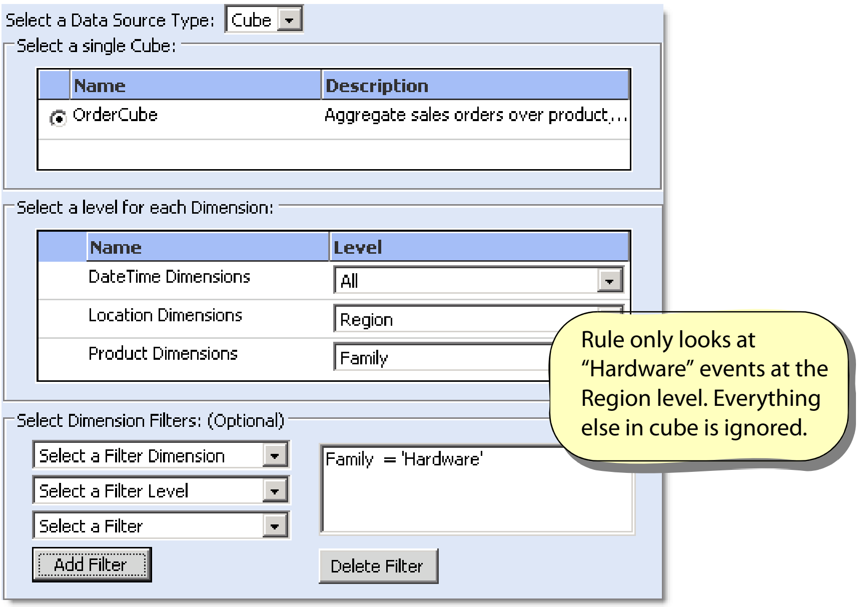
	Family	Product	Region	Total_Qty	Total_Sales
1	Hardware	Hinges	3	23	6210
2	Hardware	Hinges	6	26	70
3	Hardware	Lag bolts	2	41	16
4	Hardware	Nails	4	24	64
5	Hardware	Nails	5	24	6480
6	Hardware	Nuts	1	60	127500.00
7	Hardware	Nuts	4	21	44625.00

This data source is a view that currently has at least seven rows of data in it.

For details about creating a rule based on a view, see "Working with Rule Details" on page 43. However, to create a rule based on a cube, you need to identify the source dimension level, described next.

Cubes as the Data Source

Cubes organize their data by dimensions, and then by levels within each dimension. (See "Cubes" in the *Business Activity Monitoring Server Reference* for details.) When the rule's data source is a cube you identify the dimension level to monitor and optionally apply a filter that further restricts the data that the rule monitors.



Select a Data Source Type: Cube

Select a single Cube:

Name	Description
● OrderCube	Aggregate sales orders over product,...

Select a level for each Dimension:

Name	Level
DateTime Dimensions	All
Location Dimensions	Region
Product Dimensions	Family

Select Dimension Filters: (Optional)

Family = 'Hardware'

Rule only looks at "Hardware" events at the Region level. Everything else in cube is ignored.

Setting an Alert to Invoke an External Web Service

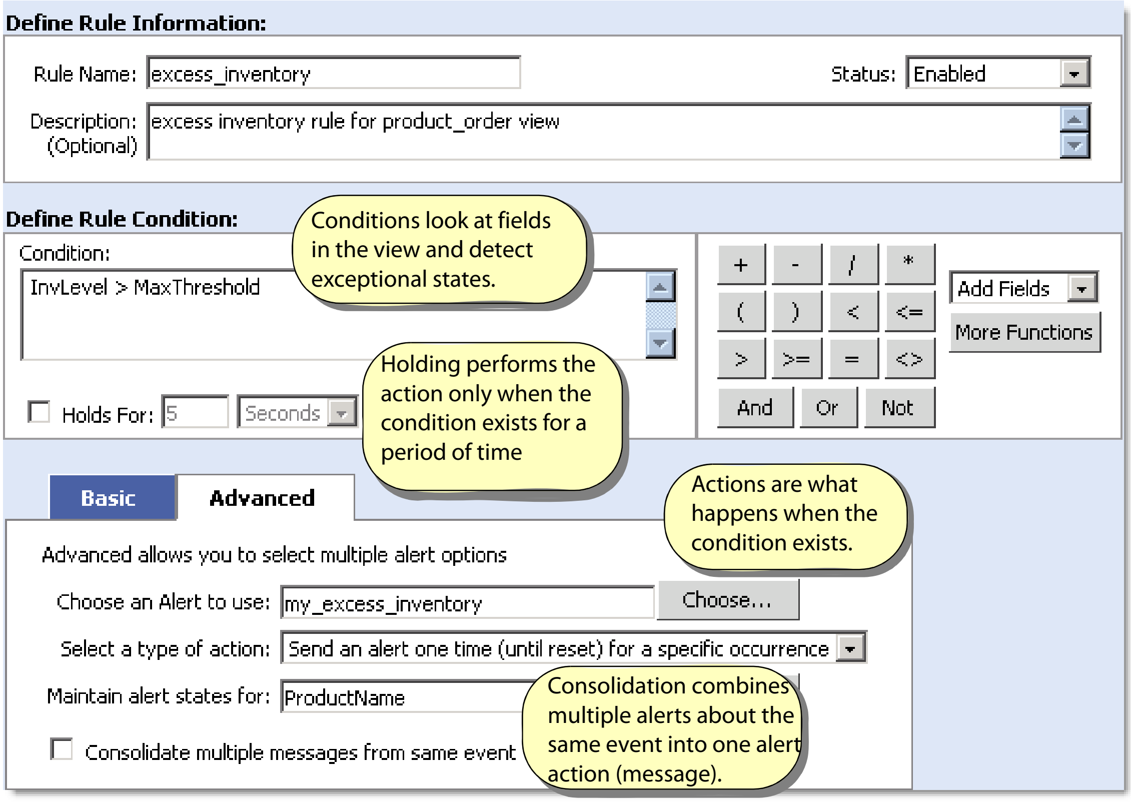
You can now configure an alert message to invoke an external Web service. This requires the creation of a function that constructs the Web service message based on the alert payload; specifically, this function determines which data elements in the alert payload are mapped to the inputs expected by the Web service. This function is then uploaded to BAM as a user-defined function (UDF). The Web service is specified in a Web service profile that calls the UDF. The Web service profile can then be set as a subscriber to the desired alert is defined.

➤ **To configure an alert to invoke a Web service:**

1. Create a function that constructs the Web service message using parts of the alert data, and package as a JAR.
2. Upload the JAR as a user-defined function in the BAM Workbench.
3. Create the Web service profile by creating a new user, then setting the Delivery Profile parameters as follows:
   - For profile type, select *Web Service*.
   - For profile name, enter an appropriate value.
   - For Web Service URL, enter the endpoint for the WSDL.
   - For method, specify the desired method (operation) to be invoked in the Web service.
   - Supply username and password, as required.
   - For UDF, select the JAR you uploaded for this Web service invocation.
   - Click **OK**.
4. When you create or modify the alert intended to invoke the Web service, you can add the new Web service profile as a subscriber.

Working with Rule Details

You define rules with the Rules Editor. Every rule has a name that identifies it, a condition to look for, and an action to take when the condition exists. The action either generates an alert, or lowers a previously raised alert. Optionally, you can tell the rule to perform the action only when the condition “holds” true for a period of time.



Define Rule Information:

Rule Name: excess_inventory

Status: Enabled

Description: (Optional) excess inventory rule for product_order view

Define Rule Condition:

Condition: InvLevel > MaxThreshold

□ Holds For: 5 Seconds

Basic | Advanced

Advanced allows you to select multiple alert options

Choose an Alert to use: my_excess_inventory

Select a type of action: Send an alert one time (until reset) for a specific occurrence

Maintain alert states for: ProductName

□ Consolidate multiple messages from same event

Conditions look at fields in the view and detect exceptional states.

Holding performs the action only when the condition exists for a period of time

Actions are what happens when the condition exists.

Consolidation combines multiple alerts about the same event into one alert action (message).

Rule Conditions

A rule condition is a formula that tests the row in the associated view or cube looking for a specific condition. When the condition exists the rule action activates an alert. They can be simple tests for a value in a column in the view, like InvLevel>MaxThreshold, or they can be complex expressions with functions, operators, and parenthesis groupings; for example, (InvLevel<MinThreshold OR InvLevel<(AverageInvLevel*.90)). For a detailed discussion about conditions, see the section “Rule Condition” in the *Business Activity Monitoring Server Reference*.

Rule Actions

Rules can have one of three actions:

- Send alerts every time the condition is found to exist (*fire*). A fire action sends an alert every time a rule identifies an exceptional condition. For example, send an alert every time inventory drops below the minimum threshold.
- Send alerts once and ignore subsequent events until the initial condition is resolved (*raise*). A raise action sends an alert message when the rule's condition applies, but ignores subsequent events until after the initial condition is resolved. A raise action is useful when you don't want multiple alerts for situations where the rule condition is true for multiple, related events. For example, once the inventory falls below the minimum threshold, don't send another alert if the inventory continues to drop.

  The "for a specific occurrence" option allows you to send alerts once for each specific occurrence of the named column. For example, send an alert every time the inventory falls below the minimum threshold *for each specific product*.
- Reset (*lower*) previously raised rules to allow them to again send new alerts. For example, when the inventory is once again above the minimum threshold, rest the alert so new ones can be sent if it subsequently falls below the minimum again.

For a detailed discussion about actions, see Rule Action.

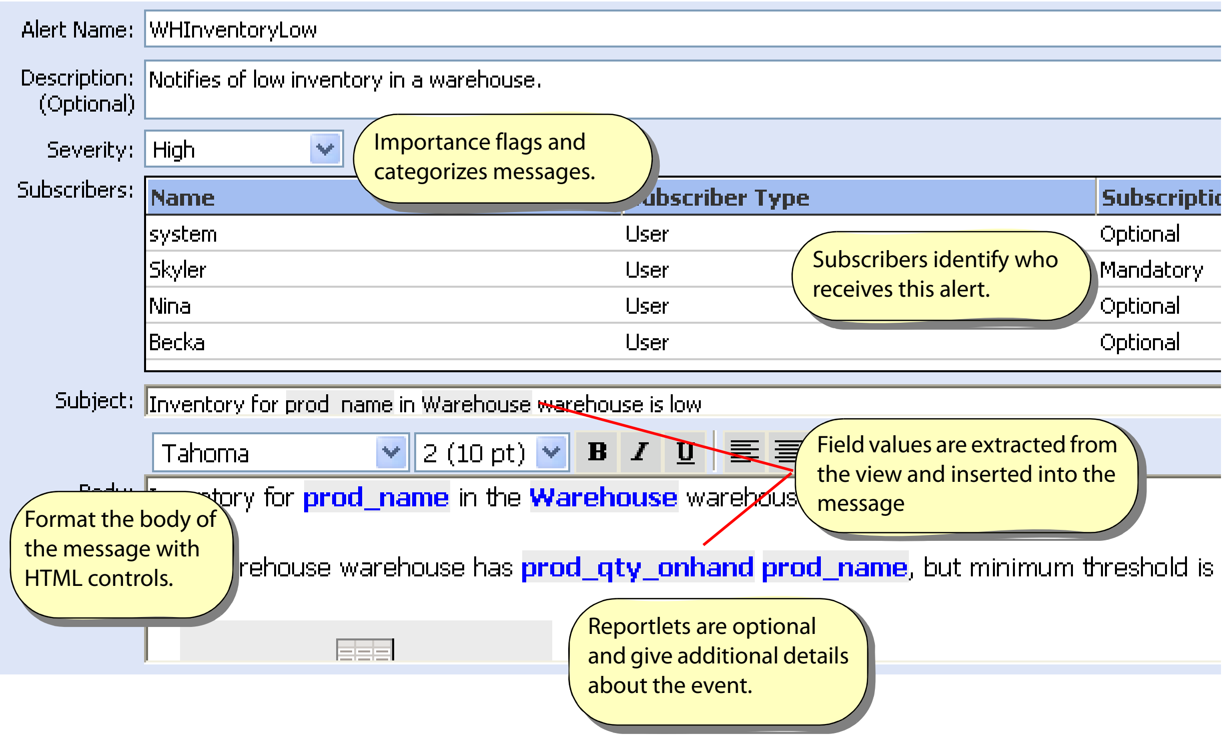
Holds For

Rules usually perform their action as soon as the condition exists. However, in some cases you would rather wait to see if the condition is corrected before alerting key personnel. In that case, you can specify a duration of time to wait. If the condition exists the entire time while the rule waits, the system then sends the alert. For example, only send the alert if the inventory remains below the minimum threshold for 2 days.

You can enter a number that the length or time or count of events to wait, or the name of a column in the view that contains the number. When you use column name option, the rule takes the value from the event in the view. In this way the event can determine how long to wait.

Working with Alert Details

You define alerts with the Alert Editor. Every alert has a name that identifies it, a subject that tells the recipient what the alert is about, and an optional message body that provides more detail in an HTML format. Further, you can attach reportlets to the alert to provide greater details about the event as retrieved from a data source. By default the system uses the view associated with the containing scenario. To choose another, click **Select Data Source**. See "Data Source" on page 41 for more information.



Alert Name: WHInventoryLow

Description: (Optional) Notifies of low inventory in a warehouse.

Severity: High

Subscribers:

Name	ubscriber Type	Subscripti
system	User	Optional
Skyler	User	Mandatory
Nina	User	Optional
Becka	User	Optional

Subject: Inventory for prod name in Warehouse warehouse is low

Body: I...tory for prod_name in the Warehouse warehous
...rehouse warehouse has prod_qty_onhand prod_name, but minimum threshold is

Importance flags and categorizes messages.

Subscribers identify who receives this alert.

Field values are extracted from the view and inserted into the message

Format the body of the message with HTML controls.

Reportlets are optional and give additional details about the event.

Subscribers

Subscribers are the users who will receive this alert. A mandatory subscriber will always receive this alert. An optional subscriber may choose to unsubscribe from it in the BAM Dashboard. For details about subscribing to alerts, see "Setting Notification Properties" in *Using Business Activity Monitoring Dashboard*.

Note: For a detailed discussion about subscribers, see Alert Subscribers in the object reference guide.

Fields

The **Add Field...** button allows you to insert a field into the subject or body text from the Add Field dialog. When the alert is generated, the value of the field in the view or cube is inserted into the text, thereby replacing the field place holder. In the illustration above, the value of ProductName from the view or cube is inserted into the Subject line.

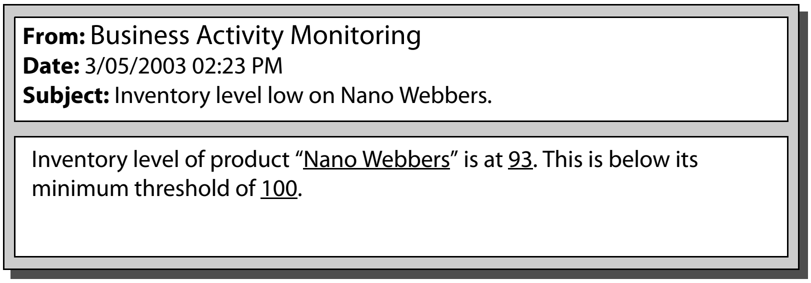
When the associated view or cube has multiple rows for the event, by default, the value from each row is inserted into the field place holder separated by commas. For example, the Subject might look like the following:

```
Inventory level low on Nano Webber, Smoke Shifter, Locking Rail Key
```

Further, when the field has the same value in multiple rows, each occurrence is inserted. Instead, you can *consolidate duplicate values* by choosing that option on the Add Field dialog. This dialog also allows you to specify the character or string that separates duplicate values.

Subject and body

The subject and body fields are similar to an e-mail message: they provide summary and detail descriptions about the event that the rule identified. Each field may contain a static message or text with placeholders to be filled in when the message is generated. For example, if the alert above is sent as an e-mail, it might look similar to the following:



From: Business Activity Monitoring
Date: 3/05/2003 02:23 PM
Subject: Inventory level low on Nano Webbers.

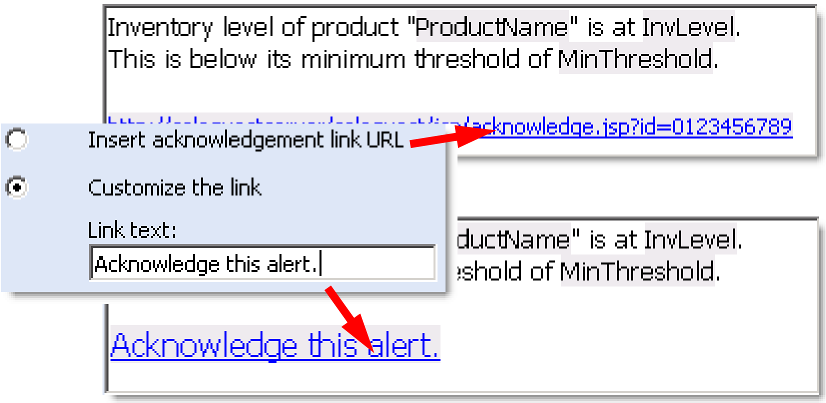
Inventory level of product “Nano Webbers” is at 93. This is below its minimum threshold of 100.

For a complete discussion about alerts, see “Alerts” in the *Business Activity Monitoring Server Reference*.

Acknowledgements

An acknowledgement is a link you can add to the body of the message. When a user receives the alert notification, they may click the link to acknowledge the alert. Doing so opens a new browser window that prompts the user to login, and then displays a message saying when the alert was acknowledged.

To add an acknowledgement link, point to the place in the body where you want the link inserted and click Add Acknowledgement. Users then click the link to go the acknowledgements page. The acknowledgement link appears as text in the body of the message. You use the default text, customize it to use your own text, or present the link URL as the text.



Inventory level of product "ProductName" is at InvLevel.
This is below its minimum threshold of MinThreshold.

/acknowledge.jsp?id=0123456789

○ Insert acknowledgement link URL

● Customize the link

Link text: Acknowledge this alert.

ductName" is at InvLevel.
shold of MinThreshold.

Acknowledge this alert.

Reportlets

Reportlets provide detailed information about the event, and are attached to or inserted into the alert when it is sent to the subscribers. Each reportlet's presentation is a table formatted and inserted into the message body, or included as an attachment formatted as either plain text, HTML, or a Microsoft Excel worksheet. For details about creating reportlets, see "Working with Reportlets" on page 54.

Note: You cannot create a reportlet on a cube.

Every alert may have none, one, or multiple reportlets that retrieve information from a view that is either same view that the alert uses, or a view derived from that view. The reportlet includes all of the information in the view unless you choose base the reportlet data on the Rule Filter of Event Data option. This option limits the reportlet information to include only those rows in the view that meet the condition identified by the rule that generated the alert. For example, when a view contains multiple rows, by default the reportlet includes all those rows. However, by using this option, the reportlet includes only those rows that met the rule condition.

Limiting the Number of Alert Instances

The alert manager only displays a specified number of alert instances for a user. This limitation prevents the system from being overwhelmed by too many alerts.

The default is to display the first 200 alert instances for each user. However, this setting is configurable.

➤ **To configure the number of alert instances to be displayed:**

1. Log in to BAM Workbench.

2. Open the Administration Console tab pane.

3. Click the System Settings button.

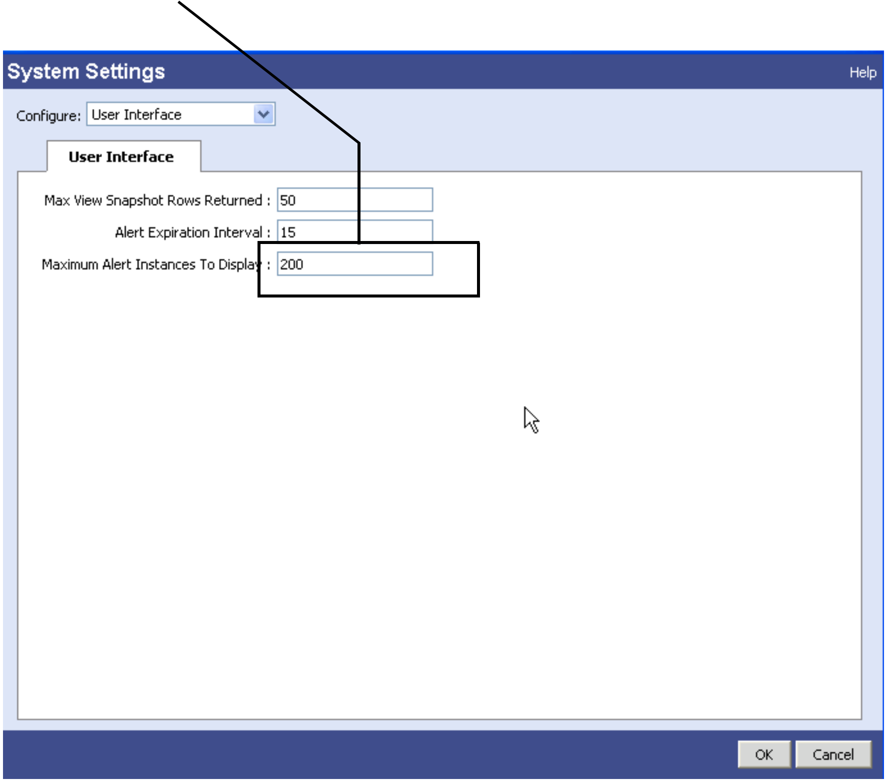
4. The System Settings dialog box opens.

5. Click the User Interface tab.

   This tab pane contains the Maximum Alert Instances To Display setting.

6. Modify the setting as desired and click **OK**.

Alert Instances setting



System Settings

Configure: User Interface

User Interface

Max View Snapshot Rows Returned : 50

Alert Expiration Interval : 15

Maximum Alert Instances To Display : 200

Working with Rule Templates

Rule templates allow BAM Dashboard users to quickly create business rules without having to define the logic of the rule's conditions or its alert message. Instead, the user chooses the template and provides the values that the rule will test for. For example, with the template condition order_total>[amount], a user provides the value of the amount parameter, and later receives notifications when that condition is true.

Defining the template once in the Scenario Modeler allows users in the BAM Dashboard to create multiple, unique business rules with the same logic, but which test for different values. For example, one user might use a template to look for orders greater than $10,000, while another looks for orders over $66,000.

Note: Business rules are based on the definition of the rule-template at the time of creation. Any subsequent changes to the template (including its alert message or reportlet) do not affect the existing business rules created from that template.

You can create a rule template that monitors either a view or cube. For views, the rule monitors every event that enters the view. For cubes, the rule monitors the results in a specific dimension level only. Additionally for cubes, alerts attached to rules cannot incorporate reportlets.

Understanding Rule Templates

Rule templates are comprised of the following:

- An activation condition — This is the test that, when true, causes the alert message to be sent.
- An optional reset condition — This test resets the alert and re-allows the activation condition to look for new events that meet the criteria.
- Parameters and prompts— Parameters are the values that the user will provide when creating the business rule. Prompts describe the parameters to the user.
- Alert message — The message template that generates the notification.
- Template and alert properties — A name and description of the template, the message's level of importance, and how the alert tracks the condition.

All of these are described in the next section, "Creating a Rule Template."

Creating a Rule Template

Creating or editing a rule template is a multi-step process where you define the conditional logic and identify the parameters that the user will provide, define the alert message, and identify the properties of the alert.

To create a rule template, you must have the following:

- Create permission for business activities
- Read and Write permission on the business activity that will contain the template
- Read Only permission on the view that will feed the rule

➤ **To create a rule template:**

1. Select the **Scenario Modeler** tab.

2. Select the scenario to maintain the template.

   The view on which the scenario is based identifies the event stream associated with the rule template. All Dashboard Objects built on this event stream will have access to the new rule-template.

3. Select the Rule Templates tab and select **New Rule**.

4. Define the activation condition.

   This is the test that, when true, causes the alert message to be sent. Conditions can be simple tests for a value in a column in the view, like InvLevel>[threshold], or that can be complex expressions with functions, operators, and groupings, like this (InvLevel>[threshold] OR InvLevel>(AverageInvLevel*.90)). For a detailed discussion about conditions, see Rule Condition.

   - Insert any parameters that the user will need to provide. See "Parameter Details" on page 53 for information.
   - Optionally define a Holds for duration. See "Holds For" on page 45 for details.

5. (Optional) Define the reset condition.

   This test resets the alert and re-allows the activation condition to look for new events that meet the criteria. If you omit this condition, the activation test sends a notification each time an event meets the criteria (a "fire" rule). Include this condition to send the notification once ("raise"), and not send another until the reset condition ("lower") is true. See "Rule Actions" on page 45 for more information.

- Insert any parameters that the user will need to provide. See "Parameter Details" on page 53 for information.

6. Define the details for the template parameters.

   - Define the prompts that describe each parameter to the BAM Dashboard user. See "Prompts" on page 53 for details.
   - Specify whether or not to also show the condition(s) to the user.

7. Define the alert message.

   The message template that generates the notification. This page is a minimum set of the parameters of an independent alert. See "Working with Alert Details" on page 45 for details about all of the parameters.

8. (Optional for view-based templates) Add a reportlet.

   Insert or attach an existing or new reportlet to include with the alert message. See "Working with Reportlets" on page 54 for details. This option is not available for templates based on cubes.

9. Define the rule template properties.

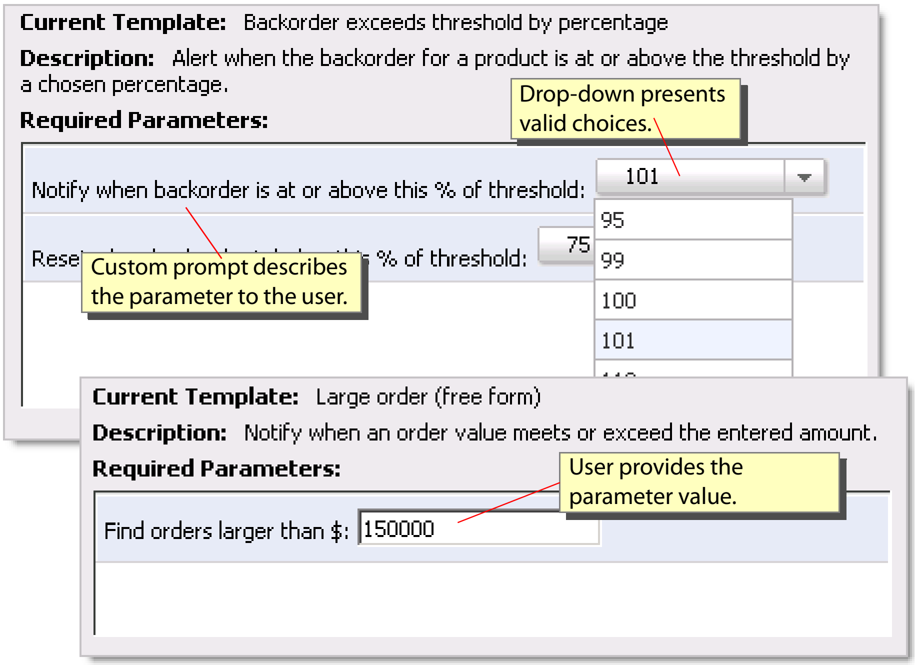
   - Provide a name and optional description. Use the description to describe the template to the BAM Dashboard user.
   - (Optional) Specify the importance level of the alert message.
   - (Optional) Specify whether or not to consolidate multiple alerts from the same event. See "Consolidating Multiple Messages" in the *Business Activity Monitoring Server Reference* for details.
   - (Optional) Identify the view's columns that uniquely describe the event to track when using a reset condition. For example, if the rule looks for product back orders that exceed a threshold, you want to track separate alerts for each product. Do that by choosing the columns that uniquely identify the products, such as product name and product family.

     Omit column selection when you do not need to track alerts individually, such as when *total sales-to-date are below target-to-date.*

10. Save the template to make it immediately available to BAM Dashboard users.

Setting Parameters

Parameters are the values that the user provides in the BAM Dashboard when creating a template-based rule. When defining the template, you can either provide list of values for the user to choose or omit the choices and let the user enter the value. Further, you can provide the prompt that the user sees next to the parameter field.



Current Template: Backorder exceeds threshold by percentage

Description: Alert when the backorder for a product is at or above the threshold by a chosen percentage.

Required Parameters:

Notify when backorder is at or above this % of threshold: 101

Rese % of threshold: 75

95
99
100
101
Drop-down presents valid choices.
Custom prompt describes the parameter to the user.

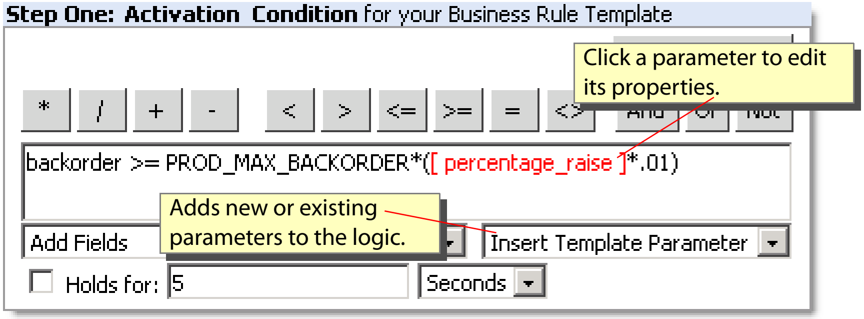
Current Template: Large order (free form)

Description: Notify when an order value meets or exceed the entered amount.

Required Parameters:

Find orders larger than $: 150000

In the Rule Template editor, add parameters to the condition logic with the Insert Template Parameter option. You can insert a new parameter or choose one already defined for the template.



Step One: Activation Condition for your Business Rule Template

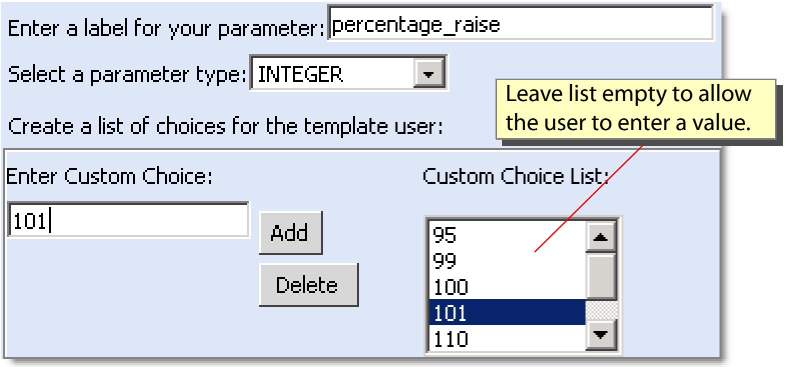
```
backorder >= PROD_MAX_BACKORDER*([ percentage_raise ]*.01)
```

Holds for: 5 Seconds

*
/
+
-
<
>
<=
>=
=
<>
And
Or
Not
Add Fields
Insert Template Parameter
Click a parameter to edit its properties.
Adds new or existing parameters to the logic.

Parameter Details

Use the Parameter Details dialog to customize the parameter's name, data type, and valid choices. Add the valid choices to the drop-down list, or omit the choices to allow the user to provide the value.



Enter a label for your parameter: percentage_raise

Select a parameter type: INTEGER

Create a list of choices for the template user:

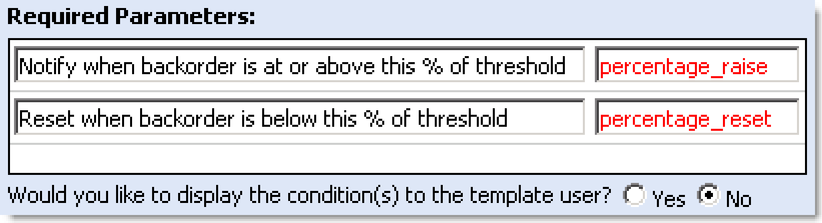
Leave list empty to allow the user to enter a value.

Enter Custom Choice: 101

Custom Choice List: 95, 99, 100, 101, 110

Prompts

When creating the business rule, the user see a prompt next to each parameter that describes what they are choosing. Define the prompt in the Rule Template editor.

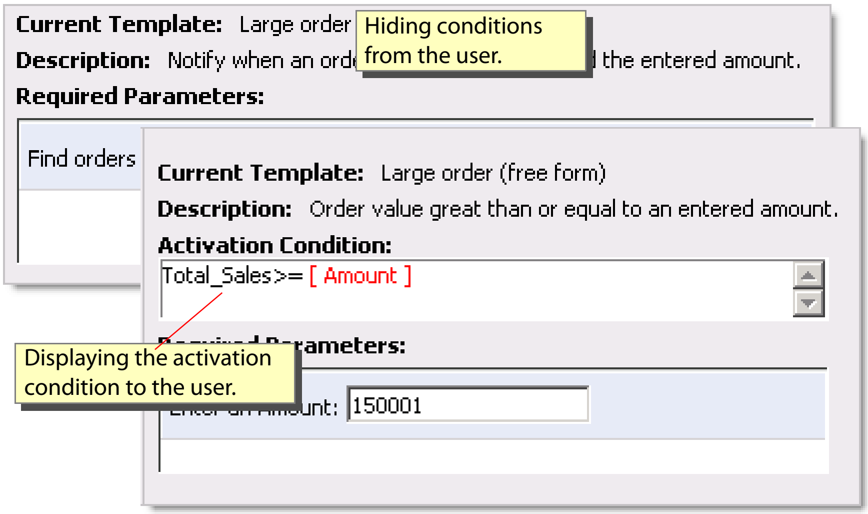


Required Parameters:

Notify when backorder is at or above this % of threshold	percentage_raise
Reset when backorder is below this % of threshold	percentage_reset

Would you like to display the condition(s) to the template user? ○ Yes ● No

In addition to the parameters and prompts, you may choose to display the rule condition or conditions to the user.



Current Template: Large order

Description: Notify when an ord the entered amount.

Required Parameters:

Find orders

Hiding conditions from the user.

Current Template: Large order (free form)

Description: Order value great than or equal to an entered amount.

Activation Condition:

Total_Sales>= [Amount]

Required Parameters:

Enter an Amount: 150001

Displaying the activation condition to the user.

Working with Reportlets

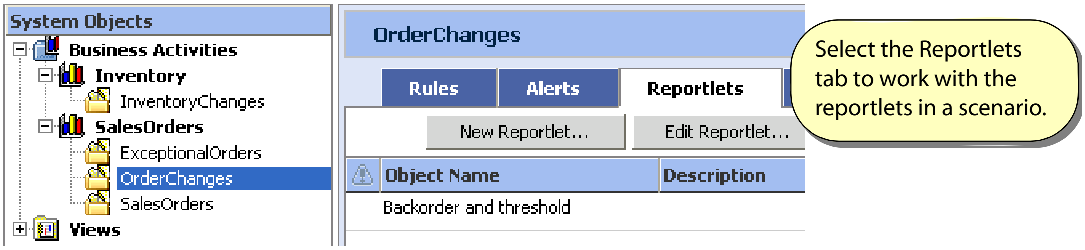
Reportlets describe the contents of a view and present that information in a report that is either attached to an alert message or sent to an external system. Frequently, reportlets provide information about an event that puts the event into context; for example, when an inventory is low for a product, and a restock shipment is overdue, an alert might notify purchasing managers of that state and a reportlet attached to the alert might list the alternative suppliers for that product. Reportlets are attached to all subscribers of the associated alert.

There are two types of reportlets:

- Internal reportlets are the visual representation of the information in a view when the alert generated the reportlet. The presentation is a table formatted in either text, HTML, or a Microsoft Excel worksheet, and contains all of the information that was in the view.
- External (3rd-party) reportlets are produced by external reporting systems. External reportlets generate their reports based on view data passed to them when the alert generates the reportlet.

Note: External reportlets are only available when an external reporting system has been defined. See "Working with External Links" on page 84 for details.

All reportlets are associated with a scenario. As such, to create, edit, or delete a reportlet, choose the Reportlets tab in the desired scenario.



Backorder and threshold

Select the Reportlets tab to work with the reportlets in a scenario.

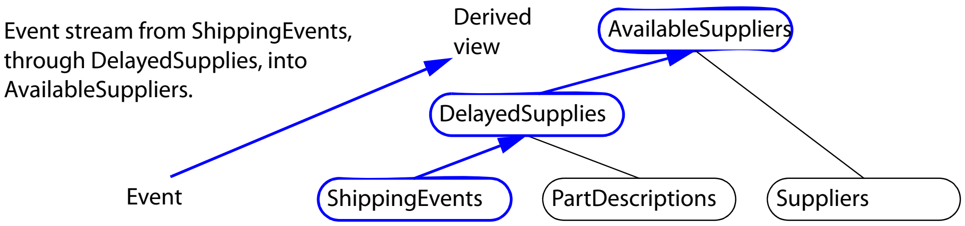
When you create or edit a reportlet, you must identify the following properties:

- The type of reportlet (internal or external)
- The business view on which the reportlet is based and from which it retrieves the values to report
- For internal reportlets, you may optionally define Microsoft Excel worksheets to contain the reportlet.

4 Enabling Support for Business Views

New events entering the system become part of a stream that flows from the event source to a business view, possibly passing through other business views along the way. As such, every business view depends on an event or another business view, and every view optionally joins the event information with context information used to provide a meaningful picture of the activity.

The illustration below shows the view *AvailableSuppliers* as part of an event stream starting with *ShippingEvents* and passing through *DelayedSupplies*. *PartDescription* and *Supplies* are each joined context sources.



The BAM Workbench in BAM Workbench is where you create and manage the events, contexts, and views that support business views.

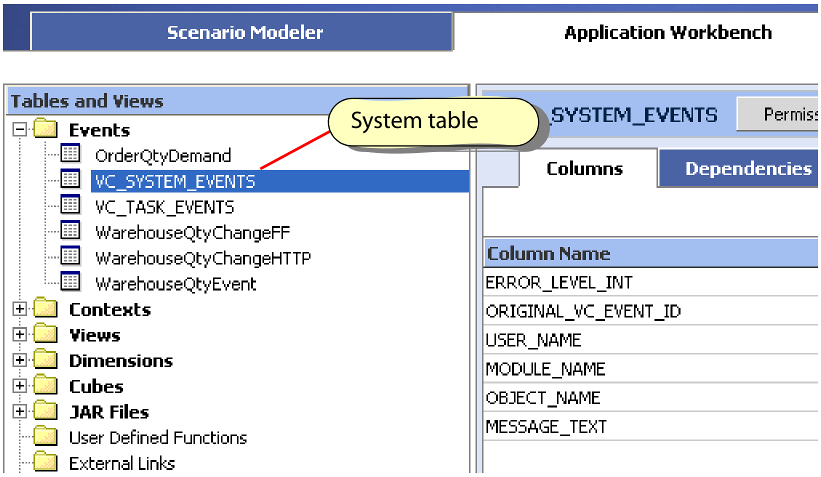
Note: Views created in the BAM Workbench can be defined with complex join conditions. You can also create views in the Scenario Modeler, but those new views are dependant on join conditions defined in the BAM Workbench.

➤ **In this Chapter:**

Working with the BAM Workbench

The left side of the BAM Workbench lists the event and context tables, views, JAR files, and User Defined Functions (UDFs) that you have created or which you have access to.

In the Events folder, tables whose names begin with "VC_" are system tables that you may access, but whose definition you cannot change. For example, the VC_SYSTEM_EVENTS table contains events generated by Business Activity Monitoring. (See "Monitoring the Logs" on page 116 for more information about this event.)



System table

Column Name
ERROR_LEVEL_INT
ORIGINAL_VC_EVENT_ID
USER_NAME
MODULE_NAME
OBJECT_NAME
MESSAGE_TEXT

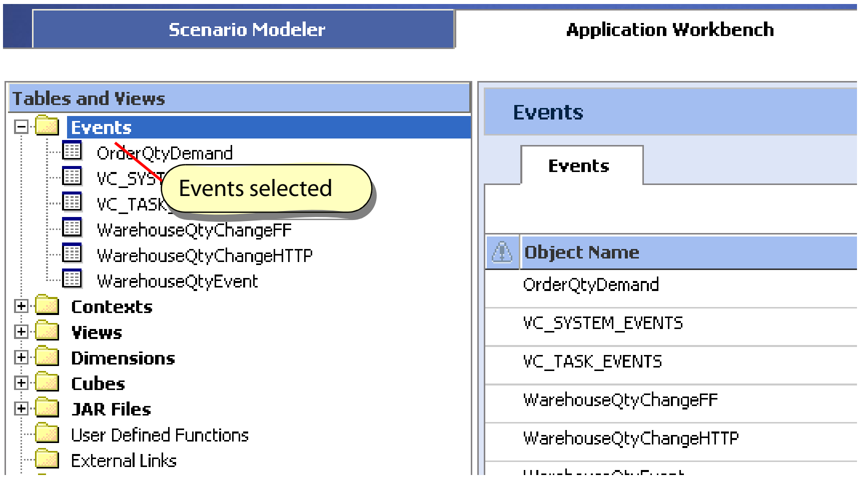
➤ **To create a new object:**

- Clicking the appropriate **New** button above the tree.

➤ **To display a detailed list of objects:**

- Click on the folders in the object tree to display a detailed list of the contained objects in the contents panel.

  Note: From the list in the contents panel you can delete the object or change its status. For more information, see "Viewing Object Status and Validity" on page 32 for a description of the Status indicator.



Events selected

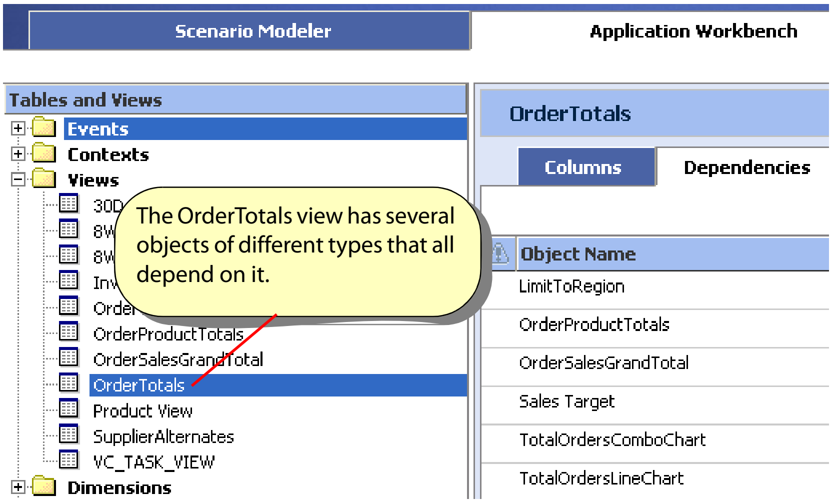
Object Name
OrderQtyDemand
VC_SYSTEM_EVENTS
VC_TASK_EVENTS
WarehouseQtyChangeFF
WarehouseQtyChangeHTTP

➤ **To display object details and access the object definition:**

- Click on an object in the tree to display object details. The object detailed information includes:
  - Dependencies, described next
  - Join Relationships (described below)
  - Columns shows the column (field) specifications defined for the view
  - Results (views only) shows a snapshot of the information currently in the view

Viewing Dependencies

The Dependencies tab lists the objects that directly depend on the selected object. For example, selecting a view lists all views directly derived from the view, and all scenarios, rules, alerts, and reportlets that are reference the view.



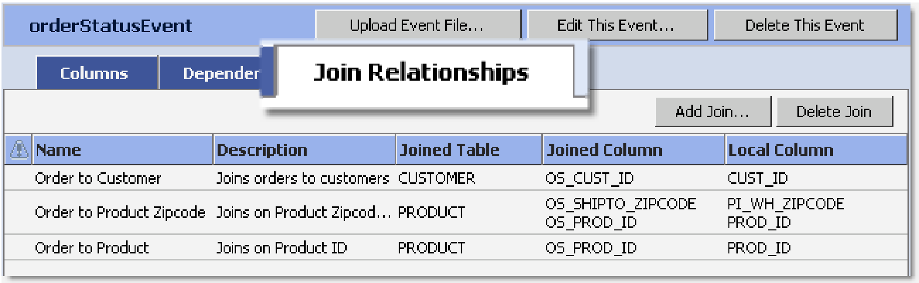
The OrderTotals view has several objects of different types that all depend on it.

Object Name
LimitToRegion
OrderProductTotals
OrderSalesGrandTotal
Sales Target
TotalOrdersComboChart
TotalOrdersLineChart

When you stop or disable an object that has dependencies, you also disable all objects that depend on the stopped object. See “Viewing Object Status and Validity” on page 32 for information about stopping and starting objects.

Viewing Join Relationship Definitions

The Join Relationships tab lists and manages the join relationships defined for all tables and views that you have access to. This tab is available in the contents panel when a table or view object is selected in the object tree in the BAM Workbench.



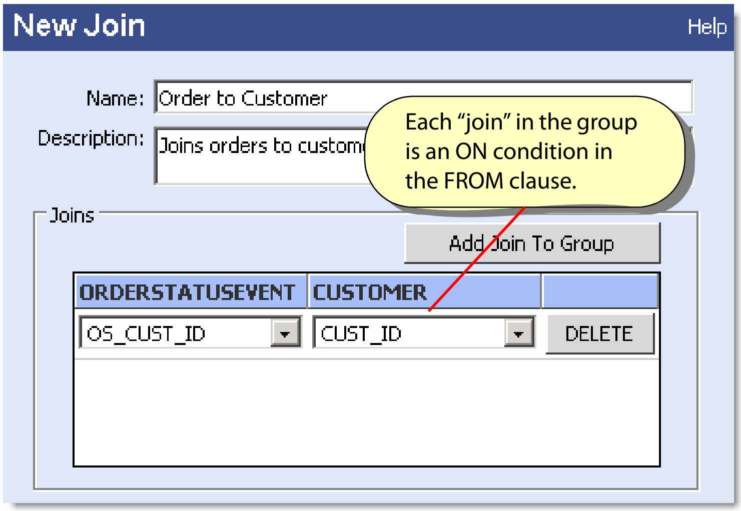
Name	Description	Joined Table	Joined Column	Local Column
Order to Customer	Joins orders to customers	CUSTOMER	OS_CUST_ID	CUST_ID
Order to Product Zipcode	Joins on Product Zipcod...	PRODUCT	OS_SHIPTO_ZIPCODE OS_PROD_ID	PI_WH_ZIPCODE PROD_ID
Order to Product	Joins on Product ID	PRODUCT	OS_PROD_ID	PROD_ID

A join relationship defines the join condition between a table and a context or between a view and a context. This relationship is used in the Scenario Modeler to facilitate view creation by hiding the details of the join from the Scenario Modeler user. In the Scenario Modeler, when users create views, they select the tables or views to join and the appropriate join relationship is then applied automatically. When multiple relationships are appropriate, the user must choose which to use. See "Understanding Relationships" on page 36 for more details about using the relationship to create a view.

A join relationship defines the inner join of the SQL SELECT FROM Clause that creates a view, similar to the following:

```
FROM (Product AS P INNER JOIN Product_Inventory AS PI
        ON P.PROD_ID = PI.PI_PROD_ID)
```

When defining this join in the BAM Workbench, it might look like the following illustration:



Name: Order to Customer
Description: Joins orders to custom

ORDERSTATUSEVENT	CUSTOMER
OS_CUST_ID	CUST_ID

Each "join" in the group is an ON condition in the FROM clause.

Each "join" in the group is an ON condition in the FROM clause. When there are multiple conditions, they are ANDed together.

➤ To create a join relationship:

1. Select one of the event, context, or view objects to be the left-side of the join.
2. Click the **Join Relationships** tab.

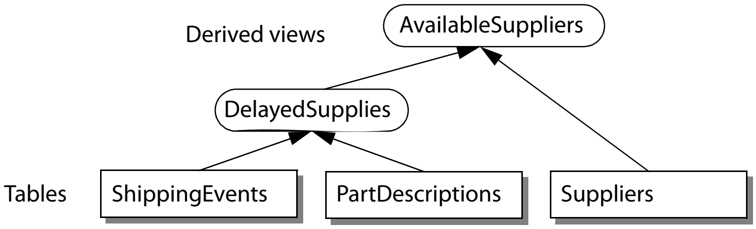
3. Click **Add Join**…
4. Select the Context table to be the right-side of the join.
5. On the New Join dialog:
   - Name the join relationship.
   - Include a description of the join. This message describes the join relationship to the Scenario Modeler user when that user is creating a view.
   - Identify the related fields by selecting **Add Join to Group** for each key.
   - Select the fields from each object on which to build the view. Usually these are key fields, such as part number or customer ID.
6. Save the join to make it available to the Scenario Modeler.

Working with Events and Context Tables

Events and contexts are the data that drive Business Activity Monitoring.

- Events feed Business Activity Monitoring and generate internal processing. Events are produced by external business applications that record transactions, identify changes in business state, and synthesize the details about the business activities.
- Contexts support event processing and provide meaningful information about events. Contexts are the business information stored in databases, data warehouses, or are provided by Web services.

Business views get their event and context information from event and context tables defined with the BAM Workbench, and the tables get their information from agents defined in the Administration Console. (See “Viewing the Agents List” on page 95 for details.) The illustration below shows *ShippingEvents* as an event table, and *PartDescriptions* and *Suppliers* as context tables that feed information into the two business views. Each of the tables must exist before you can define the business views.



With the BAM Workbench, you can create, edit, or delete the event and context tables, see which objects depend on them, and identify the join relationships between the tables and other context tables.

> **Note:** If you change a table such that one or more dependant objects become invalid — such as by deleting a column used by a view — those objects also become disabled. You then need to either modify the dependant objects to correct the problem, or restore the change to the table. Then you can re-enable the dependant objects. See "Object Status" in the *Business Activity Monitoring Server Reference* for details.

Working with Events

The Events folder in the BAM Workbench object tree lists all event tables that you have access to. Click the Events folder to see the details and object status of each table in the contents panel. Click the name of an existing table to see its dependencies and join relationships, and to edit or delete a event table. Select the Upload Event File Option to load raw event data into an event table.

> **Note:** VC_SYSTEM_EVENT is special case event stream that tracks information about Business Activity Monitoring activity. This stream is used by system administrators to monitor the system. See "Understanding Logging" on page 112 for details.

➤ **To create an event table:**

1. Select **New Event**.

   The New Event dialog box displays, prompting you to select the type of event.

2. Select the type of event.

   In most cases you will select *Single Event*. The *Consolidated Event* option is used only if you are merging multiple event sources into one stream. If so, see "Working with Consolidated Events" on page 63 for more information.

3. Click **Continue**.

   The New Event dialog box redisplays, prompting you to select the type of event source (Flat File, JDBC, and so on.).

   You can only select a type of data source if at least one has already been defined; otherwise, the option will not be available as indicated by a grayed out radio button.

   Note: For more information about agents, see "Viewing the Agents List" on page 95.

4. Select the type of event source and click **Continue**.

   The New Event dialog box redisplays, prompting you to select the specific agent for the event source.

   Some agents require additional information before you can create the event table. Specifically:

   - Flat file agents require the type of flat file: delimited, fixed width, or XML. (See "Flat File Event Tables in for complete details.) Optionally, you can identify sample data to load to assist with specifying the column definitions. When you include a sample data file, that information is presented in a table that you use to format the columns in the next step.
   - JDBC and Salesforce agents require an SQL SELECT query to retrieve data from the source database.

5. Configure and define the event table following the instructions and descriptions under "Creating Event Tables" in the *Business Activity Monitoring Server Reference*

After saving the table, it is ready to be used by business views, alerts, rules, and reportlets.

Working with Context

The Context folder in the BAM Workbench object tree lists all context tables that you have access to. Click the Content folder to see the details and object status of each table in the contents panel. Click the name of an existing table to see its dependencies and join relationships, and to edit or delete a context table.

➤ To create an context table:

1. Click **New Context**.
2. Select the type of context source.

   Only those types with previously defined agents are available. All agents are defined in the Administration Console. See "Viewing the Agents List" on page 95 for information about creating agents.
3. Identify the agent you want to use.

   Additionally, some agents require additional information before you can create the context table. For example, JDBC agents require an SQL SELECT query to retrieve data from the source database.
4. Configure and define the context table following the instructions and descriptions in Creating Event Tables.

After saving the table, it is ready to be used by business views.

Note: Context data do not change rapidly. As such, to lessen the impact on the system that provides the context data, you can elect to cache the results in memory. Subsequent requests for the same data then retrieve the results from memory instead of issuing a new query. See "Caching Context Queries" in the *Business Activity Monitoring Server Reference* for details.

Upload Event File Option

In addition to receiving events as they happen, you can upload events from a text file directly into the event stream with the Upload Event File option. This option is useful for:

- Analyzing events that were gathered in batch from another system, such as the results collected from a testing system or the Web logs gathered from HTTP servers.
- Performing "what if" scenarios by allowing you to test your formulas, rules, alerts, and so on. With this feature you can test your scenarios, modify them, and retest them with the same data.

Note: This option is only available for events whose source is a text file. See "Flat File Event Tables" in the *Business Activity Monitoring Server Reference* for detailed information about text files.

The text file to upload must include data formatted to match the format defined for the source of the event. For example, if the source is defined as receiving delimited text for 5 columns, the data in the upload file must match that format.

For XML files, the filename must end with an ".xml" extension, such as filename.xml.

➤ To upload event text into an event stream:

1. Select the event in the event tree in the BAM Workbench.
2. Select the Upload Event File option.

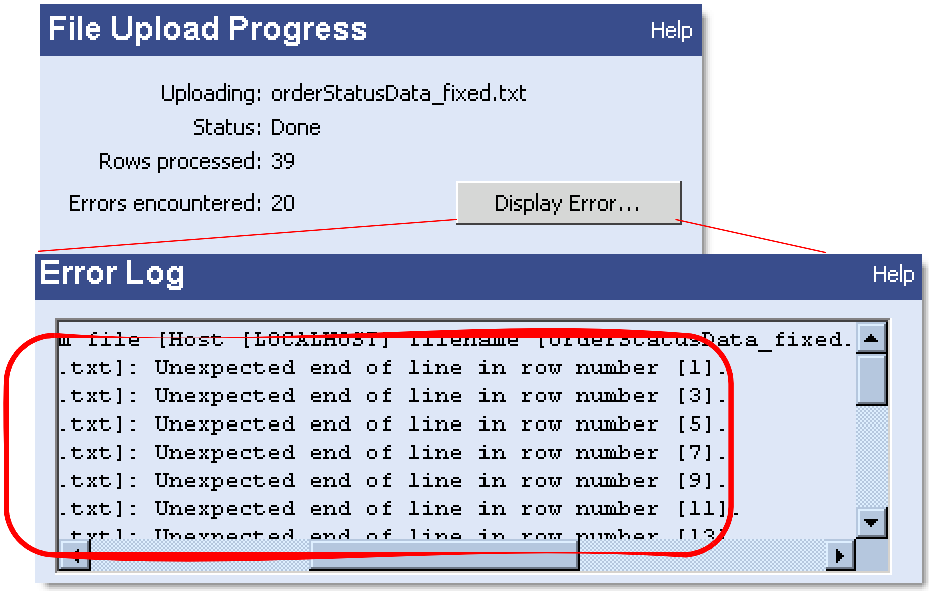
3. Identify the file to upload.

   Either enter the full path or use the browser to pick the file.

4. Click **OK** to upload the file.

5. The File Progress dialog shows the status during the upload.

   The upload is complete when the Status indicator says "Done". If the source file contains erroneous data, the Display Error button activates. Click the button to see the descriptions of the errors in the Error Log.



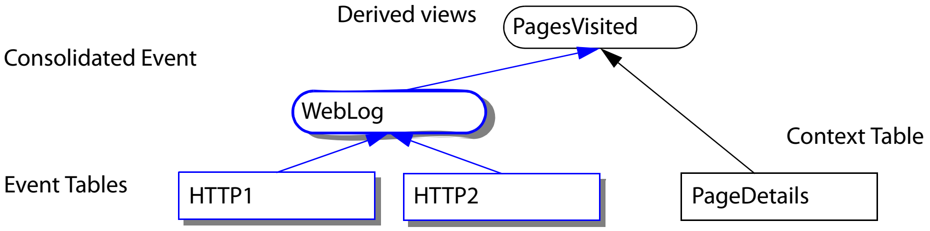
Uploading: orderStatusData_fixed.txt
Status: Done
Rows processed: 39
Errors encountered: 20

```
m file [Host [LOCALHOST] filename [orderStatusData_fixed.
.txt]: Unexpected end of line in row number [1].
.txt]: Unexpected end of line in row number [3].
.txt]: Unexpected end of line in row number [5].
.txt]: Unexpected end of line in row number [7].
.txt]: Unexpected end of line in row number [9].
.txt]: Unexpected end of line in row number [11].
.txt]: Unexpected end of line in row number [13
```

Working with Consolidated Events

Consolidated events accept events from multiple event streams. The only restriction is that *the data from each source must map directly into the consolidated events view's columns*. Consolidated events are useful for combining events from multiple similar like sources, such as the results from multiple devices testing a single product line or the records from multiple real-time point-of-sale devices.

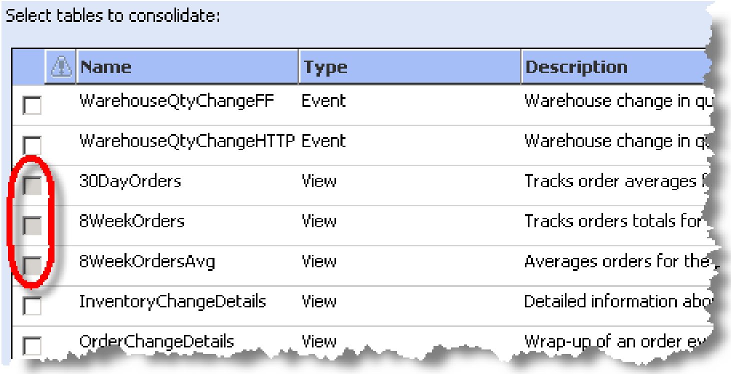
Consider the following example that combines the logs from two HTTP servers into a single event stream. A business view then combines the log information with context describing the pages to produce detailed information about the Web pages visited.



Create a consolidated event the same as you would a single event by clicking New Event in the BAM Workbench. See "Working with Events" on page 60 for details.

The tables and views on which consolidated events are built have these restrictions:

- They must supply data that maps directly into the consolidated event's columns, and the data types of the source must exactly match the target in type, width, scale and precision.
- Stateless views are unavailable for selection.



Select tables to consolidate:

	⚠ Name	Type	Description
□	WarehouseQtyChangeFF	Event	Warehouse change in qu
□	WarehouseQtyChangeHTTP	Event	Warehouse change in q
□	30DayOrders	View	Tracks order averages f
□	8WeekOrders	View	Tracks orders totals for
□	8WeekOrdersAvg	View	Averages orders for the
□	InventoryChangeDetails	View	Detailed information abo
□	OrderChangeDetails	View	Wrap-up of an order ev

These views are stateless and cannot be used as a source for a consolidated event stream.

Working with Views

Business views are data models that provide a real-time picture of a business activity. As changes and transactions occur in your business, they generate events that are sent to Business Activity Monitoring. Each new event drives an immediate update of the views (the business models) which in turn provides a real-time, updated view of the business metrics.

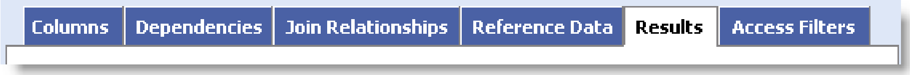
In the BAM Workbench you create business views derived from event and context tables, and from other business views. You can also define:

- How the view results are grouped and ordered
- Formulas for aggregating the information displayed in the view columns
- Complex join conditions (where clauses) that specify how to relate the source data
- Access filters that allow different users see different rows of the same view depending on the criteria specified in the filter. These filters restrict the data a user sees without having to define a new view for each user. For complete details and instructions, see "Access Filters" on page 15.

Note: You can also create views in the Scenario Modeler, but those new views are dependant on join relationships defined in the BAM Workbench (see "Viewing Join Relationship Definitions" on page 58 for details). Views created in the BAM Workbench have join relationships defined in the Where Clause.

View Tabs

Click on any view in the Views folder to see tabs that provide detailed information about that view.



- Descriptions and definitions of the view's columns.
- Objects that depend on this view (see "Viewing Dependencies" on page 57 for details).
- Join relationships for this view (see "Viewing Join Relationship Definitions" on page 58 for details).
- Reference data for plans on this view ("Viewing Plan Relationships" on page 74).
- Current values in the view (if any).
- Access filters applied to this view, as described above.

The rest of this section describes how to use the View Editor to create and modify views:

- "Creating and Editing Views in the BAM Workbench" on page 65
- "Creating View Fields" on page 66
- "Working with Clauses" on page 70
- "Maintaining Events in a Stateless View" on page 72
- "Persisting View Data" on page 72
- "Drill Back to Detail" on page 73
- "Displaying SQL Expressions" on page 73

Creating and Editing Views in the BAM Workbench

All view creation and modification is done with the View Editor. This editor displays the tables and views on which the current view gets its information (the Workset), the list of fields or columns in the view, and how the information is joined and ordered.

Note: When you change any part of the definition of a view, all results currently in the view and in any views derived from the modified view are discarded.

➤ **To create a new view in the BAM Workbench:**

1. Click the **New View** button.
2. Select an event or view that is the event stream for the new view.
3. Optionally select one or more context tables to include in the view.
4. Define the view in the View editor.

Save the view as enabled and it will immediately be ready to receive events.

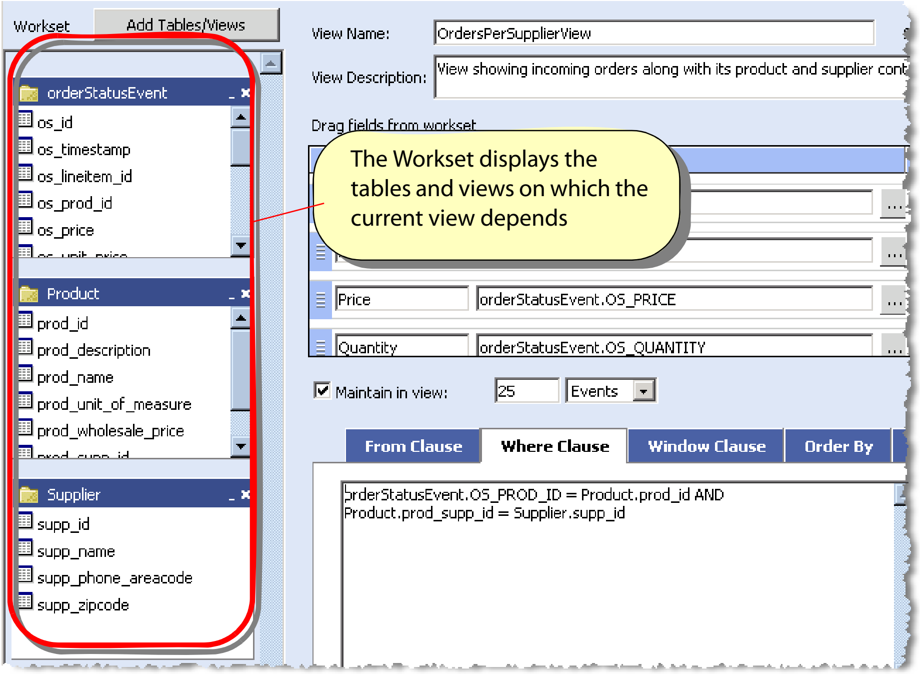
➤ **To modify an existing view in the BAM Workbench:**

1. Select the view’s name in the object browser
2. Click **Edit This View**.
3. Change the definition in the View Editor.

Save the view as enabled and it will immediately be ready to receive events.

Workset

When you create a view, you first select the event stream that drives the view. An event stream is either an event table or another business view. Optionally, you also select one or more context tables to join with the the event stream. These tables and views become part of the *workset* of the derived view. In the view editor, you can see the workset tables and views, and their columns on the left side of the page.



View Name: OrdersPerSupplierView

View Description: View showing incoming orders along with its product and supplier cont

Price	orderStatusEvent.OS_PRICE
Quantity	orderStatusEvent.OS_QUANTITY

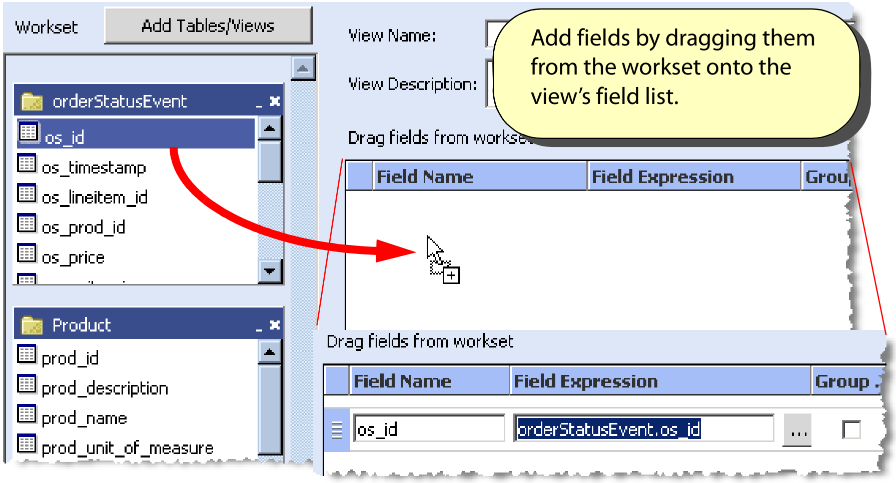
☑ Maintain in view: 25 Events

```
orderStatusEvent.OS_PROD_ID = Product.prod_id AND
Product.prod_supp_id = Supplier.supp_id
```

The Workset displays the tables and views on which the current view depends

Creating View Fields

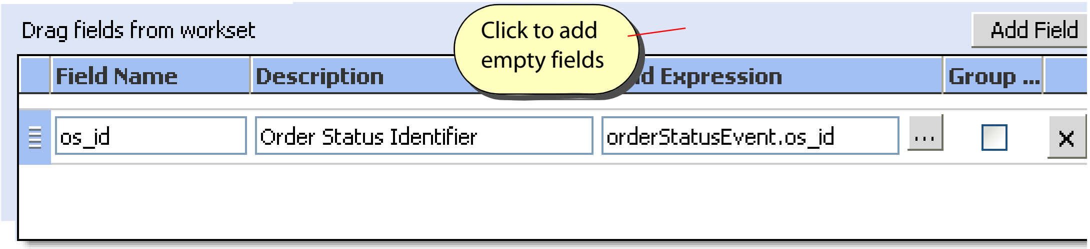
Fields in a view are usually derived from fields in the source tables and view but may be formulas independent of the source table, such as a field that identifies the current date/time. To add a field to a view, drag the fields from the source in the workset onto the view's field list.



Field Name	Field Expression	Group
os_id	orderStatusEvent.os_id	□

Add fields by dragging them from the workset onto the view's field list.

Add new, empty fields by click **Add Field** on the “Drag or type fields here” label.



Field Name	Description	...d Expression		Group ...	
os_id	Order Status Identifier	orderStatusEvent.os_id	...	□	✕

Field Names

Field names are the names that appear in view results and is the name that rules, alerts, reportlets, and other views use to reference the field. Names may be mixed case, and may include numerals and simple text symbols. However, names may not coincide with system reserved words. See “Object Names” in the *Business Activity Monitoring Server Reference* for details about names.

Field Definitions

Each field definition is an expression or formula that identifies what information to produce in the view. By default, definitions are references to the source field. For example, orderStatusEvent.OS_ID is the name of the *orderStatusEvent* event stream followed by the *OS_ID* column name in the source table.

More complex definitions include functions that modify the field value or display, or perform some analytic operation. The following are some examples:

- DISPLAY_MONEY(OS_PRICE) formats a number as a currency value, such as ‘$1,234.56’.
- TO_CHAR(CURRENT_TIMESTAMP(), 'd MMMM yy') formats the current data as ‘5 March 03’.

More complex formulas might have nested functions, like the following:

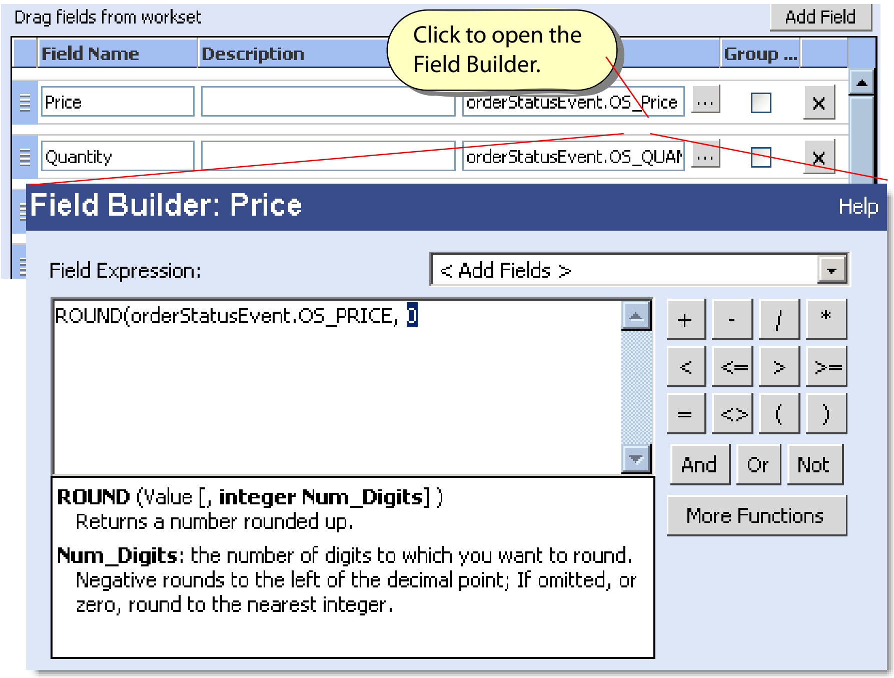
- DISPLAY_MONEY(AVG(OS_PRICE)) formats the average price as a currency value, such as ‘$1,234.56’.
- CAST(((OS_PRICE/OS_COST)-1)*100 AS DECIMAL(5,0))||'%' determines the percentage markup on the price of an item over its cost, strips out the decimals, and concatenates a percent symbol. The result might look like ‘18%’.

Note: To learn more about functions and expressions, see “Functions” in the *Business Activity Monitoring Server Reference*.

You can type formulas directly in the field definition, or you use the Field Builder to quickly create formulas.

Field Builder

The Field Builder is a dialog that assists you when creating field definitions. Open the Field Builder by clicking (…) next to the field definition.



```
ROUND(orderStatusEvent.OS_PRICE, 0
```

ROUND (Value [, **integer Num_Digits**])
Returns a number rounded up.

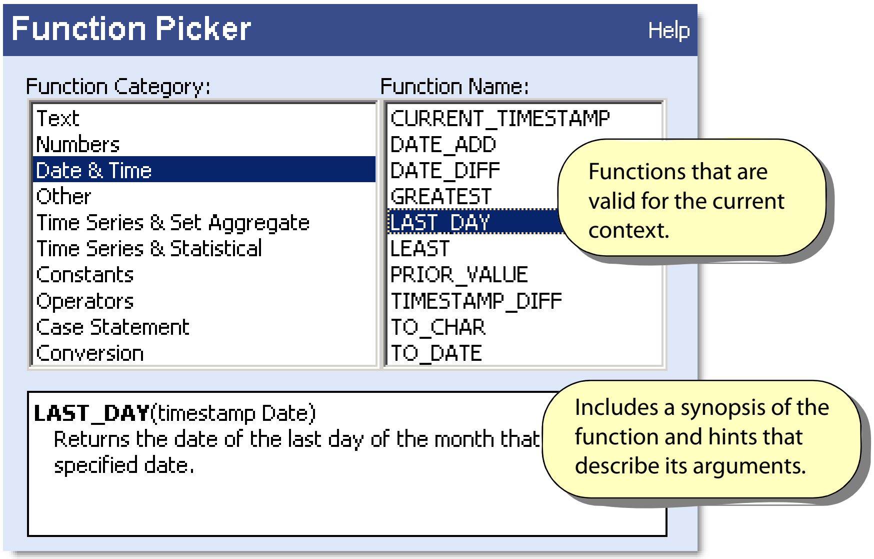
Num_Digits: the number of digits to which you want to round. Negative rounds to the left of the decimal point; If omitted, or zero, round to the nearest integer.

With the field builder you can:

- Edits formulas in a large text editor.
- Add fields from the work set by choosing them from the drop-down list.
- Insert operators by clicking on them.
- Insert functions directly into the formula.

Function Picker

The More Functions option displays the Function Picker which lists the functions that may be inserted into the current field. Each function is listed by category, and each includes a synopsis that describes the function and its arguments.



LAST_DAY(timestamp Date)
Returns the date of the last day of the month that specified date.

Functions that are valid for the current context.

Includes a synopsis of the function and hints that describe its arguments.

For detailed descriptions about the function, click **Help** on the dialog box or see "Functions" in the *Business Activity Monitoring Server Reference*.

Group By

The Group by option produces summary information for groups of rows whose the values in the selected fields are the same. Consider this set of data:

```
Name           Quantity
------------   --------
Nano Webber          10
Fizzy Lifter        700
Nano Webber          50
Nano Webber          20
Nano Webber          15
Smoke Shifter       310
```

If you create a view that groups by Name and determines the sum of the quantity for each group, it would look like the following:

```
Name: product.name              GROUP BY
Qsum: SUM(product.quantity)

Name           Qsum
------------   ----
Nano Webber      95
Fizzy Lifter    700
Smoke Shifter   310
```

You can also group on multiple fields: for example:

```
Name:  product.name             GROUP BY
Local: product.location         GROUP BY
Qsum:  SUM(product.quantity)

Name          Location Qsum
------------  -------- ----
Nano Webber   West       10
Fizzy Lifter  East      700
Nano Webber   East       85
Smoke Shifter West      310
```

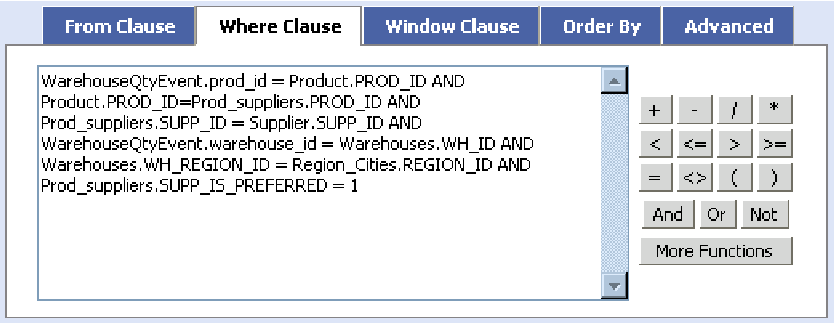
Note: When you use the Group by option, every field must either be part of the Group by or its definition must include a set function, such as SUM() or AVG().

For detailed information about Group by semantics, see "GROUP BY Clause" in the *Business Activity Monitoring Server Reference*.

Working with Clauses

The three clause fields specify the source of the data on which the view is derived, how they are joined together, under what conditions should the data be joined, and how the result is sorted.

- From clause names the source tables and views on which the view is derived.
- Where clause accepts only those rows that meet the specified condition.
- Window clause defines query windows. See "Query Windows" in the *Business Activity Monitoring Server Reference* for details.
- Order By sorts the resulting view based on column names or on expression results.
- View Persistence records view results in the database. See "Persisting View Data" on page 72 for details.



```
WarehouseQtyEvent.prod_id = Product.PROD_ID AND
Product.PROD_ID=Prod_suppliers.PROD_ID AND
Prod_suppliers.SUPP_ID = Supplier.SUPP_ID AND
WarehouseQtyEvent.warehouse_id = Warehouses.WH_ID AND
Warehouses.WH_REGION_ID = Region_Cities.REGION_ID AND
Prod_suppliers.SUPP_IS_PREFERRED = 1
```

Note: These tabs correspond to the clauses in the C-SQL SELECT statement: FROM Clause, and WHERE Clause, and Window Clause. For details, see SELECT.

From Clause

The From clause contains a list of the tables and view on which to derive the view. Every table or view referenced in the Field Definitions must also be listed here. This list usually corresponds with the table and view objects shown in the Workset.

You can also use the From clause to define an inner join relationship between the tables and views in the workset. An inner join is one where the rows in the result table are the rows from the first table that meet the specified criteria, combined with the corresponding rows from the second table that meet the specified criteria. For example:

```
Product AS P INNER JOIN Manufacturer AS M
ON P.productName = M.ProductName
```

For a detailed discussion of join relationships, see "FROM Clause" in the *Business Activity Monitoring Server Reference*.

Where Clause

The Where clause defines how to join multiple source tables and views. It also filters the resulting view to accept only those that meet the specified condition. For example, this clause specifies how to join the *Product* context to the *orderStatusEvent* event, and how to join the *Supplier* context to the *Product* context:

```
orderStatusEvent.OS_PROD_ID = Product.prod_id AND
Product.prod_supp_id = Supplier.supp_id
```

The clause could be expanded to filter out all rows whose price is greater than or equal to 10 as follows:

```
orderStatusEvent.OS_PROD_ID = Product.prod_id AND
Product.prod_supp_id = Supplier.supp_id AND
orderStatusEvent.OS_PRICE >= 10
```

Window Clause

The Window clause defines query windows that can be used in aggregations over sets of events and view rows. See Query Windows for complete details.

Order by Clause

Orders (sorts) the resulting view based on column names or on expression results. Without this clause there is no guarantee that the same query will produce rows in the same order on subsequent queries.

By default, the view is ordered in ascending order (ASC). To order in descending order, specify the DESC option.

This example has two fields (*Name* and *Quantity*) and the results are ordered by *Name* in descending order:

```
Name:     product.name
Quantity: product.quantity
Order by: product.name DESC

Name          Quantity
------------  --------
Smoke Shifter      310
Nano Webber         10
Nano Webber         50
Nano Webber         20
Nano Webber         15
Fizzy Lifter       700
```

Maintaining Events in a Stateless View

By default, a stateless view contains only rows representing the effect of the last event; rows from previous events are discarded. With this option you can specify a set of recent event rows to maintain in the view's contents on the Results tab.

Note: You cannot maintain previous stateful views; this option is only available for stateless views.

➤ **To maintain events in a stateless view:**

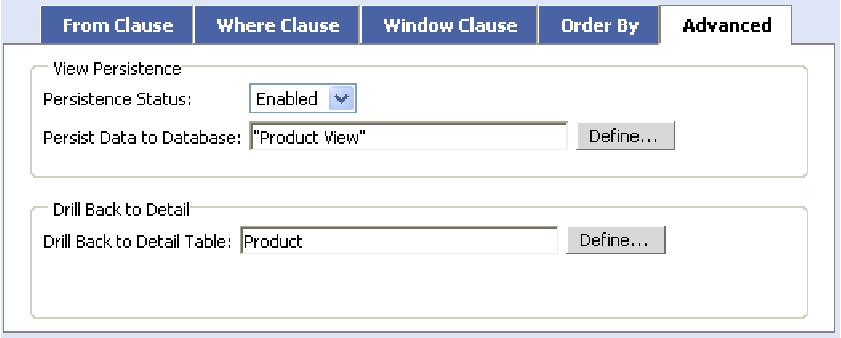
- Enter either the count or time-span of events to retain:
  - An event count is the maximum number of events to maintain that satisfies the view condition. The view discards the oldest event rows that do not fit in the specified size.
  - A time interval defines a set of the most recent events. The count of events in the view varies depending on the number of events in the interval when the view was updated. For example, if an event arrived that did not meet the view criteria, it is excluded from the view, but the view still recalculates the interval at that time.

Note: The set of events is determined *when the last event was inserted*, not at the current time. For example, an interval of one hour shows all the events that arrived in the view for the hour previous to the last update. If no events were inserted in the last day, the view might still show an hour's worth of events from the previous day. However, as soon as new event arrives at the view, all those events are discarded.

Note: See also "Moving Set Functions" in the *Business Activity Monitoring Server Reference* for a means of performing aggregations on sets of recent events.

Configuring Advanced Features

The Advanced tab allows you to configure "View Persistence" and "Drill Back to Detail" in Business Activity Monitoring.



Persistence Status: Enabled
Persist Data to Database: "Product View"
Drill Back to Detail Table: Product

Persisting View Data

Business Activity Monitoring can persist business view data to an external RDBMS for future reporting by third party tools. The information in the table is sufficient for the reporting tools to recreate a complete snapshot of the view. When persisting, the view information is written to a table in the RDBMS at a rate following a policy that you define.

Note: To use view persistence, you will first need to have a defined JDBC agent to access the RDBMS that will store the data.

For detailed instruction about using this feature, see “Persisting Views to a Database” in the *Business Activity Monitoring Server Reference*.

Drill Back to Detail

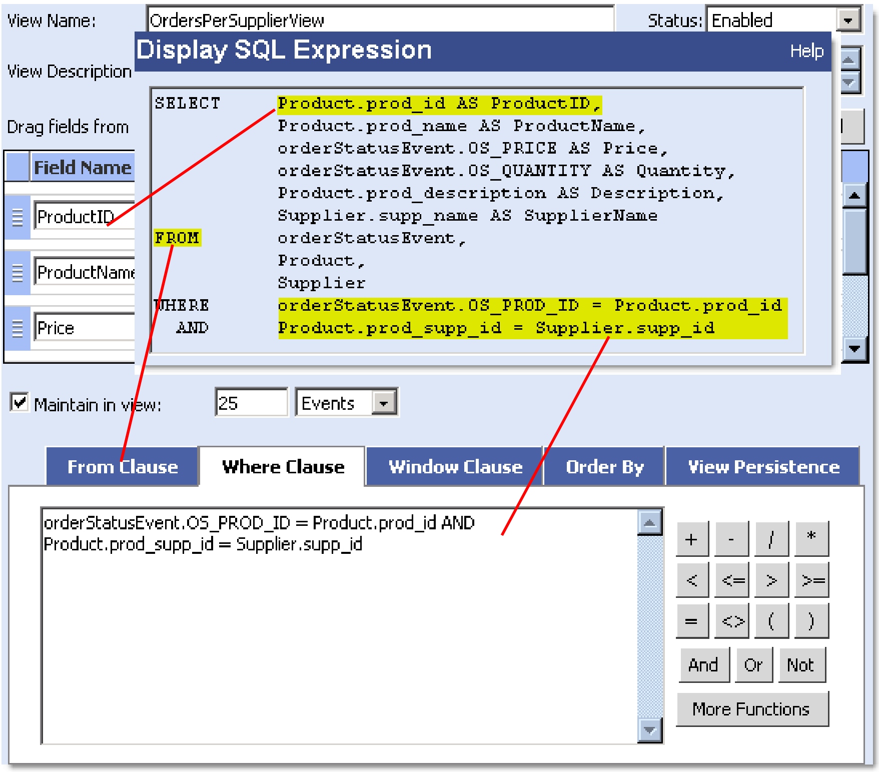
You can create views that make it possible to drill back to detail on a chart. This allows a user to see the columns in the fact table of a cube and all levels of all cube’s dimensions when viewing a dashboard object. “Drill Back to Detail” has the following requirements:

- The data source of the dashboard object must be a cube.
- The view on which the cube is built must be stateless.
- The dimensions of the cube must come from a single external database.
- If there is more than one table in the view, you can use a persisted table for the drill back to detail table.

For detailed instruction about using this feature, see “Enabling Drill Back to Detail” in the *Business Activity Monitoring Server Reference*.

Displaying SQL Expressions

You create views with the view editor in BAM Workbench; however, behind the scenes, the view is defined as an SQL SELECT statement. From the editor, you can see the SQL statement by choosing the Display SQL Expression option.



View Name: OrdersPerSupplierView
Status: Enabled

Display SQL Expression

```
SELECT      Product.prod_id AS ProductID,
            Product.prod_name AS ProductName,
            orderStatusEvent.OS_PRICE AS Price,
            orderStatusEvent.OS_QUANTITY AS Quantity,
            Product.prod_description AS Description,
            Supplier.supp_name AS SupplierName
FROM        orderStatusEvent,
            Product,
            Supplier
WHERE       orderStatusEvent.OS_PROD_ID = Product.prod_id
  AND       Product.prod_supp_id = Supplier.supp_id
```

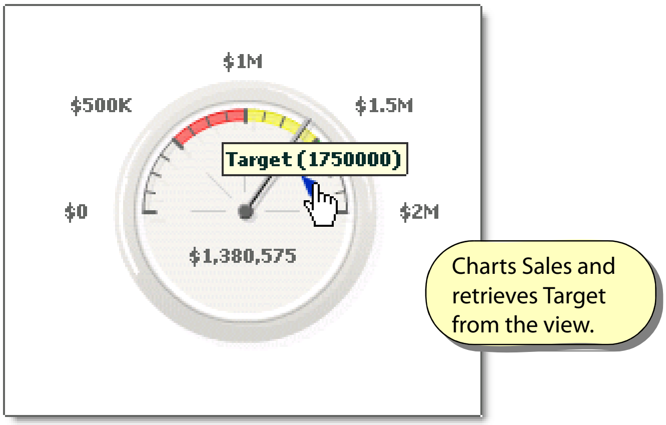
☑ Maintain in view: 25 Events

```
orderStatusEvent.OS_PROD_ID = Product.prod_id AND
Product.prod_supp_id = Supplier.supp_id
```

For a detailed description of the SELECT statement, see “SELECT” in the *Business Activity Monitoring Server Reference*.

Viewing Plan Relationships

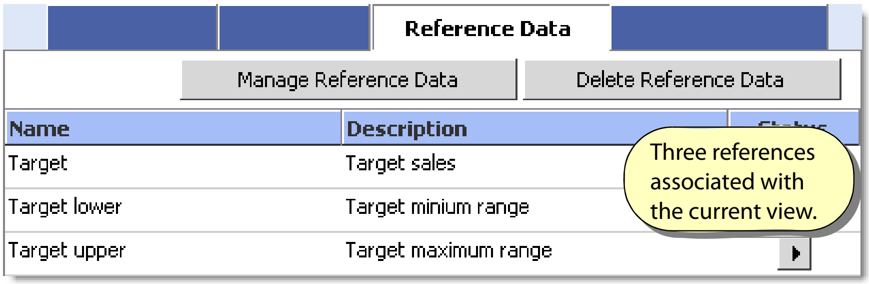
Reference data are the data sources for references in BAM Dashboard objects. For example, if a speedometer shows the current total sales, a reference line can show the target sales.



Charts Sales and retrieves Target from the view.

Note: See "Plans" in the *Business Activity Monitoring Server Reference* for details about using reference data in dashboard objects.

Before the dashboard objects can use the reference, you have to define them in the BAM Workbench. To access the references associated with a view, select the view in the Views folder and click the **Reference Data** tab. For example, the view in the following illustration has three references.



Name	Description
Target	Target sales
Target lower	Target minium range
Target upper	Target maximum range

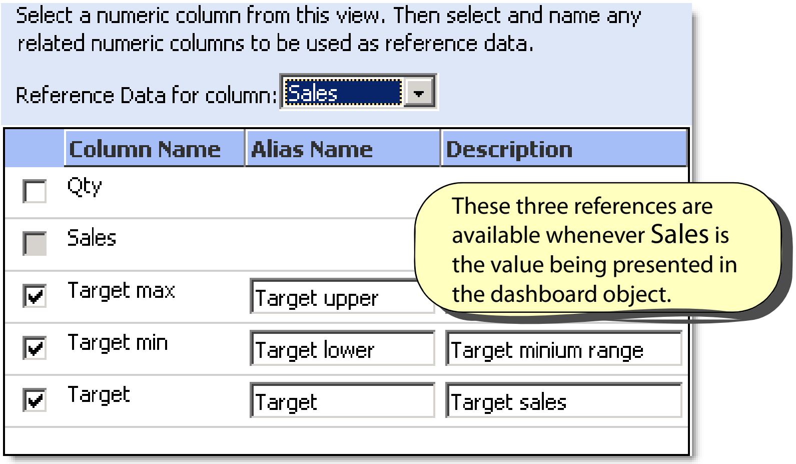
Three references associated with the current view.

➤ **To create and edit references:**

- Click **Manage Reference Data**.

Reference Data Source

Each reference is a numeric column in the same view as the source for the dashboard object. Use Manage Reference Data to associate references with the column that will be presented in the dashboard object. For example, if a speedometer charts the value in the view's Sales column, and the Target column contains the sales plan (target), create a reference to Target and associate it with Sales. The following illustration show three references associated with the Sales column:



Select a numeric column from this view. Then select and name any related numeric columns to be used as reference data.

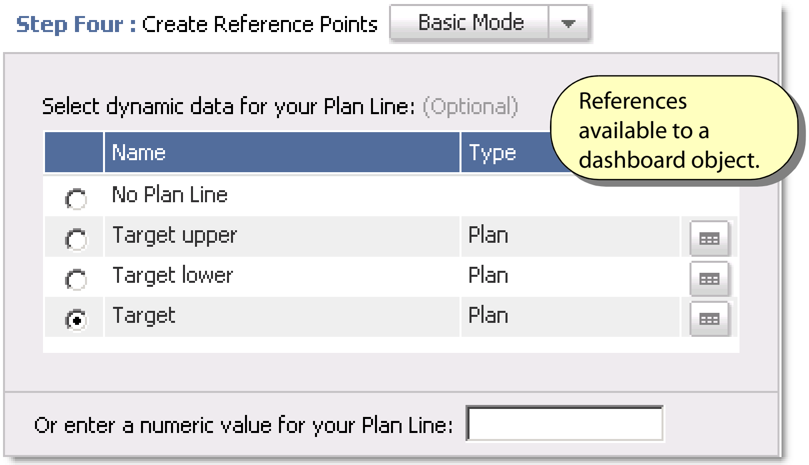
Reference Data for column: Sales

	Column Name	Alias Name	Description
□	Qty		
□	Sales		
☑	Target max	Target upper	
☑	Target min	Target lower	Target minium range
☑	Target	Target	Target sales

These three references are available whenever Sales is the value being presented in the dashboard object.

Note: A column may be associated as a reference to multiple presentation columns; however, the Alias Name must be unique within the view.

Later, when you create or edit the dashboard object, and when the object measures the Sales column, these references will be available to the object.



Step Four : Create Reference Points

Select dynamic data for your Plan Line: (Optional)

	Name	Type
○	No Plan Line	
○	Target upper	Plan
○	Target lower	Plan
●	Target	Plan

Or enter a numeric value for your Plan Line:

References available to a dashboard object.

Working with Cubes and Dimensions

A *cube* is a set of data organized by dimensions and measures for the purpose of aggregating different subsets of the larger set of data. When rendered as a Dashboard Object, cubes allow you to quickly choose categories that "filter" data to show the results that meet your selection. For example, a cube of "sales" data might provide aggregations of the same data by product, by time, or by sales region dimensions. Looking at the cube you might choose to view the total sales of a product (Nails) within a business region (West) during a fiscal quarter (Q1):

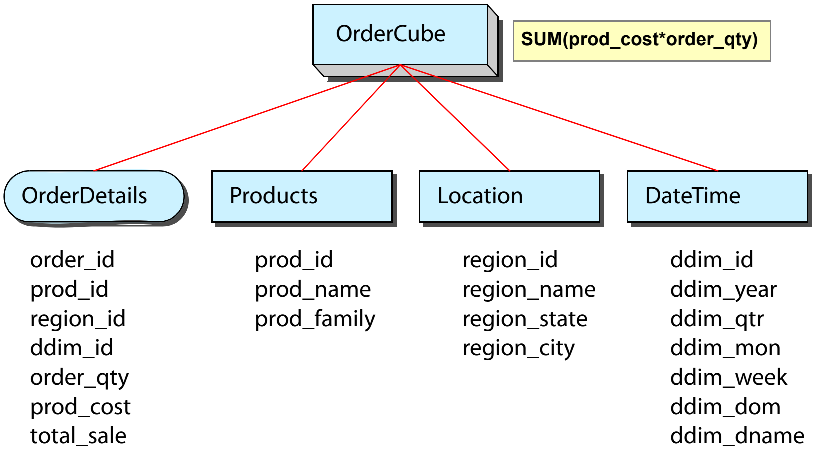
Cubes are similar to business views in that they aggregate event data, but they do so across different dimensions. The view that a cube aggregates is a *fact table*: a view or event table in an event stream that contains one or more columns to measure (aggregate), and which also contains columns that identify the dimensional elements associated with the event. For example, you could imagine a fact table containing an event similar to this:

```
Cost   Quantity Product State      Month
------ -------- ------- ---------- -------
200.00     1600 Nails   California January
```

However, in practice the dimensional elements are stored in dimensions (special context tables) and referenced by IDs, like this:

```
Cost   Quantity prod_id region_id ddim_id
------ -------- ------- --------- -------
200.00     1600     100         7      39
```

Similar to a view, as new event enter the system they become part of an event stream which can feed a cube. As new events enter a cube, each measure of each dimension is recalculated to provide the latest aggregation. The following illustration shows a cube built from the OrderDetails fact table and which measures total sales across various business regions, products, and time:



➤ To define a cube:

1. Select the fact table and columns to measure.
2. Specify how to measure them (aggregate formulas to use).
3. Select one or more dimensions that classify the measurements.
   - *Measures* are the central value that are aggregated and analyzed. Measure columns define the aggregations. A cube must have at least one measure column and may have more. Each measure column defines an expression that contains a C-SQL set function that aggregates other columns from the fact table. For example, to determine the “total sales” from the OrderDetails fact table, a measure column might be defined as:

     ```
     SUM(OrderDetails.prod_cost*OrderDetails.order_qty) AS TotalOrderSales
     ```

   - A *dimension* is a ranked order of classifications that, from highest to lowest level each describe smaller, more distinct sets of related data. For example a business region is one level of a geographical dimension, the quarter and month columns can each be levels of a time dimension, and a product family and product name part of an inventory dimension. In a time dimension, months are smaller sets of fiscal quarters, just as product name is a smaller set of the product family level. The following are some examples of dimensions:

Time	Geography	Inventory	Security	Taxonomy
year	continent	classification	type	kingdom
quarter	country	type	rating	phylum
month	region	manufacturer	company	class
week	state	model	cusip	order
day	county	configuration		family
hour	city			genus
minute	district			species

Dimension columns categorize the measurements. A cube must have at least one dimension, and may have more. Further, the data in the fact table must be able to identify a unique element in each associated dimension. For a complete discussion, see “Dimensions” in the *Business Activity Monitoring Server Reference*. (Note that while the mathematical term “cube” implies three dimensions, a database cube can have any number from one or more.)

Creating Dimensions

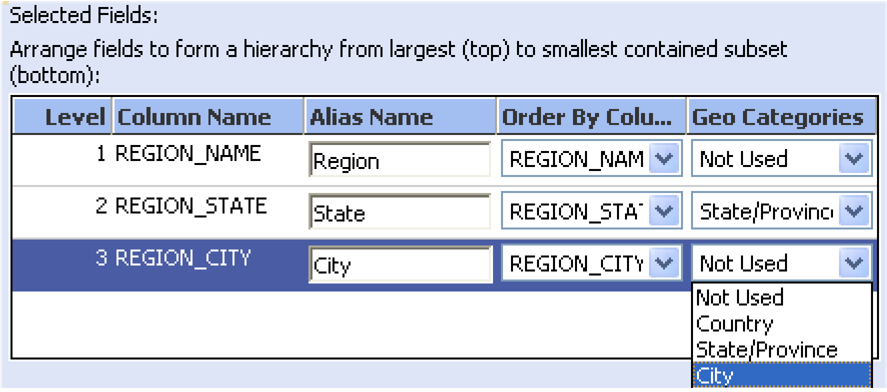
Before creating a dimension, you must have:

- Create permission for Views, Cubes, and Dimensions
- At least Read-Only access to the Context table that provides the dimension elements

Note: Turn on caching for the context table for optimum performance. When caching is off, performance for cubes can be slowed dramatically. See "Caching Context Queries" in the *Business Activity Monitoring Server Reference* or "Working with Data Caching" on page 26 for information about controlling the cache.

➤ To create a dimension:

1. Open the Workbench tab of BAM Workbench, select the Dimensions folder, and click **New Dimension**.
2. Identify a name, and optionally provide a description of the dimension.
3. Select the **Context Table** that contains the dimension elements.
4. (Optional) Select "Use this for Geography Charts" if you plan to implement geographical maps. This displays a Geo Categories column to the Selected Fields list.
5. Define the levels of the hierarchy:
   - Add fields to the hierarchy from the Available Fields list.
   - Order the levels from largest set (top) to smallest (bottom).
   - Identify one or more Key Columns to include.
   - Optionally assign alias names to the levels.
   - Optionally assign Geo Categories to each level, if you selected the "Use this for Geography Charts" check box, the Selected Fields list displays a Geo Categories column. From this column, you can assign Not Used, County, State/Province, City, Street, or Postal Code geo category to a level in the hierarchy.



Selected Fields:
Arrange fields to form a hierarchy from largest (top) to smallest contained subset (bottom):

Level	Column Name	Alias Name	Order By Colu...	Geo Categories
1	REGION_NAME	Region	REGION_NAM	Not Used
2	REGION_STATE	State	REGION_STA	State/Provinc
3	REGION_CITY	City	REGION_CITY	Not Used

Save the dimension and you can immediately use it in cubes.

Creating Cubes

Before creating a cube, you must have:

- Create permission for Views, Cubes, and Dimensions.

- At least Read-Only access to an existing fact table (business view).
- At least Read-Only access to the dimensions to include. See "Creating Dimensions" on page 78 for details.

➤ To create a cube:

1. Open the **Workbench** tab of BAM Workbench, select the **Cubes** folder, and click **New Cube**.

2. Identify a name, and optionally provide a description of the cube.

3. Select the Fact Table that contains the data to measure, and which contains columns that identify the dimension elements.

4. Define one or more Measure columns.

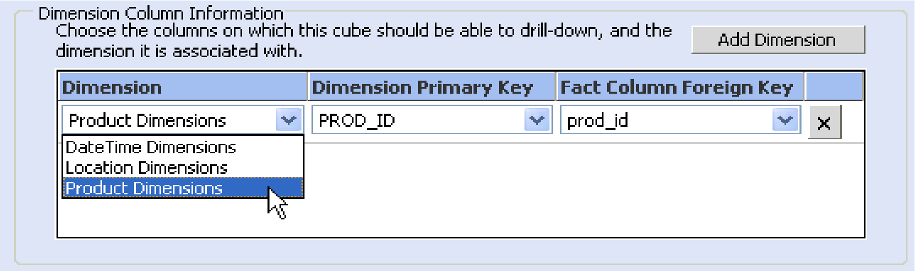
   - Click **Add Measure Column** to define a column.
   - Name the column in the Measure Name field.
   - Define the measure formula with a C-SQL set function in the Aggregate Expression field. The function should reference a column from the fact table. For example, this SUM() expression totals the product of the cost and quantity columns):

     ```
     SUM(OrderDetails.prod_cost*OrderDetails.order_qty)
     ```

5. Define one or more Dimension columns.

   - Click **Add Dimension** to define a column.
   - Select the dimension to include from the Dimension column drop-down list.

     This list includes all dimensions that you have at least Read-Only access to.



Dimension Column Information
Choose the columns on which this cube should be able to drill-down, and the dimension it is associated with.

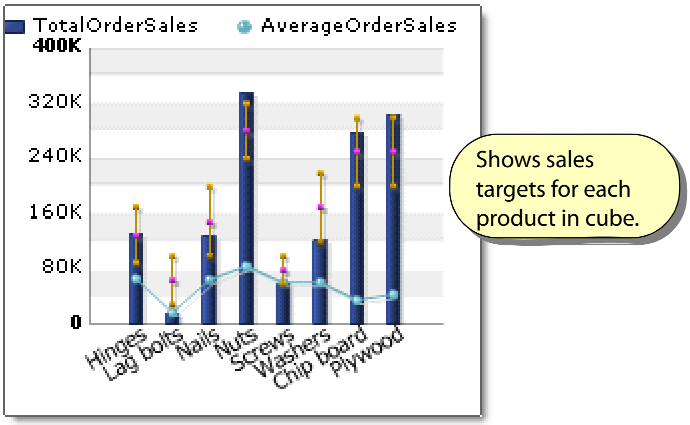
Dimension	Dimension Primary Key	Fact Column Foreign Key	
Product Dimensions	PROD_ID	prod_id	✕

   - Identify the key columns in the dimension and in the fact table. See "Dimensions" in the *Business Activity Monitoring Server Reference* for more details about the keys.
   - The data type for the key in the fact table must be the same for the key in the dimension (context table). You cannot, for example, mix integer and decimal types; both must be either integer or decimal.

6. Save the cube and you can immediately begin building Dashboard Objects on top of it.

Cube Plan Relationships

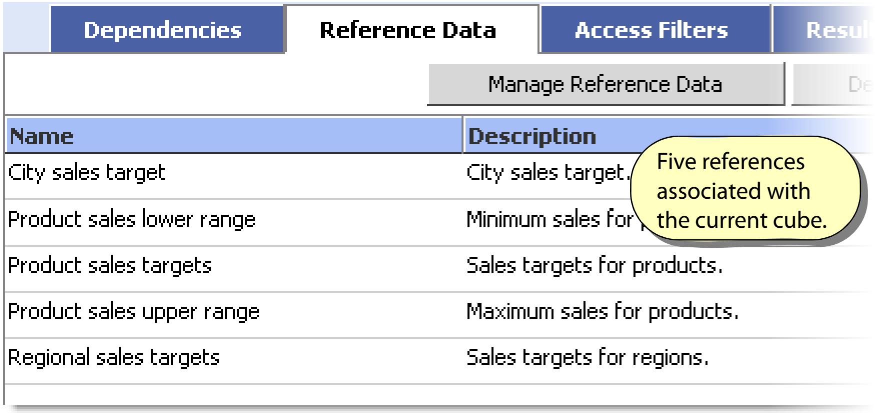
Reference data are the data sources for references in BAM Dashboard objects. For example, if a bar chart measures total sales, a reference line can show the target sales for each category, while a range shows the sales target range for the same.



Shows sales targets for each product in cube.

Note: See Plans for details about using reference data in dashboard objects.

Before the dashboard objects can use the reference, you have to define them in BAM Workbench. To access the references associated with a cube, select the cube in the Cubes folder and click the **Reference Data** tab. For example, in the following illustration, this cube has five references.



Name	Description
City sales target	City sales target.
Product sales lower range	Minimum sales for
Product sales targets	Sales targets for products.
Product sales upper range	Maximum sales for products.
Regional sales targets	Sales targets for regions.

Five references associated with the current cube.

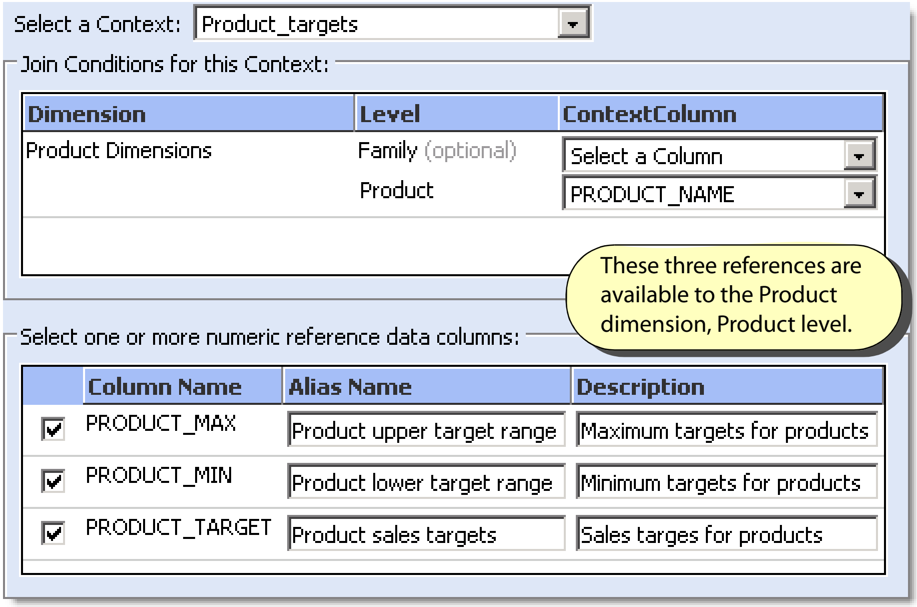
To create and edit references,

- Click **Manage Reference Data**.

  Each reference is a column in a context table that is related to a column in the cube. For example, to have the sales targets for a product, there needs to be a product name column in the cube, and it needs to map to a product name column in the context table that contains the sales targets for each product. The following is an example of some of ten data from the Product_Targets context table:

PRODUCT_NAME	PRODUCT_MIN	PRODUCT_TARGET	PRODUCT_MAX
Nails	100000.00	150000.00	200000.00
Screws	60000.00	80000.00	100000.00
Lag bolts	30000.00	65000.00	100000.00
Washers	120000.00	170000.00	220000.00

Further, each reference is associated with a measure in the cube (such as total sales), and is available to a single level of a single dimension of the cube. For example, the data above is only applicable to the Product Dimension, Product level of a cube.



Select a Context: Product_targets

Join Conditions for this Context:

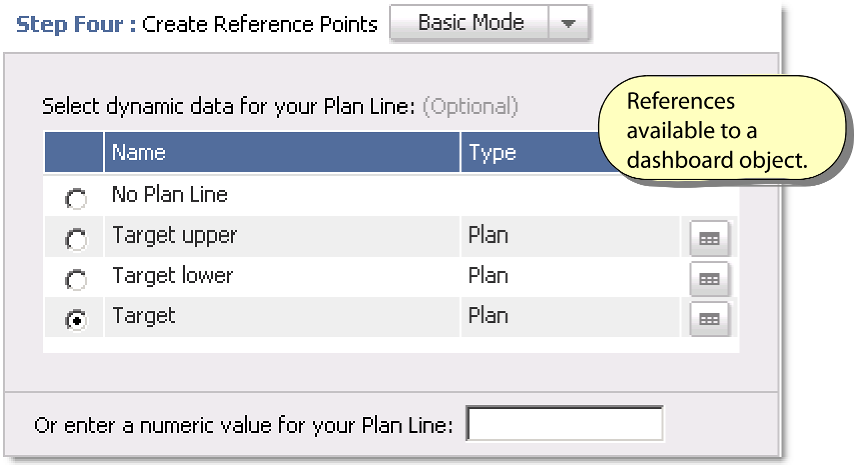
Dimension	Level	ContextColumn
Product Dimensions	Family (optional)	Select a Column
	Product	PRODUCT_NAME

Select one or more numeric reference data columns:

	Column Name	Alias Name	Description
☑	PRODUCT_MAX	Product upper target range	Maximum targets for products
☑	PRODUCT_MIN	Product lower target range	Minimum targets for products
☑	PRODUCT_TARGET	Product sales targets	Sales targes for products

These three references are available to the Product dimension, Product level.

Later, when you create or edit the dashboard object, and when the object measures the Total_Sales column, these references will be available to the object.



Step Four : Create Reference Points

Select dynamic data for your Plan Line: (Optional)

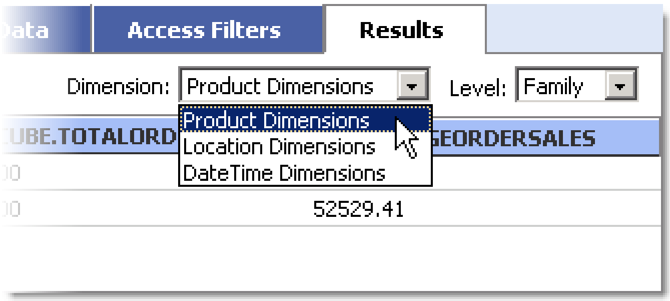
	Name	Type
○	No Plan Line	
○	Target upper	Plan
○	Target lower	Plan
●	Target	Plan

Or enter a numeric value for your Plan Line:

References available to a dashboard object.

Checking Cube Results

To see the contents of a cube, use the Results tab. The Dimension and Level controls determine the cube-face to view.



If the cube is empty, or if you do not have access to see the data in a dimension or level, you see the "None available" message instead.

With the BAM Dashboard, you can chart the contents of a cube or display values in a table.

Working with UDFs and JAR Files

Application developers may create user-defined functions (UDFs) for use in queries, views, and rules. For example, you might have a UDF that takes a set of values and concatenates them alphabetically while ignoring NULL values. In a Field expression, you would use that UDF in an expression such as the following:

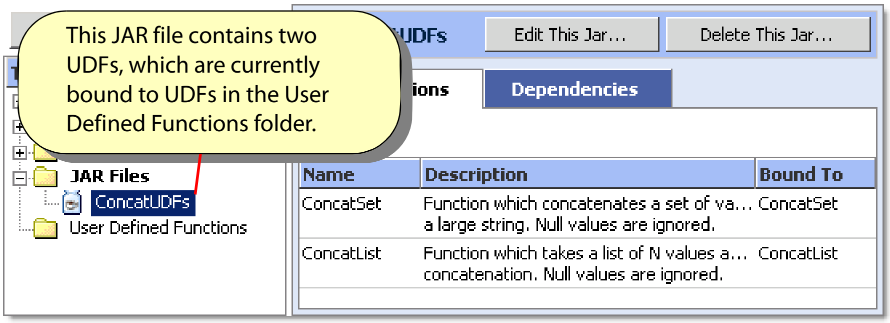
```
ConcatSet(Product.Name)
```

UDFs are small Java programs that take arguments and return a value, just like the internal Business Activity Monitoring functions. See "User-Defined Functions" in the *Business Activity Monitoring Server Reference* for complete details on implementing UDFs.

The UDF Java programs are stored in JAR files. To access the UDFs, you need to upload the JAR file to Business Activity Monitoring servers, then identify the functions to make them available.

JAR Files Folder

JAR Files contain one or more Java programs. The JAR Files folder provides access to the JAR files available to Business Activity Monitoring. Select the folder to see a list of the JAR files already installed, and to upload new JAR files. Select a JAR file in the folder to see which functions that JAR contains and which objects depend on the functions in that JAR.



This JAR file contains two UDFs, which are currently bound to UDFs in the User Defined Functions folder.

Name	Description	Bound To
ConcatSet	Function which concatenates a set of va... a large string. Null values are ignored.	ConcatSet
ConcatList	Function which takes a list of N values a... concatenation. Null values are ignored.	ConcatList

➤ To upload a JAR file:

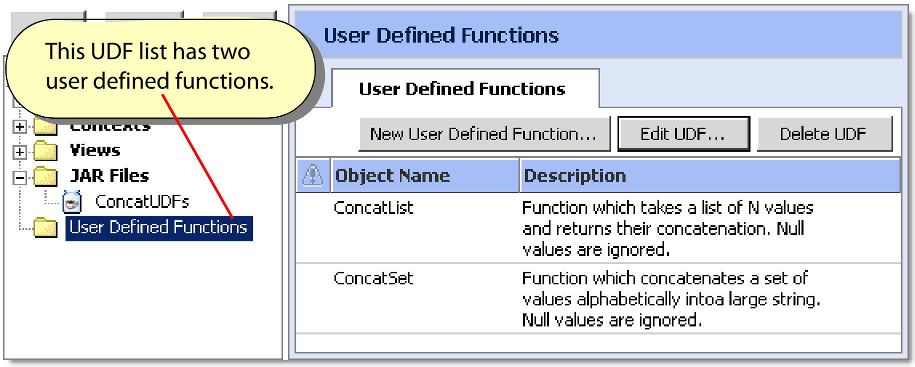
1. Select the **JAR Files** folder and select **New JAR File**.

2. Assign a name, such as ConcatUDFs in the figure above, and optional provide a description. Locate the file in the Path field and select Finish.

The JAR file is now available to the system.

User Defined Functions Folder

The User Defined Functions folder controls access to the UDFs available to Business Activity Monitoring. Select the folder to see the list of functions available, to upload or identify new functions, or to edit or delete functions already in the list.



This UDF list has two user defined functions.

Object Name	Description
ConcatList	Function which takes a list of N values and returns their concatenation. Null values are ignored.
ConcatSet	Function which concatenates a set of values alphabetically intoa large string. Null values are ignored.

User Defined Functions (UDFs) are Java programs stored in JAR files. To make a UDF available to the system, you must first upload the JAR file that contains it. You can either follow the steps listed in "JAR Files Folder" on page 83, or choose to upload it when you create a new UDF in the list.

➤ To add a UDF to the list:

1. Select the **User Defined Functions** folder and select **New User Defined Functio**n.

2. The Create UDF dialog prompts you to either select one of the JAR files already in the system or to upload a new JAR file.

   If you upload a new file, it will appear in the JAR Files folder after you complete these steps.

3. After identifying the JAR file, select the UDFs that you want to include in the folder.

   By default, all UDFs in the JAR are selected. Further, each UDF is assigned a name the same as it appears in the JAR file; however, you may assign another name. The name is what appears in the Bound To column in the JAR Files Folder list.

4. Click **Finish** and the UDFs are immediately available to the system.

Working with External Links

External links are URLs that locate external systems. They can then be used to create associated reports that are inserted into reportlets for communicating to external report mechanisms.

Property	Description
Name	Name of the external link object. This is the name that appears in the list of links when creating an external reportlet or associated report.
Description	Text description of this object.
URL Link	Complete URL for locating the external system. If you do not assign a value for this field, the system assumes an HTTP connection to port 80 on the same domain as the application.

The URL is the string that locates the reporting system. Later, when the reportlet is generated, it extends the URL with data parameters to pass to the report system. Note that the URL must contain the http:// qualifier. For example,

```
http://reports:8091
```

Later, the reportlet might expand the URL similar to the following:

```
http://reports:8091?ProdDescrPDF=MyReprt&product=prod_id
```

See “Working with Reportlets” on page 54 for information about creating reportlets in BAM Workbench and “Reportlets” in the *Business Activity Monitoring Server Reference* for detailed information about reportlets.

➤ To create an external link:

1. Log in to BAM Workbench.
2. Open the Application Workbench tab.
3. Click on **Extenral Links** in the left column.

   The right panel redisplays to show the External Links tab pane.
4. Click the **New External Link** button.
5. Name and describe the link, and define the URL link.
6. Click OK.

   The new link is immediately available for reportlets and associated reports.

Working with Process Definitions

Process definition objects associate a process definition file with an aggregate view or cube, and optionally with a search context table. BAM Dashboard uses the object when rendering a process diagram to identify the associated aggregate source for statistics, and for identifying the search context table and search label to show the user.

Note: For a complete discussion about processes and how they work in Business Activity Monitoring, see "Processes" in the *Business Activity Monitoring Server Reference*.

Property	**Description**
Name	Name of the process definition object. This is the name that appears in the list of processes when creating process diagram in the BAM Dashboard.
Description	Text description of this object.
Process (process definition file)	Process identified in the uploaded process definition file. The process definition file is generated by the external process management system.
Aggregate View or Cube	View or cube that aggregates process instance data. One of the columns identifies the steps in the process, such as a step name or ID.
Detail Search	(Optional) A context table that queries the external process management system about a specific process. Users viewing a process diagram in the BAM Dashboard can enter an ID of a specific process instance and receive the information about the instance as reported in the context table.

Creating Process Definitions

Before creating a process definition, you must have:

- Read Only access permission on the view or cube that aggregates the process events.
- (Optional) Read Only access permission on the context search table.

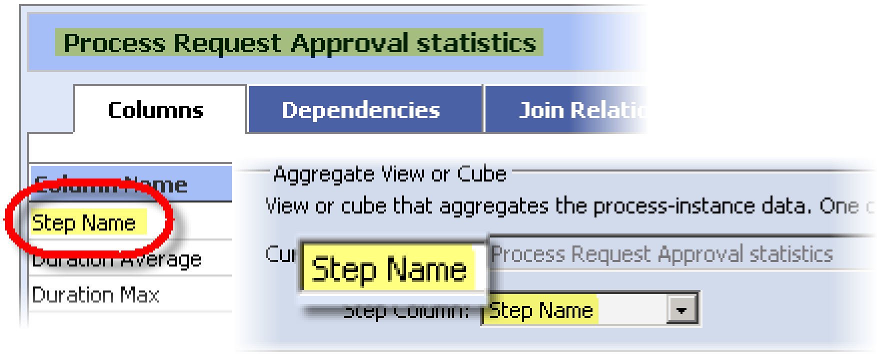
➤ **To create a process definition:**

1. Open the Process Definitions folder in the BAM Workbench and select **New Process Definition**.
2. Name the object and upload the process definition file generated by the BPM.

   The process name from the definition file appears in the Process Name field after uploading the file.

3. Select the aggregate view or cube, and select the column that identifies the steps in the process.

   For example, each step in the process might be identified by a name or ID number. The illustration below shows the Step Name column in the Process Request Approval statistics view.

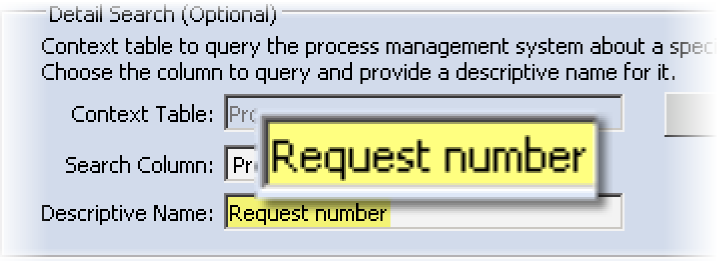


Process Request Approval statistics

Aggregate View or Cube
View or cube that aggregates the process-instance data. One c
Cur Process Request Approval statistics
Step Column: Step Name

4. (Optional) Select the detail search context table, select the column to search, and optionally provide descriptive text to appear in the BAM Dashboard.

   The illustration below shows how the descriptive name appears in the search dialog:



Detail Search (Optional)
Context table to query the process management system about a spec
Choose the column to query and provide a descriptive name for it.
Context Table: Prc
Search Column: Pr
Descriptive Name: Request number

5. Save the process definition.

You can now create process diagrams based on this definition.

Creating Associated Reports

An Associated Report links a View to an external reporting system through a previously defined external link. External links are URLs. The Associated Report configuration includes specifying parameters in the URL that can be substituted with values drawn from columns in the View definition.

After configuration, Associated Reports can be accessed from the Activities submenu in BAM Dashboard for any Dashboard Object that is based on the View defined in the Associated Report configuration.

> **Note:** For more information on using Associated Reports, see the BAM Dashboard documentation.

➤ To create an Associated Report:

1. Log in to BAM Workbench.

2. Open the Application Workbench tab.

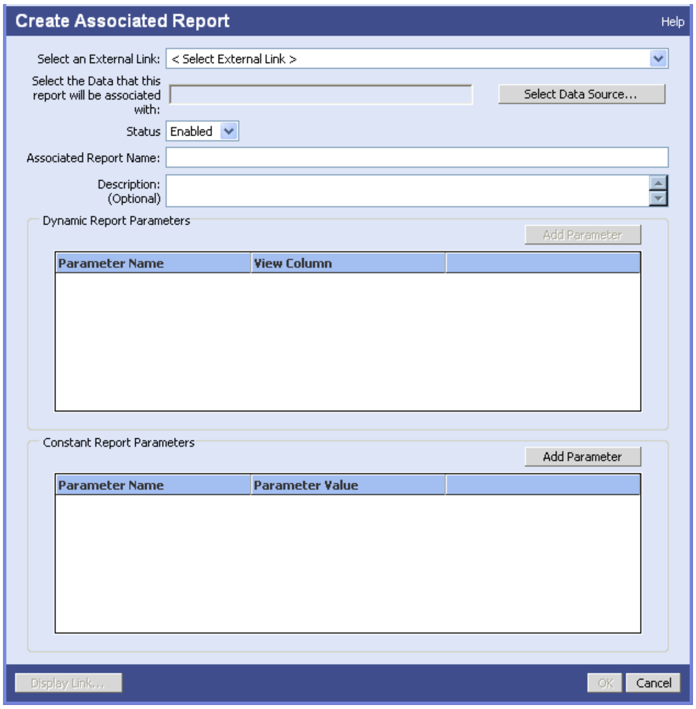
3. Create an external link for the associated report, as described in “Working with External Links” on page 84.

4. Click on **Associated Reports** in the left column.

   The right panel redisplays to show the Associated Reports tab pane.

5. Click the **Create Report** button.

   The Create Associated Report dialog box displays.



Create Associated Report

Select an External Link: < Select External Link >

Select the Data that this report will be associated with:

Status: Enabled

Associated Report Name:

Description: (Optional)

Dynamic Report Parameters

Parameter Name	View Column	

Constant Report Parameters

Parameter Name	Parameter Value	

6. This dialog contains the following settings:

Property	**Description**
Selected an External Link	This dropdown menu lists all the external links that have been configured for BAM.
Select the Data...	Click the Select Data Source button to open a secondary dialog box that lists the configured Views with which a report may be associated.
Status	Indicates whether the report is available or not when a dashboard is creating with the specified View.
Associated Report Name	The name of the report as it will appear in the Associated Reports tab pane in BAM Workbench and in the Activities list when creating dashboard objects in BAM Dashboard.
Description	Text description of this object.
Report Name	Enter a valid business report name.

Property	**Description**
Dynamic Report Parameters	This panel specifies the column from which values can be assigned to the variable parameters in the URL specified in the external link. For example, consider a case in which the external link defines a URL for a Google search but the parameter defining the search term is missing: http://www.google.com/search?hl=en&btnG=Google+Search In the Report Parameters, you would add a parameter with a Parameter name of “q” (the Google parameter for a search term); then, from the View Column, you select from a list the column from the View that contains the potential terms for the search. Thus, when the Associated Report is called from the dashboard object based on the same View, the values currently contained in the column are displayed in a list from which the user can choose. Thus, at runtime, the actual URL request is constructed as follows: http://www.google.com/search?hl=en&btnG=Google+Search &**q=Coyotebreath Perfume** The Associated Report configuration appends the “&q=Coyotebreath Perfume” string, which represents the Parameter (q) and View Column (ProductName). **Note:** You can create multiple parameters, as the search criteria allows.
Constant Report Arguments	This panel specifies argument values that can be assigned to arguments in the URL specified in the external link. For example, consider the case in which the external link defines a base URL for a Cognos report, and the base URL needs the following arguments: **ui.tool**=CognosView **ui.object**=contect/folder[@name=’Demo Report’]/report[@name=’Inventory Levels by Retailers’] **ui.action**=run **run.outputFormat**=PDF The base URL is http//server-name/cognos8/cgi-bin/cognos.cgi?b_cation=xts.run&m=portal/launch.xts& When the Associated Report is called from the dashboard object, the values specified for the arguments in the Report Arguments panel, the actual URL request is constructed as follows: http//server-name/cognos8/cgi-bin/cognos.cgi?b_cation=xts.run&m=portal/launch.xts&**ui.tool**-CognosView**&** **ui.object**=contect/folder[@name=’Demo Report’]/report[@name=’Inventory Levels by Retailers’]&**ui.action**=run&**run.outputFormat**=PDF

7. Complete the settings as desired and click **OK**.

   The dialog box close and the new Associated Report is now listed on the Associated Reports tab pane.

5 Managing System Administration

All system administration tasks are performed on the Administration Console. This chapter describes the configurations that you can define and change, and the activities you can perform with the Administration Console.

➤ **In this Chapter:**

Managing Users

Each user that interacts with Business Activity Monitoring is *known* to the system by their account information. The Users list in the Administration Console provides access to the user accounts. From this list you can create, view, edit, or delete user accounts.

> **Note:** You can also view and edit your own account information by clicking Account Settings in the upper right of the application.

For detailed information about user accounts, see "Users" in the *Business Activity Monitoring Server Reference*.

➤ **To create, edit, or delete a user account:**

1. Open the Administration Console.
2. Click **Users** to see a list of all users currently defined in the system. To
   - Create a new user account, click **New User** and define the user's account information. For details about the information, see "Users" in the *Business Activity Monitoring Server Reference*.
   - Edit an existing user's account information by double-clicking the name in the list.
   - Delete an existing user by selecting one or more users in the list and clicking Delete Users. Note that you cannot delete the System User or yourself.

Permission Requirements

The user accounts in the list are the ones for which you have Read Only or Read and Write permission; you do not see those users for whom you have No Access. Further,

- To create a new user, you need Creating Permission for user objects.
- To edit or delete a user's account information, you need Read and Write permission for that user account.

For more information about permissions, see "Permissions" in the *Business Activity Monitoring Server Reference*.

Managing Roles

The Roles list displays the current roles in the system, and provides access for creating, editing, and deleting roles.

Roles define the minimum sets of Permissions associated with Users. Roles provide a way to quickly assign *the same permissions* to an object or class of objects, for sets of users without having to set those permissions for each individual user of the set. For a detailed discussion about roles, see "Roles" in the *Business Activity Monitoring Server Reference*.

Note: For information about permissions, see "Permissions" in the *Business Activity Monitoring Server Reference*.

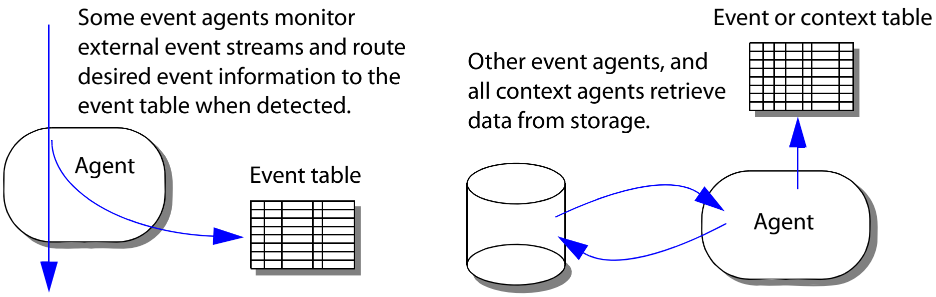
➤ To administer roles:

1. In the Administration Console, click on **Roles** to see a list of the roles in the system.
2. Create roles by clicking **New Role** and defining the role attributes. For details about role attributes, see Role Attributes.
3. Modify a role by double-clicking on the role's name in the Roles list.
4. Delete roles by selecting one or more roles in the Roles list and clicking Delete Role.
5. To see which roles a user belongs to, use the user account editor described in the "Managing Users" on page 93.
6. To see which roles you belong to, view your user account by clicking Account Settings.

Note: See "Users" in the *Business Activity Monitoring Server Reference* for more information about user accounts.

Viewing the Agents List

Agents are Business Activity Monitoring objects that know how to receive or retrieve information from external sources, such as external event streams, DBMS, or file storage systems. When an agent detects or locates desired information, it inserts the data into an event or context table for use by the view system.


Some event agents monitor external event streams and route desired event information to the event table when detected.
Agent
Event table
Event or context table
Other event agents, and all context agents retrieve data from storage.
Agent

In the Administration Console, click **Agents** to see a list of the defined agents. From this list, you can perform the following tasks:

- Create agents by clicking **New Agent** and defining the agent attributes.

  Note: For details about agent attributes, see "Agents" in the *Business Activity Monitoring Server Reference*.

- Modify an agent by double-clicking its name in the list.
- Delete agents by selecting one or more of them in the list and clicking **Delete Agents**.
- Disable or re-enable agents by clicking their status indicator.

  Note: For a detailed discussion about agents and about specific agent sources, see "Agents" in the *Business Activity Monitoring Server Reference*.

Viewing the External Processes List

External processes are external Web service methods that receive specific alert messages when initiated by a user viewing the message in the BAM Dashboard. Business Activity Monitoring publishes the alert message, and any attached reportlets as an XML document to the service. See "Web Service External Processes" in the *Business Activity Monitoring Server Reference* for complete details.

In the Administration Console, the External Processes list provides access to the processes. From this list you can perform the following tasks:

- Create processes by clicking New Process and defining the process attributes. For details about the attributes, see "External Process Attributes" in the *Business Activity Monitoring Server Reference*.
- Modify a process by double-clicking its name in the list.
- Delete processes by selecting one or more of them in the list and clicking Delete Processes.
- Disable or re-enable processes by clicking their status indicator.

Working with System Settings

System settings affect how Business Activity Monitoring is configured for normal operation.

➤ To change system settings:

1. Open the **Administration** tab.

2. Click **Edit System Settings**.

3. Choose the tab of the settings to alter.

   Each tab's settings are described in the following sections:

   - "Working with Checkpoint and Recovery" on page 97 describes the settings that enable you to control the system checkpoint and recovery mechanism.
   - "Working with Custom Branding" on page 98 identifies the logo images that appear on various pages throughout Business Activity Monitoring.
   - "Working with Encryption Configuration" on page 99 describes how keystores are used to encrypt user and administration passwords.
   - "Working with Execution Throttling" on page 100 describes the settings that monitor and adjust the rate that events are published and propagated into the view system.
   - "Working with Geography Maps" on page 101 describes the settings for configuring Geography Maps.
   - "Working with the LDAP" on page 101 describes the settings for configuring LDAP user authentication.
   - "Setting Logging" on page 103 describes the settings for how the system logs messages, and which messages to log.
   - "Setting Logging Levels" on page 104 describes the settings for which messages to log, and which system components to generate the messages.
   - "Configuring Memory Monitoring" on page 105 configures how the system behaves when the amount of virtual memory on the server becomes low.
   - "Working with Single Sign-on Settings" on page 106 allows users to bypass the login pages on Business Activity Monitoring applications.
   - "Viewing SMTP Configuration" on page 107 is the e-mail transport that the system uses to deliver e-mail notifications.
   - "Setting Alert/Notifications Limitations" on page 107 shows you how to set limitation to number of alerts that can be sent in a specified time interval.
   - "Viewing System Controls" on page 108 shows you how to administer the operation of the installation.
   - "Viewing Thread Configurations" on page 109 shows you how to change Business Activity Monitoring processes information internally.
   - "Viewing User Interface" on page 109 settings that apply to the user interface seen by all users.

4. Save the changes by clicking Apply Changes.

The changes take effect immediately after being saved.

Working with Checkpoint and Recovery

A checkpoint is a "snapshot" of the state of the system, similar to a backup. A recovery log tracks event data that have entered the system since the last checkpoint. (The recovery helps to rebuild the system's state in the event of an abnormal shutdown: one where the normal shutdown checkpoint was unable to run or complete.) Together these features are used to restore the system's state when the system restarts.

What is Saved

The snapshot records the state of all objects defined in memory except disabled or invalid objects. Every object in an event stream is consistent to the same event.

The recovery log tracks new event details that occurred after the checkpoint started, though only for those events for which recovery logging is enabled with the Log event data for recovery option (see the section "Event Properties" in the Chapter "Events" in the *Business Activity Monitoring Server Reference* for details). Every time a new checkpoint starts, a new recovery log is created, and the old log is purged once the snapshot finishes.

What is Recovered

When the system restarts, and the Recover State on Restart option of the Viewing System Controls is *on*, the system restores its state consistent to the last checkpoint. If a recovery log exists, the system also "replays" the logged events, possibly raising, lowering, and firing events as it does so.

> **Note:** The replay restores the system to a state consistent with the current context data. If dependant context data have changed since the last checkpoint, the new data are used. Similarly, if the context data are missing, views might not match their state prior to the checkpoint.

Time-based moving windows are restored to the state at the *time* of the snapshot and recovery. As such, it is possible for a restored view to contain "stale" information. For example, if a window tracks events for the last hour, the restored view contains the data from the last hour when the snapshot and recovery were recorded, not the current system time. However, the next event that updates the view discards all old information.

What is Not Saved

Context data in the data cache are not saved. After a restart, the context cache is empty.

Alerts with pending Holds For conditions that have not been met. For example, if an event met a rule condition, but the associated alert had a Holds for condition that had not been met (such as 1 hour left on a 2-hour hold) that state is not saved and the pending alert is discarded when the system shuts down.

Disabled or invalid objects do not have a state and as such, are not saved.

Error Conditions

When schedules overlap, the subsequent checkpoints are ignored. For example, if one schedule that is set to run hourly overlaps with one set to run daily, the first one runs and the second is ignored without error.

If an error occurs when creating the snapshot, the entire checkpoint is invalid and an error message is logged. Further, if this happens during shutdown, the shutdown continues without creating a valid snapshot. If this happens during a scheduled checkpoint, the error is logged and no automatic checkpoint activity occurs until the next scheduled activity.

If the recovery log encounters any errors during startup, such as invalid permission on the drive, the startup fails and the system shuts down. Errors during runtime, such as running out of space during logging, stops all agents receiving events, a checkpoint occurs, and the servers shutdown.

E-mail alert notifications might be duplicated if the system shutdown abnormally while sending the notifications. The system writes a warning message to the system log under such conditions. Further, if the SMTP e-mail server also terminated abnormally, it is possible that the recipient will never receive the alert.

By default, the system begins checkpointing and recovery logging when the system starts. You can disable this feature for future restarts from the System Settings menu. However, to disable it when starting the servers, include the following option when starting the application server:

```
"-Dcom.celequest.property.Recover State on Restart"
```

The first time you start the servers, you can identify a recovery logging directory by including the following option on the command-line that starts the application server:

```
"-Dcom.celequest.property.Recovery Log Directory=/cqst_logs"
```

Viewing Checkpoint and Recovery Options

Thew following tables describes the checkpoint and recovery options.

Option	Description
Enable checkpoint scheduling	When on, causes checkpoints to occur on the defined schedules. Regardless of this setting, checkpoints always occur during normal server shutdown, and a checkpoint occurs whenever the system starts-up and no previous checkpoint have occurred.
Run Now	Executes an immediate checkpoint, regardless of schedule.
Checkpoint Schedule	List of checkpoint schedules. Add, edit, or remove schedules from this list. **Note:** Months that do not include the specified day, such as the 31st in June, are ignored.
Recovery Log Directory	Directory that contains the recovery and checkpoint log files. The filename is DEFAULTRECOVERYLOGGER_<n>. **Note:** If the source event is a flat-file agent, consider pointing to a different disc to improve input/output performance.

Working with Custom Branding

Custom branding settings are located in the Custom branding tab.

Custom branding allows you to change the images that appear on some screens throughout Business Activity Monitoring. The new images must be GIF, JPG, or PNG format, and they must size as specified in the labels on the Custom Branding tab. Further, you must change all of the images, or none; you cannot change less than all of the images.

These are the images you can change:

Component	**Image**	**Location**
BAM Dashboard	Login Screen	Login page.
	Header	Across the top of the application page.
	Information Pane	Information panel in the Dashboard Manager.
BAM Workbench	Login Screen	Login page.
	Header	Across the top of the application page.

After changing the images, the “Powered by Cognos” logo appears on the same pages.

Working with Encryption Configuration

Business Activity Monitoring encrypts passwords with public-key (asymmetric) encryption, which consists of a public key and a private key. The private key is used for encryption. The public key is used for decryption. Both keys are kept in a keystore, which is a Java KeyStore object. (For detailed information about KeyStore, refer to the javadoc.) To provide additional security, the private key is placed in a certificate, and the certificate is stored in the keystore. Business Activity Monitoring uses two keystores: one for the Administrator and one for users. Keystore entries (that is, the keys and certificates) are accessed through a unique alias, which is also used to created the public/private key pair. The aliases for Business Activity Monitoring are:

- CQ_USER_PASSWORD_ENC— This is the alias for encrypting the keys for customer created users.
- CQ_ADMIN_PASSWORD_ENC— This is the alias for encrypting the keys for the system user and system_monitor user.

Business Activity Monitoring bootstraps at startup with Administrator’s keystore. This keystore is provided on the installation CD. The file is admin.jks. You will need to specify its location with the “Admin Keystore Location” property in the System Settings Properties file. For more information about the System Setting Properties file, refer to the *Installation and Configuration Guide*.

After start-up you will need to provide the keys for the user keystore and specify its location. (A sample keystore user_sample.jks is provided on the installation CD.) To create your own keys, the Administrator will need to use the keytool utility. For instruction on how to use the keytool utility, go to http://java.sun.com/j2se/1.3/docs/tooldocs/win32/keytool.html.

The Encryption Configuration tab is where the Administrator specified the location of the user and Administrator keystores. In cases where the Administrator creates a new keystore, users are automatically migrated from the old keystore to the new one the next time they log in to the BAM Dashboard. New users are always added to the user keystore.

Caution: If your are changing your admin or user keystore, make sure that the old keystore remains in the same location with the same name until after all users have been migrated. If you are changing the location of your admin keystore, you must specify the same location in your system settings properties file.

Setting	Description
Admin Keystore Location	The location and name of the Administrator's keystore. You can set the Administrator's keystore location at start-up in the System Setting Properties file. For more information, refer to the *Installation and Configuration Guide*.
Admin Keystore Password	The Adminstrator's keystore password. This password is used when changing the name or location of the Admin keystore. The default Admin keystore password is 40fd2442fa.
Admin PrivateKey Password	The Administrator's password for accessing the BAM Workbench. The default admin private key password is 40fd2442fa.
Keystore Location	The location and name of the user keystore.
Keystore Password	The user keystore password. This password is used when changing the name or location of the user keystore. The default user keystore password is 8deb5102f8.
User PrivateKey Password	The user password that users will use to access the BAM Dashboard. The default user private key password is 8deb5102f8.

Working with Execution Throttling

The Execution Throttling tab monitors the rate that events are published and propagated into the view system and adjusts (*throttles*) the publication rate when levels exceed the propagation rate by a specified threshold. Use these settings to reduce the amount of memory that Business Activity Monitoring consumes storing unprocessed events.

Setting	Description
Throttling on Overproduction Enabled	Adjusts the publication rate for all event producers when they exceed the Event Threshold. This control does not affect the HTTP Post, Web Services, or system event logging event publishers. Use the Configuring Memory Monitoring to adjust the rate of those event publishers.
Throttling on Low Memory Enabled	Adjusts the publication rate for all event producers when the amount of available memory is low. See "Configuring Memory Monitoring" on page 105 for details about the memory thresholds.
Event Threshold	Throttle an event producer when it has published at least this many events that have not yet been processed. Throttling stops when the producer's backlog falls below this level.
Publisher Sample Interval	How frequently, in seconds, to examine the count of over-published events. Samples regardless of the Publisher Sample Event Count setting. Set to zero (0) to turn this off.

Setting	Description
Publisher Sample Event Count	How frequently, in event counts, to examine the count of over-published events. Samples regardless of the Publisher Sample Interval setting. Set to zero (0) to turn this off.
Average Smoothing Factory	Smooths fluctuations in the overflow sampling. High values are more sensitive to fluctuations than lower values. Do not change this setting without specific direction from Adobe Systems Incorporated.
Deviation Multiplier	Adjusts how rapidly an event producer is throttled relative to the event source overflow. High values are faster than lower values. Do not change this setting without specific direction from Adobe Systems Incorporated.
Backlog Weight	Influences how previous throttling actions affect current and future actions. Do not change this setting without specific direction.

Working with Geography Maps

The Geography Maps tab configures Geography Maps. The creation of Geography Maps is enabled when the “Use this for Geography Charts” checkbox is selected when configuring a dimension.

Setting	Description
Geo Cache Size	The number of geocodes (addresses) to cache. A geocode is added to the cache the first time it is used.
Geocoder Thread Pool Size	The number of threads available for simultaneous geocode requests.
Base URL for Geocoder	The URL to use for obtaining mapping information. For Google Maps, the URL is http://maps.google.com/maps/geo?
Geocode Key	Translates the address to longitude/latitude coordinates.For Google Maps, obtain the key from http://www.google.com/apis/maps/signup.html
Geocoder Classname	The Java class name that initializes geocoding and handles the retrieval of the latitude and longitude of the given address. For Google Maps, the class is com.celequest.maps.GoogleGeocoder.
Mapping URL	The mapping URL is always http://maps.google.com/maps/geo?
Mapping Key	A Google mapping key. This key is used for accessing map graphics. You can obtain a mapping key from http://www.google.com/apis/maps/signup.html

Working with the LDAP

In addition to allowing you to manually create users and user permissions, Business Activity Monitoring enables you to import user information from supported LDAP providers. Business Activity Monitoring can be set up for scheduled synchronizations with the LDAP server to update the user and roles; manual synchronization is also an option.

Importation takes place at the group level; therefore, users are imported only if they are members of a targeted group. When a user is imported into Business Activity Monitoring, a default e-mail UserProfile (called “PrimaryEmail”) is created and that cannot be modified. When Business Activity Monitoring synchronizes with the LDAP server, it uses this e-mail value to ensure that the UserProfile is consistent with the group memberships on the LDAP server.

When users are imported, they are assigned permissions based on the roles they are assigned in the source directory. If Business Activity Monitoring does not recognize a role, it creates the role and assigns a set of default permissions. You can later modify the permissions as necessary.

Setting Logging

The Logging tab configures how the system logs messages and identifies where the messages are logged. For detailed information about system logging, see "Understanding Logging" on page 112. To identify which system modules generate log messages, see the description in "Setting Logging Levels" on page 104.

Setting	Description
Logging Directory	Directory to receive log messages written to files. By default, no directory is defined and as such, no files are written.
Standard Out Log Level	Standard output (stdout) on the server hosting Business Activity Monitoring. May be redirected to a console. See "Understanding Logging Levels" on page 114 for details.
Summary File Log Level	General log file named bam.log in the log directory. Use this destination for normal logging. Typically this is set to receive ERROR level messages. See "Understanding Logging Levels" on page 114 for details.
Detailed File Log Level	Use this log when trying to track a problem where you do not want to add to the messages in the Summary file. This file is bam_detailed.log in the log directory. Typically this is set to OFF. See "Understanding Logging Levels" on page 114 for details.
System Appender Log Level	Publishes rows to the VC_SYSTEM_EVENTS event stream. This allows administrators to create views and rules that monitor that Business Activity Monitoring. Typically this is set to receive WARNING or ERROR level messages. Views and rules can then further filter messages as they arrive. See "Monitoring the Logs" on page 116 for more information.
Logging Synchroneity	Determines whether or not to buffer messages before writing them to the log. The default mode is asynchronous: messages are buffered for a short while before output to the log destination. Although asynchronous logging is more efficient, there can be delays between when a message is logged and when a message becomes visible in the destination. The delay should be no longer than about 10 seconds. If this delay is unacceptable, change the setting to synchronous, which outputs messages as they are published.
Log File Rollover Period	How often to close the log file and start a new one. Use this option to avoid creating overly large log files and to facilitate archiving the files. When a file is closed it is renamed with a date-time appended to the filename. The choices are Monthly, Weekly, Biweekly, Daily, Bidaily, Hourly, and Minutely. This field is only applicable when "DailyRollingFileAppender" is selected for Appender Class. **Note:** Bidaily logs are closed at noon and midnight.

Setting	Description
(FileAppender) Max Log File Size (MB)	Specifies the maximum size (in Megabytes) of a log file before roll over occurs. When the file reaches the maximum size a new log file is created. This field is only applicable when “RollingFileAppender” is selected for Appender Class.
(FileAppender) Max Number of Log Files	Specified the maximum number of log files. When the maximum number of files is reached the oldest log file is deleted. This field is only applicable when “RollingFileAppender” is selected for Appender Class.
Appender Class	Allows you to select one of two different policies for rolling over and archiving log files: rollover daily or rollover when the file reaches a size limit. To roll over daily, select “DailyRollingFileAppender.” For roll over based on file size, select “RollingFileAppender.” The "Log File Rollover Period" specified how often to roll over the log when the "DailyRollingFileAppender" is selected. The "Max Log File Size" setting specifies the maximum size of the log file before a new file is started. The "Max Number of Log Files" setting specifies the maximum number of log files to maintain.
Additional Log4j Properties	Points to a log4j.properties file that modifies or augments the logging infrastructure, such as setting default values, configuring appenders and modifying logger module levels. When this property is set, the file is loaded during startup and every minute thereafter. For details about this file, consult the “Class PropertyConfigurator” documentation in the Log4j product from the Apache Jakarta Project (http://jakarta.apache.org).
Log File Prefix	Adds the specified prefix to the log files.

Setting Logging Levels

The Logging Levels tab configures identifies which system modules generate log messages, and what level of messages they publish. For details about using this tab, see the description in “Understanding Logging Modules” on page 113. For detailed information about system logging, see “Understanding Logging” on page 112.

Configuring Memory Monitoring

The Memory Monitor tab configures how the system behaves when the amount of Java application virtual memory heap on the server becomes low. When low memory conditions occur, these settings tell the event producing agents to slow or stop publishing new events to the event streams.

The threshold settings identify percentage of memory consumed before the action occurs. For example, a setting of 85 means that there is 15% free memory available. Once consumed memory crosses a threshold, the threshold action remains in effect until the memory level falls below the triggering threshold.

Adjust the thresholds smaller only if your system frequently encounters out of memory conditions not trapped by the Working with Execution Throttling. Do not raise these levels without specific direction from Adobe Systems Incorporated (see note below); rather, you should increase the memory on the server or off-load non-Business Activity Monitoring processes to other servers.

Setting	**Description**
Memory Monitor Interval	Frequency to look at memory levels, in seconds. Disable sampling by setting the interval to zero (0).
Low Memory Threshold	Slows event production and instructs the system logger to discard debug- and information-level messages when this percentage of memory is consumed.
Dangerous Memory Threshold	Impedes event production and restricts the system logger to record fatal errors only when this percentage of memory is consumed.
Critical Memory Threshold	Blocks event production and randomly sheds system logging events when this percentage of memory is consumed.

The thresholds apply to the amount of memory currently allocated by the application server. It is possible to cross a threshold for the current memory allocation, but not be near the same threshold for maximum memory available to the server. Normally the application server will allocate more memory before a threshold is crossed. However, for optimum performance of this feature, you should set the application server’s minimum virtual memory setting (*ms*) to be the same as its maximum (*max*) setting.

Working with Single Sign-on Settings

Single sign-on settings are located in the Single sign-on tab.

Single sign-on allows users on networks that are running an identity and access management software to bypass the login pages on Business Activity Monitoring applications. When enabled, users login only to their network. When they later connect to a Business Activity Monitoring application, the access management software takes care of user authentication and opens the application in the user's browser.

All users must have an account in Business Activity Monitoring, in addition to having a network login account.

Setting	Description
Enable single sign-on	Allows users to by-pass the application login. You must restart Business Activity Monitoring servers to affect and changes in this status.
Sign-off URL	Where to send the user's browser when they sign-off from Business Activity Monitoring application. Normally the application sends the user to the application's login page.
Request Header	HTTP request header that contains the user's Business Activity Monitoring user name. This header is provided in the request from the access management software when the user navigates to a Business Activity Monitoring application. See the access management software documentation to determine the header name. For example, Netegrity 3.0 uses SMUSER. If the header is missing or invalid in the HTTP request, the user is directed to the application login page.
Cognos 8 SSO SOAP Endpoint URL	The location of the web service for Cognos 8 single sign-on. This is a C8 URL that will return the username for a particular passport ID.
Cognos 8 SSO CAMNamespace	The namespace ID used to identify the TSO provider on the Cognos 8 platform.

Viewing SMTP Configuration

The SMTP (simple mail transfer protocol) configuration specifies how Business Activity Monitoring connects to the mail server that delivers e-mail notifications. The server is external to Business Activity Monitoring and is managed by your e-mail system administrator. Contact that administer to setup an account for Business Activity Monitoring, and for details about these configuration settings.

Setting	Description
SMTP Host	Name of the e-mail host that provides the transport. A typical name might look like this: mail.mydomain.com
SMTP From address	Address that appears in the From field for all e-mail messages sent by the system.
SMTP User	User name that the system uses to access the transport.
SMTP Password	Password that validates the user to the transport.

Setting Alert/Notifications Limitations

The Notifications tab in the System Settings dialog box enables you to set limitations on the number of alerts that can be set during a specified time interval. For example, if you set the maximum number of alerts to 100 and the interval to 30 minutes, the 101st (and every subsequent) alert will not be sent.

Note: These settings are *per defined alert*.

The Notifications tab enables you to set the follow properties:

Setting	Description
Alert Acknowledgment URL	A URL to include in the alert.
Maximum Alerts To Be Fired Per Interval	Specifies the maximum number of alerts that can be set during the defined interval period.
Interval Size	Duration of the interval in number of time units (seconds or minutes, depending on the below setting).
Interval Units	Set to seconds or minutes as desired.

Viewing System Controls

System controls, located in the System Control tab, administer the operation of Business Activity Monitoring installation.

Setting	Description
Recover State on Restart	Loads the last checkpoint data when the server restarts. Turn this off to start with no checkpoint data. See "Working with Checkpoint and Recovery" on page 97 for details about checkpoints.
Seconds Before Starting Pollers	How long to wait before the event pollers start looking for new data. This delay gives developers time to disable events on restart before they begin feeding data into the event stream.

Viewing Thread Configurations

Thread configurations can be viewed in the Thread Configuration tab.

Business Activity Monitoring uses execution threads that allow the system to processes views, alerts, rules, context queries, on so on in parallel — multiple at the same time — when running on multiple CPU hosts. These settings tune the thread processing.

Note: Do not change these settings without direction from Adobe Systems Incorporated.

Setting	Description
Number of Runtime Execution Threads	Count of processing threads to use. Increase this value only when using four or more CPUs. Default is 8.
Maximum Dequeue Batch Size	Count of processes to extract from the waiting pool. Default is 5.
Maximum Dequeue Block Wait in Milliseconds	When to check for missed extractions. Default is 50.

Viewing User Interface

These settings, located in the User interface tab, apply to the user interface seen by all users.

Setting	Description
Maximum View Snapshot Rows Returned	Greatest count of rows to display when showing the contents of a view. Zero (0), the default, is unlimited.
Alert Expiration Interval	Maximum days to retain alert notifications. This is the greatest count of 24-hour days to retain alerts in the BAM Dashboard. Timing begins when the alert is generated. For example, if the expiration period is 1 day, the alert is removed from the BAM Dashboard at least 24 hours after it was generated. **Note:** Expired alerts are removed regardless of their state. Alerts do, however, retain their state; they just no longer appear in the BAM Dashboard
Maximum Alerts Instances to Display	Maximum number of alert notifications displayed in the BAM Dashboard.
Webservice Event Host	The name of the host to use for Web Service events. For example, localhost.
Webservice Event Port	The port to use for Web Service events. For example, 8080.

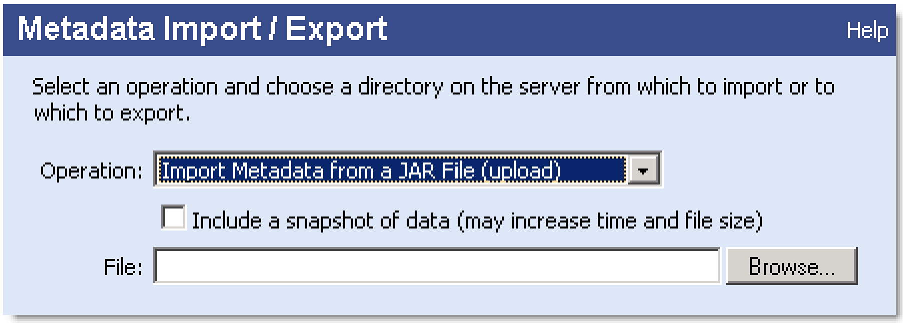
Importing/Exporting Metadata

Metadata are descriptions of the object and user definitions, and system settings. This information is stored in an RDBMS database. The Export action writes this information to one of the following:

- XML files in a directory on the server
- A single compressed JAR file on the server or local client.

Later you can use the Import action to load those objects back into the installation.

By default, the operation includes the definition of the metadata objects only. Optionally, you can include a snapshot of the state of the data in the objects by turning on "Include a snapshot of data." This option is available for both import and export, but for import it only has effect if the exported data included a snapshot. Note also that using this option can greatly increase the time to perform the operation. See "Working with Checkpoint and Recovery" on page 97 for details about snapshots.


Metadata Import / Export

Select an operation and choose a directory on the server from which to import or to which to export.

Operation: Import Metadata from a JAR File (upload)

□ Include a snapshot of data (may increase time and file size)

File:

Exporting Metadata

The exported contain the descriptions of all objects in the system, and all system settings, similar to a backup of the installation. (See "Working with Checkpoint and Recovery" on page 97 for another form of system backup.)

Alert subscribers are not exported. Checkpoint history is included only when "Include a snapshot of data" is checked.

Where is it Written

When exporting XML files to a directory on the server, the files are duplicated into two sibling subdirectories. One subdirectory is always named *latest*. The other is a new folder whose name matches the timestamp of the export. In this way, you can always quickly identify the last export by looking in *latest*, while the timestamp-named directories act as specific, past export instances.

> **Note:** Files written to the server have the access permission of the account used to start the Business Activity Monitoring server. Further, the files are not encrypted. As such, all user account information is available to anyone with access to the files.

When exporting to a JAR file, the file contains compressed definitions of all of the objects.

➤ **To export metadata:**

1. Open the Administration tab.
2. Click **Import/Export**.
3. Select one of the Export options.
4. Locate the file or directory of files to export.

   For server options:

   - Identify the full path to the directory to contain the exported metadata.
   - Write the data to storage by clicking OK. The filename on the server is _export.jar.

   For client (download):

   - Click **OK** to begin the download.

   Note: Depending on how your browser is configured, you might receive a prompt asking you to confirm that it is OK to download a .JAR file. Select **Yes**.

   - Use the browser download dialog to identify the location — and optionally change the filename — on your local machine.
   - When finished, click **Cancel** to close the Import/Export dialog.

This completes the steps for exporting metadata.

Importing Metadata

The import action reads the previously exported metadata. When importing from the server, you must know the full path to the location for the .JAR file or directory of files. To import from the local machine (upload), you may browse to identify the location.

Note: Importing a large set of metadata from a busy DBMS can take several minutes to complete. In some cases, the application server might time-out waiting for the entire operation to complete. If you get a log error message about transaction incomplete, increase the time that the server allocates to transactions. For example, by default BEA WebLogic times out after 5 minutes (300 seconds).

What is Read

All definitions not already defined in the installation are imported. If an object is already defined in memory, it is ignored and left as-is in memory. No error occurs if an object is already defined in memory.

If an object would be invalid, it is not imported. For example, if a scenario definition is missing from the exported data, importing that data set will not import any of the objects contained by the missing scenario.

If the exported data contains snapshot data, you may include that snapshot with the import by turning on “Include a snapshot of data.” If the exported metadata does not contain a snapshot and this option is on, no error occurs (though a warning does get logged).

➤ **To import metadata:**

1. Open the Administration tab.

2. Click **Import/Export**.

3. Chose one of the Import options.

4. Locate the file or directory of files to import.

   For server options, identify the full path to the directory containing previously exported metadata.

   For client (upload), use the **Browse** button to identify the file on your local machine.

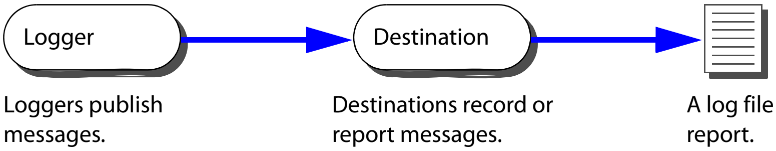
5. Read the data from storage by clicking **OK**.

6. If the dialog remains open after the import completes, choose **Cancel** to close it.

This completes the steps for importing metadata.

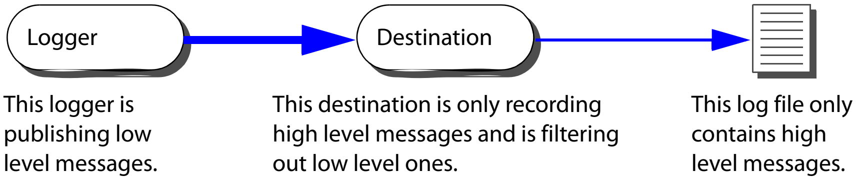
Understanding Logging

Logging facilitates software service and maintenance by producing reports suitable for analysis by end users, system administrators, support engineers, and software development teams. The various Business Activity Monitoring subsystems (see "Understanding Logging Modules" on page 113) generate messages that provide information about the application state or about internal system events. These messages can be logged to Viewing Logging Destinations that include files, console, or an event stream suitable for monitoring by a rule in a scenario.


Logger
Destination
Loggers publish messages.
Destinations record or report messages.
A log file report.

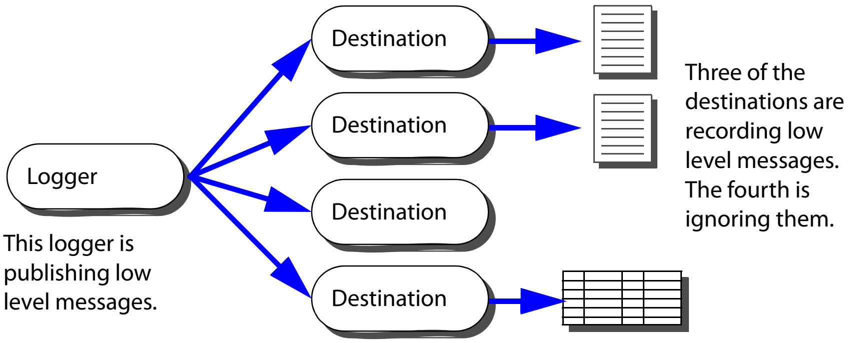
Most of the messages are not useful for general consideration: message that application developers use to trace behavior of the program. However, some message are important and can indicate problems with the system. These messages can help system administrators identify problems and to possibly see the events that led up to the problem.

Messages have associated *logging levels* that indicates the importance and urgency of the message. Messages with higher levels indicate higher levels of importance. Using the levels you can tell the loggers which messages to publish, and you can tell the destinations which messages to include. Destinations ignore messages of lower priority than they are set to record.


Logger
Destination
This logger is publishing low level messages.
This destination is only recording high level messages and is filtering out low level ones.
This log file only contains high level messages.

Note: When configuring logging, the destination must be set to match the level of the logger if you want to see all of the messages the logger publishes. Otherwise, the destination will filter and omit the lower level messages.

Business Activity Monitoring has dozens of loggers, most of which are for assisting application development are not interesting to installations. When a logger publishes a message, each of the destinations checks the message level and publishes it if it matches the level that the destination is configured to accept. As such, it is possible for all of the destinations to record a message published by one logger.



This logger is publishing low level messages.

Three of the destinations are recording low level messages. The fourth is ignoring them.

Viewing Logging Destinations

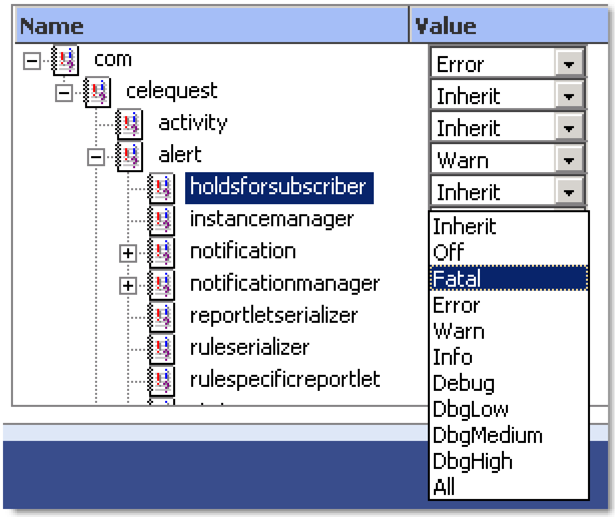
Logging destinations identify where log messages are recorded when the destination receives messages of the level it is configured to receive. The logging destinations are:

Setting	Description
Logging directory	Directory where log files are written. By default, no directory is defined and as such, no files are written.
Standard Out	Standard output (stdout) on the server hosting Business Activity Monitoring. May be redirected to a console.
Summary File	General log file named bam.log in the log directory. Use this destination for normal logging. Typically this is set to receive ERROR level messages.
Detailed File	Use this log to track problems when you do not want messages to go to the summary file. The file bam_detailed.log is in the log directory. This is OFF by default.
System Appender	Publishes rows to the VC_SYSTEM_EVENTS event stream. This allows administrators to create views and rules that monitor that Business Activity Monitoring. Typically this is set to receive WARNING or ERROR level messages. Views and rules can then further filter messages as they arrive. See “Monitoring the Logs” on page 116 for more information.

Understanding Logging Modules

Loggers publish messages that provide information about the application state or about internal system events. Each logger publishes messages of at least the log level that you request, and discards those of lower priority.

Each logger has a name that determines where that logger fits into the logger hierarchy tree. The logger hierarchy tree allows you to configure logging details at various levels of granularity. Because each logger inherits its logging level from its parent in the logger hierarchy tree, you can assign a logging level to the top-most logger (com) and all Business Activity Monitoring loggers inherit that setting. By assigning a level to a specific logger, you can override the value inherited from the parent. For example, in this illustration, all loggers report Error level messages only, while all “alert” loggers report Warning level and above, except for “holdsforsubscriber” which is only reporting Fatal errors.



Name	Value
com	Error
celequest	Inherit
activity	Inherit
alert	Warn
holdsforsubscriber	Inherit
instancemanager	
notification	
notificationmanager	
reportletserializer	
ruleserializer	
rulespecificreportlet	

Note: Do not turn on Debug logging for modules without direction from Adobe Systems Incorporated. Turning on this excessive logging for high volume modules like system, query, or exec can rapidly flood your system and kill the servers.

Understanding Logging Levels

Each log message has an associated log level that gives a rough guide to the importance and urgency of the message. Each level has an associated integer value usable by rules that monitor system messages. Higher values indicate higher priorities. As such, a rule might look for Error and Fatal messages by looking for values greater than or equal to 40,000 (Level>=40000).

Level	Value	Description
Inherit	—	Inherit the level from the parent logger.
Off	—	No logging.
Fatal	50,000	Very severe error events that will presumably cause the application to terminate.
Error	40,000	Error events of considerable importance and which will prevent normal program execution, but which might still allow the application to continue running.
Warning	30,000	Potentially harmful situations of interest to end users or system managers, or which indicate potential problems.
Info	20,000	Informational messages that might make sense to end users and system administrators, and which highlight the progress of the application.
Debug	10,000	Relatively detailed tracing used by application developers. The exact meaning of the three debug levels varies between subsystems.

Level	Value	Description
DbgLow	9,000	Information broadly interesting to developers who do not have a specialized interest in the specific subsystem. Might include things like minor (recoverable) failures and issues indicating potential performance problems.
DbgMedium	8,000	Fairly detailed tracing messages. Calls for entering, returning, or throwing an exception are traced at this level.
DbgHigh	7,000	Highly detailed tracing messages. Produces the most voluminous output.
All	—	All messages.

From time-to-time, it may be necessary to increase the logging level of a logger in order to diagnose or debug a problem. The default level for all loggers is Inherit, and the default level for the root logger is Info.

Do not turn on Debug or higher logging without direction from Adobe Systems Incorporated. Turning on this excessive logging for high volume module like system, query, or exec can rapidly flood your system and kill the servers.

Assigning Log Levels to Loggers and Destinations

This section shows you how to assign a log level.

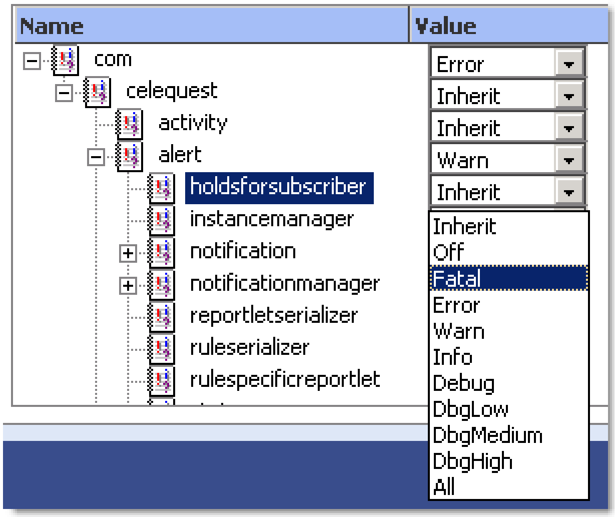
➤ To assign a log level to a logger or destinations:

1. Open the **Administration** tab.
2. Click **Edit System Settings**…
3. Choose the **Logging** tab.

   Choose the value for the specific destination or destinations.

4. Choose the **Logging Levels** tab.

   Choose the value for the specific logger. Note that a sub component can have a higher or lower priority level than its parent.



Name	Value
com	Error
celequest	Inherit
activity	Inherit
alert	Warn
holdsforsubscriber	Inherit
instancemanager	
notification	
notificationmanager	
reportletserializer	
ruleserializer	
rulespecificreportlet	

5. Click **OK** to save the settings.

The changes take effect immediately.

Monitoring the Logs

You can build a rule that monitors the messages, looking for high priority error conditions, and that reports them in an alert to key administrators. When the destination of System Appender Logging is capturing log messages, it writes them to the VC_SYSTEM_EVENTS event, which has the following fields:

Field	Description
ERROR_LEVEL_INT	Message level. Understanding Logging Levels" lists the values and what they mean.
ORIGINAL_VC_EVENT_ID	ID of the event that generated the message.
USER_NAME	User that was using the application that generated the message.
MODULE_NAME	Subsystem that generated the message. See "Understanding Logging Modules" on page 113 for information.
MESSAGE_TEXT	Text of the message.
OBJECT_NAME	Name of the object that generated the message.

Note: You cannot edit or delete this table.

You can build views on top of the VC_SYSTEM_EVENTS event, and then build rules that monitor the views that are looking for specific messages, usually those that have a high priority.

➤ **To monitor the logs:**

1. Turn on logging for the System Appender and set the logging levels to the priority you want to record.
   - Open the Administration Console and click **System Settings**.
   - Set the Understanding Logging Levels to publish the messages you want. For example, set com.celequest to Warning.
   - On the Logging settings, set System Appender Log Level to the same level, such as Warning.
   - Click **OK** to save the setting and close the dialog.
2. Create a new view based on all fields in the VC_SYSTEM_EVENTS event and with no context.
   - Open the Application Workbench and click **New View** and choose the VC_SYSTEM_EVENTS event and no context.
   - Assign a view name, such as *SysEventView*.
   - Drag all of the event fields into the view.
   - Click **Save View** to save the view and close the dialog.
3. Create a Business Activity that contains a Scenario for the system events.
   - Open the Scenario Modeler.
   - Create a Business Activity name *SysEventsActivity*.
   - Create a Scenario within that activity named *SysEventsScenario* based on the *SysEventView* view.
4. Create a rule to monitor the view, and an alert and reportlet to report on the messages.
   - Create a Rule named *SysEventsRule*.
   - Define the Rule Condition as TRUE to report all messages that enter the view.

     Optionally, if you want to further filter messages, define a more specific rule. For example, to alert only Error messages and higher, create a rule that filters for ERROR_LEVEL_INT>=40000 (the Error level). For a list of log level values, see “Understanding Logging Levels” on page 114.
   - Create an Alert name SysEventsAlert, and assign the SYSTEM as a mandatory subscriber. Optionally assign other subscribers.
   - Optionally, create a Reportlet to contain the message information.
   - Finish defining the alert.
   - Click **Finish Rule** to save the changes.

The system immediately begins monitoring the message log system.

Index