



FileNet Business Activity Monitor

Using FileNet BAM Workbench

Release 3.5.1

November 2005

FileNet is a registered trademark of FileNet Corporation.
All other product and brand names are trademarks or registered trademarks of their respective companies.
Due to continuing product development, product specifications and capabilities are subject to change without notice.

Copyright © 2001, 2005 FileNet Corporation. All rights reserved.

FileNet Corporation
3565 Harbor Boulevard
Costa Mesa, California 92626
800.FILENET (345.3638)
Outside the U.S., call:
1.714.327.3400
www.filenet.com

Notices

For notices regarding this documentation, refer to [Help Directory > Notices](#) in the FileNet P8 online documentation.

Typographical Conventions

This document uses the conventions in the following table to distinguish elements of text.

Convention	Usage
UPPERCASE	Environment variables, status codes, utility names.
Bold	Paths and file names, program names, clickable user-interface elements (such as buttons), and selected terms such as command parameters or environment variables that require emphasis.
<i>Italic</i>	User-supplied variables and new terms introduced in text.
<i><italic></i>	User-supplied variables that replace everything between and including the angle bracket delimiters (< and >).
Monospace	Code samples, examples, display text, and error messages.

Contents

Getting Started	6
Modeling Business Data	8
Working with the scenario modeler	9
Viewing object status and validity	10
Working with business views	11
Views in the Scenario Modeler	12
Creating views	13
Understanding relationships	14
Working with view fields	15
Affecting the results	16
Filtering results	16
Ordering results	17
Maintaining results	17
Working with business activities and scenarios	18
Working with rules and alerts	19
Working with rule details	21
Working with alert details	22
Working with rule templates	25
Understanding rule templates	25
Creating a rule template	26
Setting parameters	28
Working with reportlets	30
Receiving Metrics	31
Understanding active reports	32
Managing active reports	33
Managing ad-hoc queries	34
Managing real-time charting	35
Troubleshooting active reports charts	37

Enabling Support for Business Views	38
Working with the FileNet BAM Workbench	39
Viewing dependencies	40
Viewing join relationship definitions	41
Working with events and context tables	42
Working with events	43
Working with context	44
Upload Event File option	44
Working with consolidated events	45
Working with views	46
Creating and editing views in the FileNet BAM Workbench	47
Creating view fields	49
Working with clauses	53
Maintaining events in a stateless view	55
Persisting view data	55
Displaying SQL expressions	56
Viewing plan relationships	57
Working with cubes and dimensions	59
Creating dimensions	61
Creating cubes	61
Cube plan relationships	63
Checking cube results	65
Working with UDFs and JAR files	65
JAR files folder	66
User defined functions folder	66
Working with external links	67
Working with process definitions	68
Creating process definitions	69
Managing System Administration	71
Managing users	72
Managing roles	73
Viewing the agents list	74
Viewing the external processes list	74
Working with system settings	75
Working with checkpoint and recovery	76

Working with execution throttling	78
Working with the LDAP	79
Setting logging	79
Setting logging levels	81
Configuring memory monitoring	81
Working with single sign-on settings	82
Viewing SMTP configuration	83
Viewing system controls	83
Viewing thread configurations	84
Viewing user interface	84
Importing/exporting metadata	85
Exporting metadata	85
Importing metadata	86
Understanding logging	87
Viewing logging destinations	88
Understanding logging modules	89
Understanding logging levels	89
Assigning log levels to loggers and destinations	91
Monitoring the logs	92
Index	94

Getting Started

FileNet Business Activity Monitor 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.

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

- [“Receiving Metrics” on page 31](#)
- [“Modeling Business Data” on page 8](#)
- [“Enabling Support for Business Views” on page 38](#)
- [“Managing System Administration” on page 71](#)

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

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 FileNet Business Activity Monitor is through a Microsoft Internet Explorer browser using one of FileNet Business Activity Monitor applications, at an address provided by your site administrator:

FileNet 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://filenetHost:port/filenet
```

FileNet 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://filenetHost:port/filenet/dashboard/default.htm
```

3. Everyone who uses FileNet BAM Workbench or FileNet 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 72](#)].

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

System administrators create and modify agents with the Administration Console [see [“Viewing the agents list” on page 74](#) 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 FileNet BAM Workbench [see [“Enabling Support for Business Views” on page 38](#)].

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

You can create complex business views with the FileNet BAM Workbench. You can also create business views with the Scenario Modeler [see [“Working with views” on page 46](#)].

7. At the heart of FileNet Business Activity Monitor 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 8](#)].

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

This application presents the details about the alerts and metrics, and allows users to subscribe to new alerts [see [Alert Messages](#)].

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

Modeling Business Data

Business data modeling is a technique for describing the events, context, and rules that depict how your business functions. Modeling in FileNet Business Activity Monitor 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 FileNet BAM Workbench, all data modeling is performed with the Scenario Modeler. This chapter describes how to use:

In this Chapter:

[“Working with the scenario modeler” on page 9](#)

[“Working with business views” on page 11](#)

[“Working with business activities and scenarios” on page 18](#)

[“Working with rules and alerts” on page 19](#)

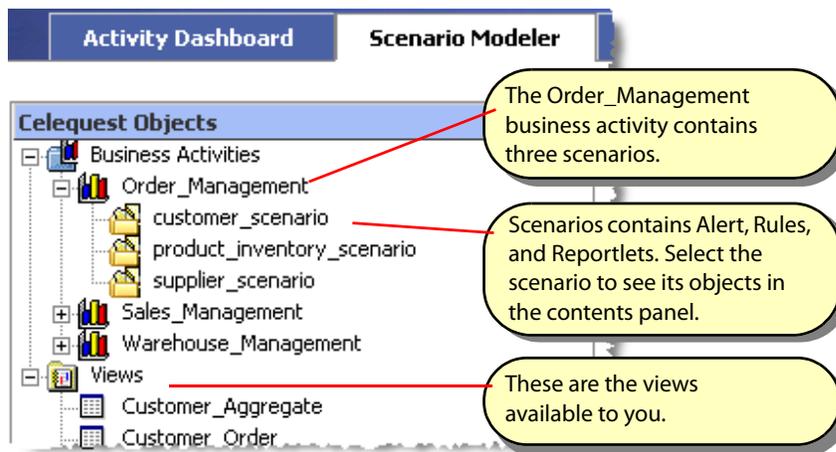
[“Working with rule templates” on page 25](#)

[“Working with reportlets” on page 30](#)

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.



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 great detail.

- [“Working with business views” on page 11](#)
- [“Working with business activities and scenarios” on page 18](#)
- [“Working with rules and alerts” on page 19](#)
- [“Working with reportlets” on page 30](#)

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 40](#) 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	Stand-by 
OrdersPerCustomerCountView	This is the orderspercustomer vie	Running 
 Invalid ProductCountView	This is the ordersperproduct vie	Stopped 
OrdersPerProductView	View showing incoming orders alo...	 

NOTE: See *Object status* in the *Server Reference* documentation for more detailed information about object states, and the effects of changing a state.

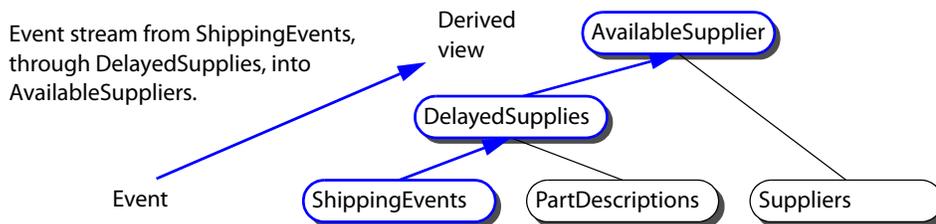
Further, if an object has a reference to another object which cannot be satisfied, that object is *invalid*. A reference can be invalid because an object does not exist or because some attribute of an object does not match the requirements of the dependent (such as a data type mismatch), not because 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 FileNet Business Activity Monitor. 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 FileNet BAM Workbench to create join relationships (see [“Viewing join relationship definitions” on page 41](#) for details).

A key power of business views is the ability to aggregate event and context information; 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.

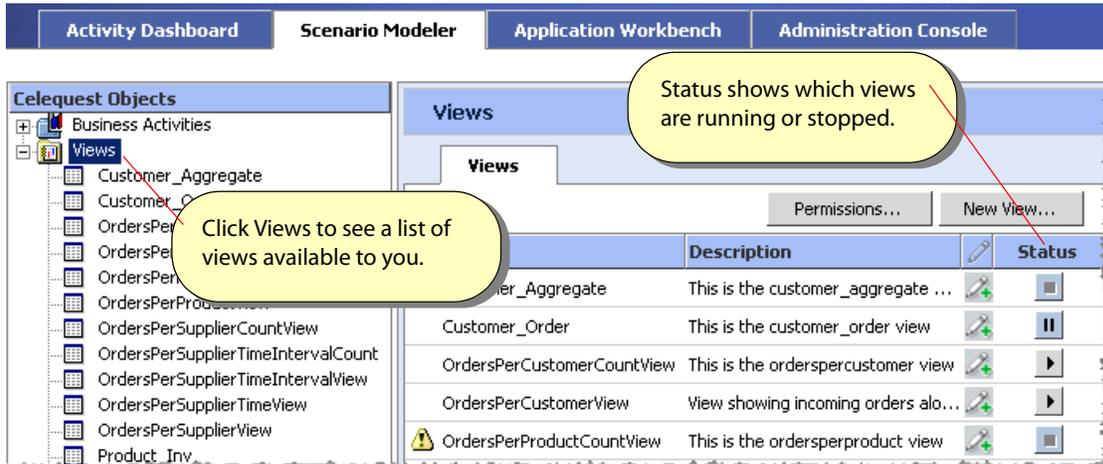
By combining aggregate data with other context, you can create more meaningful metrics for various scenarios. For example, a view might track the performance of sales representatives to identify which are on target to meet their individual quotas for the fiscal quarter. Each sale is recorded as an event, 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 15](#) for information about creating formulas in fields.

The rest of this topic 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 46](#).

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.

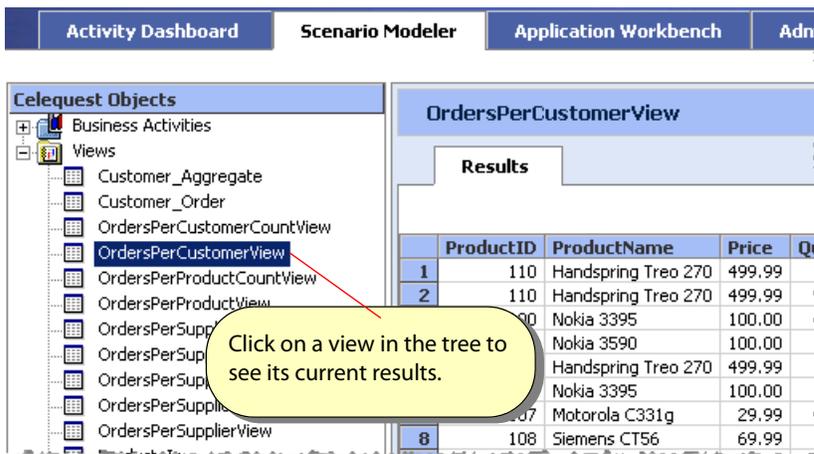


The Views list identifies each view available to you, shows the views valid/invalid state, and shows its running or stopped status. When a view is stopped, it is not receiving events, nor is it passing them to any views, rules, alerts, or reportlets that depend on that view. See [“Viewing object status and validity” on page 10](#) 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 13](#).

Viewing results

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



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 FileNet BAM Workbench. See [“Enabling Support for Business Views” on page 38](#) 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,”](#) below, for details. If the event or base view doesn’t 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 FileNet BAM Workbench. See [“Understanding relationships,”](#) below, 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 these following topics:

- [“Understanding relationships” on page 14](#)
- [“Creating views” on page 13](#)
- [“Affecting the results” on page 16](#)

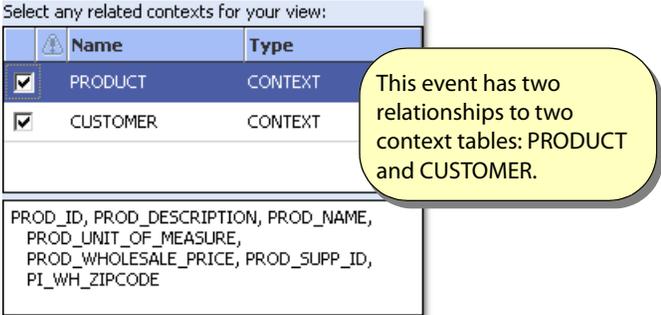
Save the view in a running state (*enabled*) and it will immediately be ready to receive events. For a discussion about the view states, see [“Viewing object status and validity” on page 10](#).

Understanding relationships

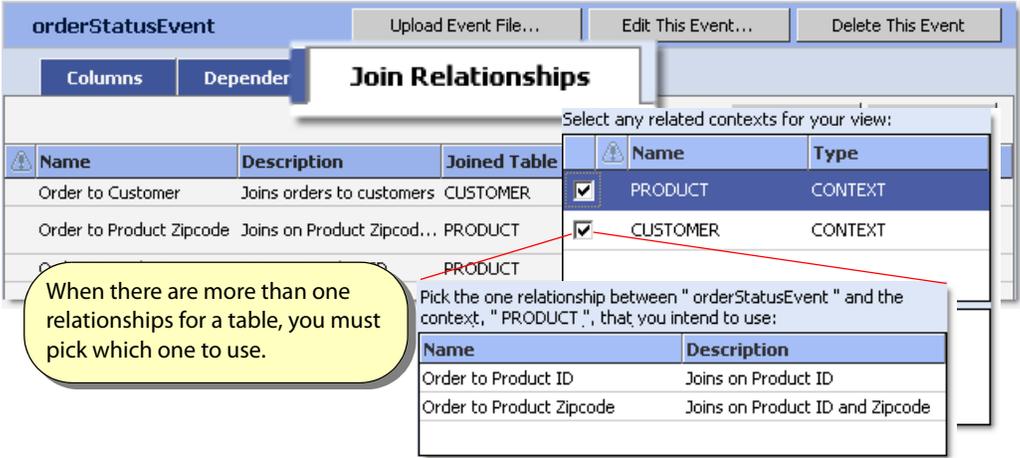
Join relationships define the connection between tables or views and context tables. In the Scenario Modeler you use join relationships to create views that join events and context.

NOTE: Define join relationships in the FileNet BAM Workbench. See “Viewing join relationship definitions” on page 41 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.

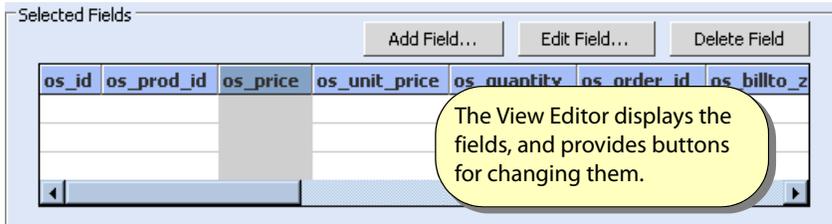


When there are multiple relationships to the selected table, a subsequent dialog prompts you to choose the one you want to use. For example, this illustration shows the relationships defined in the FileNet BAM Workbench, and then the two dialogs in the Scenario Modeler that prompt for the relationships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

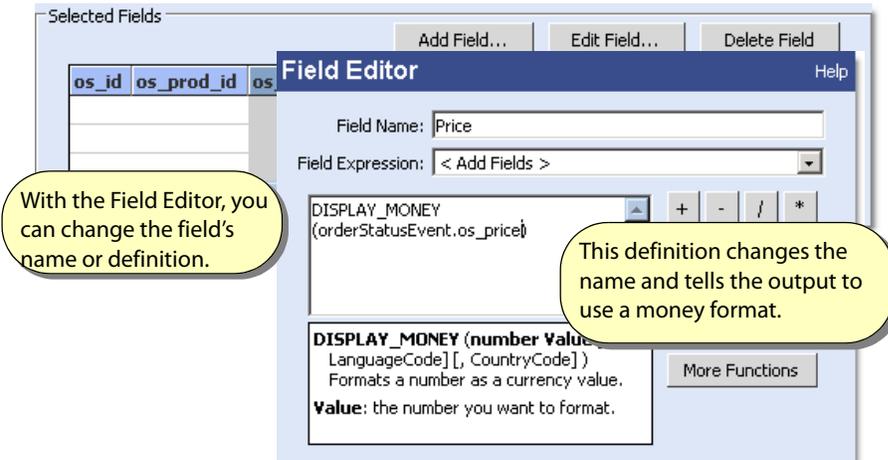


Working with view fields

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.

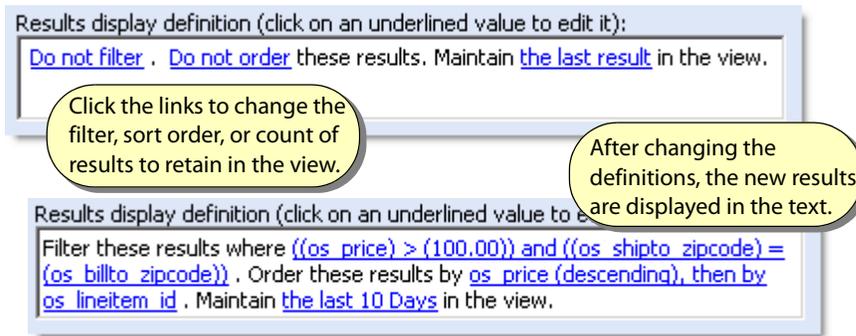


To change a field's name, or to change the definition of the field, click **Edit Field** and use the Field Editor.



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.



The following topics describe these effects in detail:

- Filtering results
- “Ordering results” on page 17
- “Maintaining results” on page 17

Filtering results

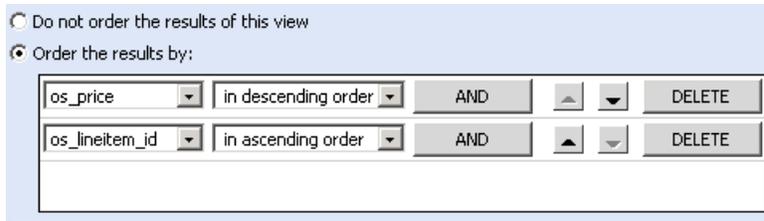
Filters allow you to restrict the view results to include only those rows that meet the specified condition. The condition can be simple, like *only those products whose price is greater than 100.00*. Filters can also be complex and with multiple conditions, all of which must be met before the result is allowed into the view. The example in this illustration 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.



To create more complex filters, use the FileNet BAM Workbench and define the filter in the view’s Where clause. See “Working with clauses” on page 53 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, and then in order from there. For example, in this illustration, the results are sorted by price, and items with the same price are sorted by ID:

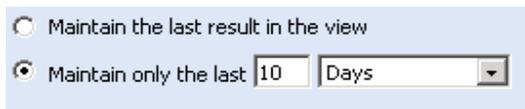


NOTE: The Order Results option is only available for stateful views; stateless views cannot be ordered. For a detailed explanation, see Stateless and stateful views.

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. This illustration keeps 10 days worth of events:

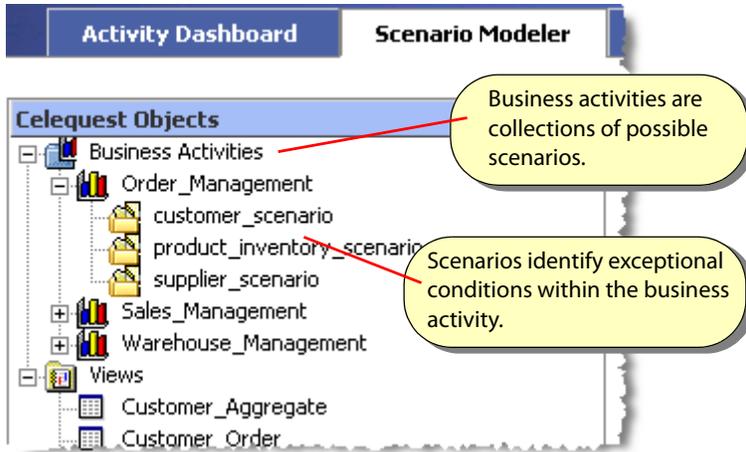


The number specifies the maximum number of events to maintain. For example, when maintaining 10 events, when the eleventh event arrives it is kept, but the oldest is discarded.

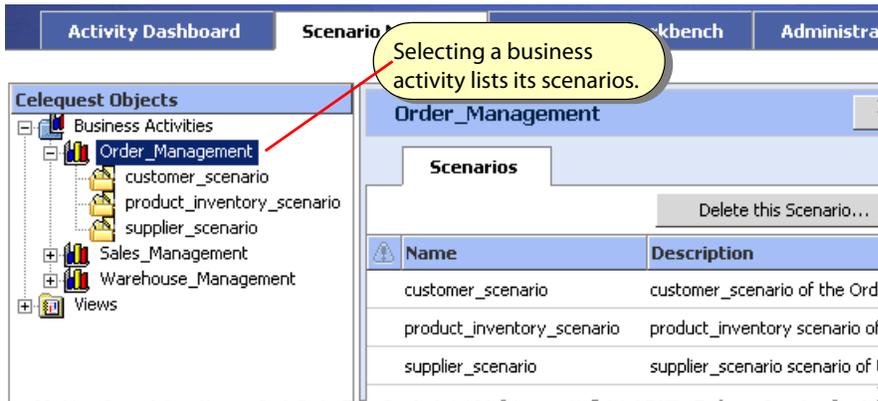
When using a time interval, the actual count of events in the view varies depending on the number of events in the interval when the view updates. The set of events is determined *when the last event was inserted into the view*, not at the current time. For example, an interval of one hour shows all 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 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.



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.



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 10](#) for details.

NOTE: For detailed information about each of these objects, see Business Activities and Scenarios

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.



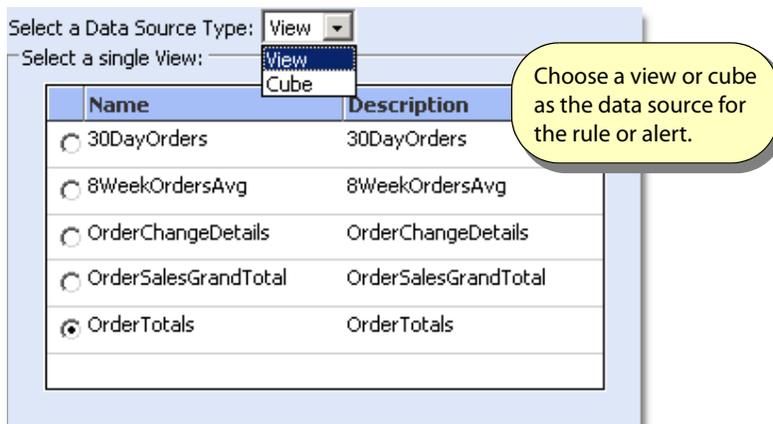
To access a rule, alert or reportlet:

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**.



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 Select Data Source

	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 21](#). 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 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
<input checked="" type="radio"/> OrderCube	Aggregate sales orders over product,...

Select a level for each Dimension:

Name	Level
Date/Time Dimensions	All
Location Dimensions	Region
Product Dimensions	Family

Select Dimension Filters: (Optional)

Select a Filter Dimension: Family Family = 'Hardware'

Select a Filter Level: Region

Select a Filter:

Add Filter Delete Filter

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

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: Status:

Description: (Optional)

Define Rule Condition:

Condition:

Holds For:

Basic **Advanced**

Advanced allows you to select multiple alert options

Choose an Alert to use:

Select a type of action:

Maintain alert states for:

Consolidate multiple messages from same event

Callouts:

- Conditions look at fields in the view and detect exceptional
- Holding performs the action only when the condition exists for a period of time
- Actions are what happens when the
- 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, like this $(InvLevel < MinThreshold \text{ OR } InvLevel < (AverageInvLevel * .90))$. For a detailed discussion about conditions, see Rule condition.

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, reset 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

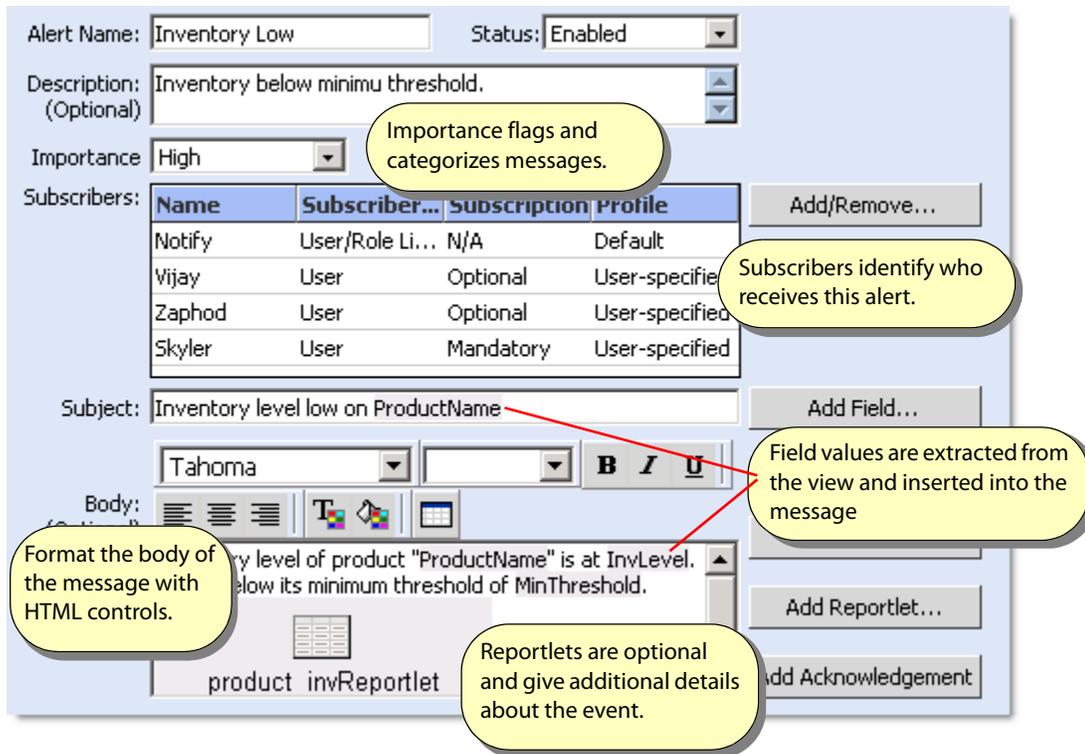
Rules usually perform their action as soon as the condition exists. However, in some cases you would rather wait to see if the condition is corrected before alerting key personnel. In that case, you can specify a duration of time to wait. If the condition exists the entire time while the rule waits, 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 19 for more information.



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 FileNet BAM Dashboard. For details about subscribing to alerts, see Setting notification properties.

NOTE: For a detailed discussion about subscribers, see Alert subscribers.

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 picture above, the value of ProductName from the view or cube is inserted into the Subject line.

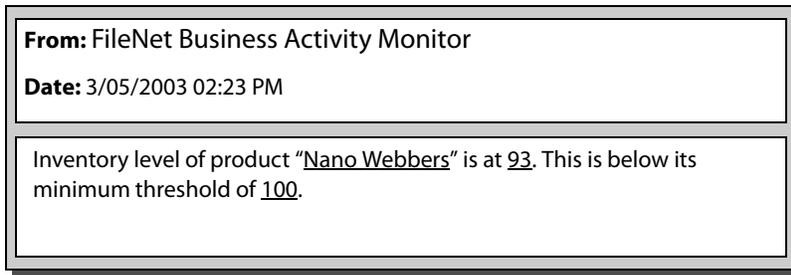
When the associated view or cube has multiple rows for the event, by default, the value from each row is inserted into the field place holder separated by commas. For example, the Subject might look like this:

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 this:

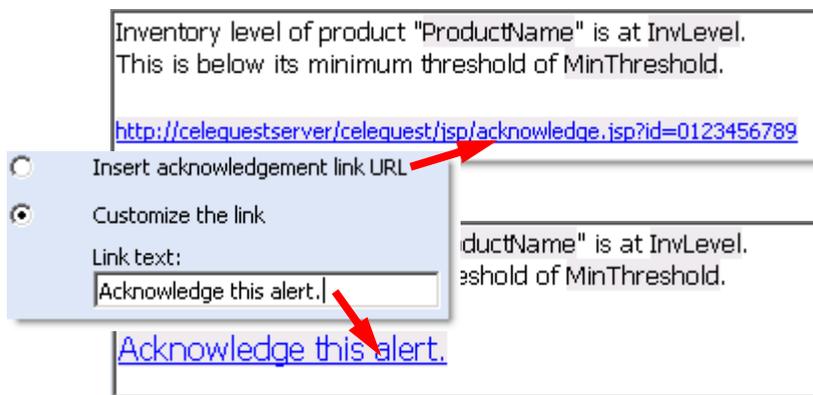


For a complete discussion about alerts, see Alerts.

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.



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 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 30.

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.

Working with rule templates

Rule templates allow FileNet 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 FileNet BAM Dashboard to create multiple, unique business rules with on 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:

- 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, and 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 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 29](#) for information.
- Optionally define a Holds for duration. See [“Holds for” on page 22](#) 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 22](#) for more information.

- Insert any parameters that the user will need to provide. See [“Parameter details” on page 29](#) for information.

6. Define the details for the template parameters.

- Define the prompts that describe each parameter to the FileNet BAM Dashboard user. See [“Prompts” on page 29](#) 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 22](#) 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 30](#) for details. This option is not available for templates based on cubes.

9. Define the rule template properties.

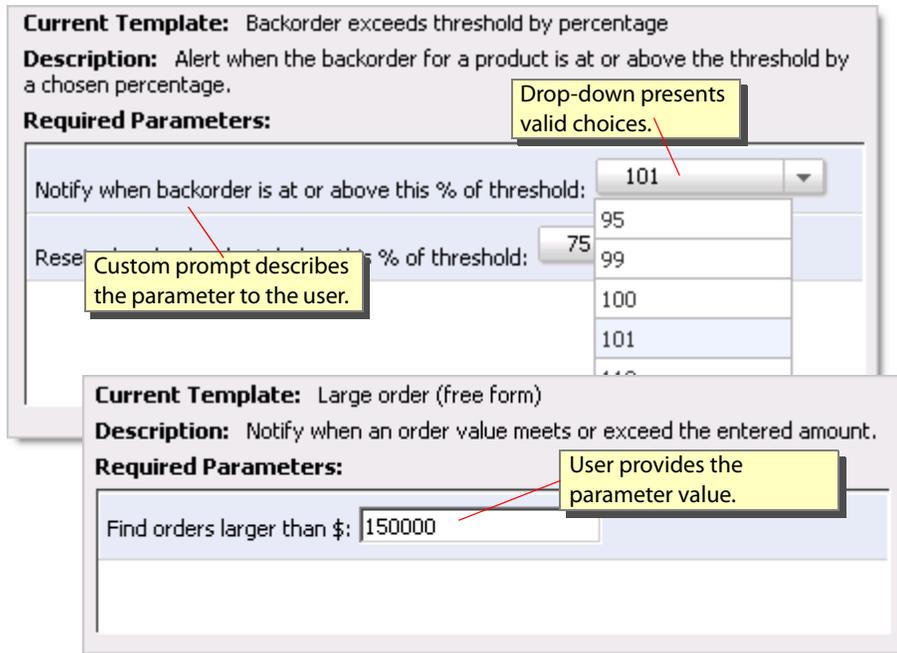
- Provide a name and optional description. Use the description to describe the template to the FileNet BAM Dashboard user.
- (Optional) Specify the importance level of the alert message.
- (Optional) Specify whether or not to consolidate multiple alerts from the same event. See [Consolidating multiple messages](#) 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*.

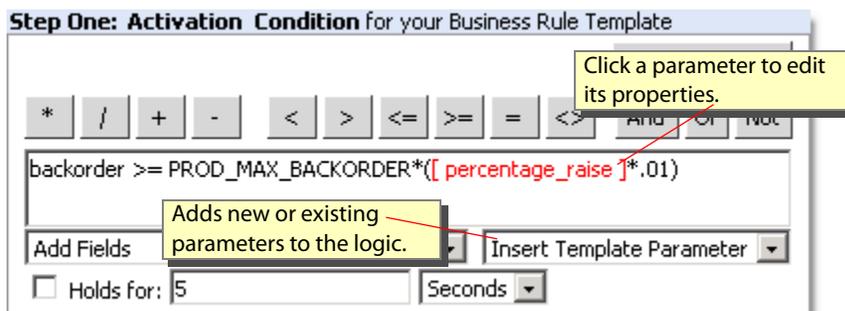
10. Save the template to make it immediately available to FileNet BAM Dashboard users.

Setting parameters

Parameters are the values that the user provides in the FileNet 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.

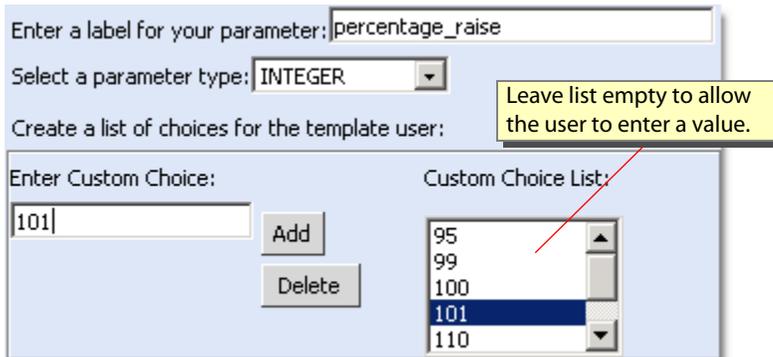


In the Rule Template editor, add parameters to the condition logic with the Insert Template Parameter option. You can insert a new parameter or choose one already defined for the template.



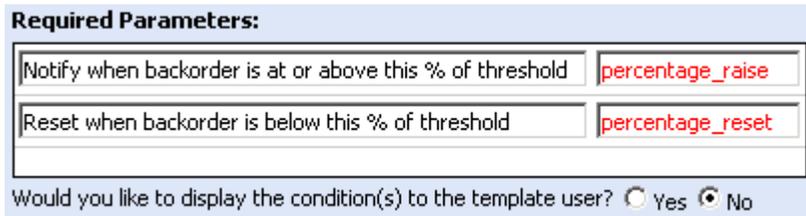
Parameter details

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.

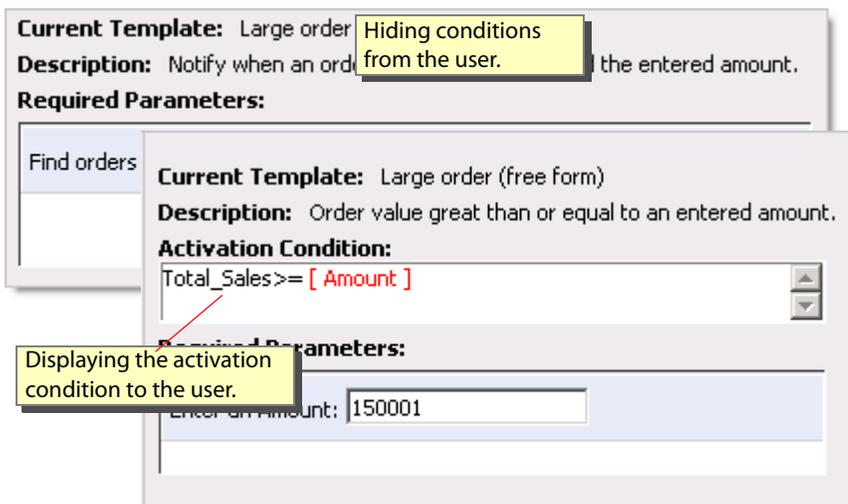


Prompts

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.



In addition to the parameters and prompts, you may choose to display the rule condition(s) to the user.



Working with reportlets

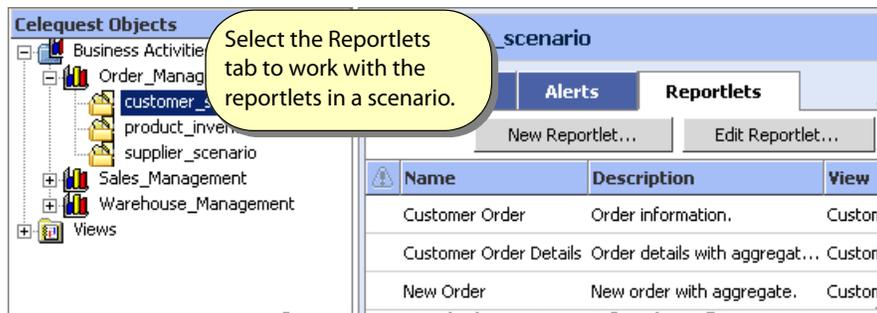
Reportlets describe the contents of a view and present that information in a report that is either attached to an alert message, or sent to an external system. Frequently reportlets provide information about an event that puts the event into context. For example, when an inventory is low for a product 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, such as Business Objects or Cognos. 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 67](#) 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.



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.

For Excel worksheets, you also need to define the following:

- The active report to use (if any)
- The name of the worksheet
- The address or cell location on the worksheet to insert the view data

For information about working with spreadsheet templates, see [“Managing active reports” on page 33](#).

Receiving Metrics

FileNet Business Activity Monitor generates metrics about your business' activities. *Metrics* are measurements taken over time that monitor, assess, and communicate vital information about an activity, and are presented in a spreadsheet. The Excel Dashboard tab in FileNet BAM Workbench is where you view and pick the metrics you wish to receive, and specify how the metrics get populated in the spreadsheet.

NOTE: To view metrics, you must have Microsoft Excel installed on your computer. To use Real-time charting, you need Microsoft Excel 2002, or later version.

This discussion covers these metrics topics:

[“Understanding active reports” on page 32](#)

[“Managing active reports” on page 33](#)

[“Managing ad-hoc queries” on page 34](#)

[“Managing real-time charting” on page 35](#)

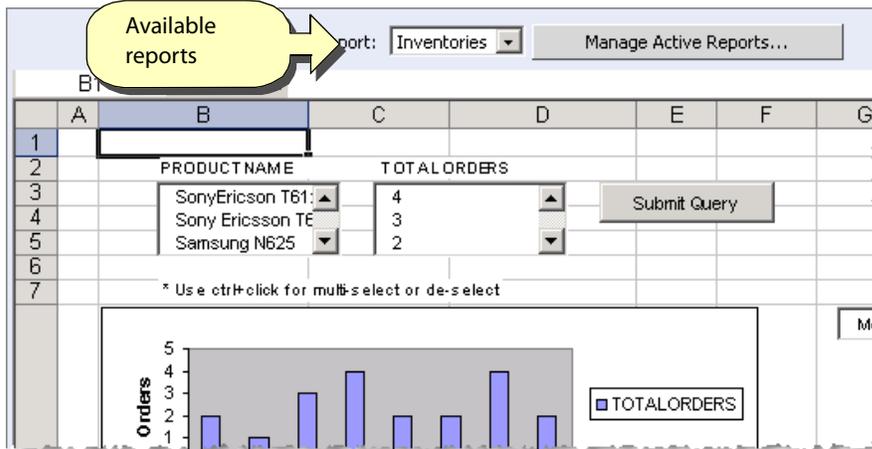
[“Troubleshooting active reports charts” on page 37](#)

Understanding active reports

Each set of metrics is defined by an *active report* that identifies the metrics and tells how to format the presentation. You may have any number of active reports, each representing a different sets of metrics. There are two types of active reports:

- Real-time charts that update automatically whenever the underlying view updates. (Requires Microsoft Excel 2002, or later version.) For information about this report, see [“Managing real-time charting” on page 35](#).
- Ad hoc queries display charts and data that update when you request it. Further, you may filter the query results that the chart uses. For information about this report, see [“Managing ad-hoc queries” on page 34](#).

Select the metrics to show by picking the report name from the list of available reports.

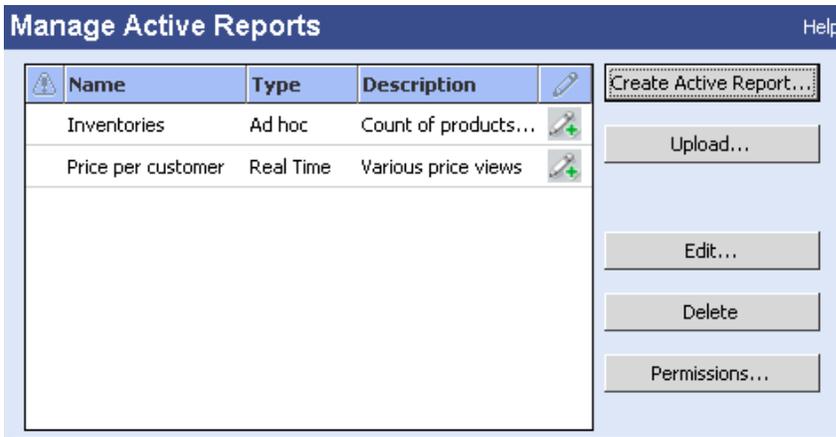


NOTE: When the browser opens the spreadsheet, it might prompt for permission to download an .xls file. Select **Open** to do so. See [“Troubleshooting active reports charts” on page 37](#) for details.

Create and edit reports, and declare access permissions with the Manage Active Reports option. See [Managing active reports](#) for details.

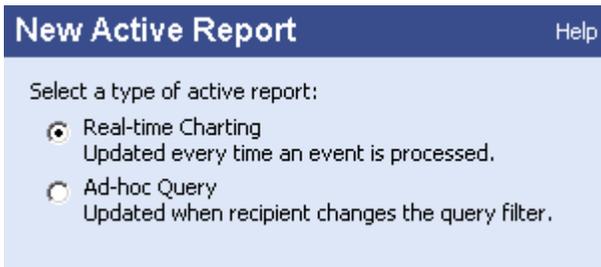
Managing active reports

The Manage Active Reports dialog lists all the active reports available to you; is where you create, edit, and delete reports; and is where you assign access permissions to the reports. You may also copy active report spreadsheets from your computer and install them as reports available to yourself and others.



For information about access permission, see Permissions.

When you create a new report, you select which type to use. For detailed information about the report types, see [“Managing real-time charting” on page 35](#) and [“Managing ad-hoc queries” on page 34](#).



Managing ad-hoc queries

Ad-hoc queries display charts and the view data that the chart is drawn from. Define the view on the Data tab and the chart on the Graph tab. The display updates when the user manually requests an update.

In the active report, the spreadsheet shows the data from the named business view. Note that you can increase the count of view rows to include in the report by increasing the Maintaining events in a stateless view setting. For example, defining ten events presents up to ten rows of events in the report.

You can filter the view data by choosing values to display.

To filter the view data:

- On the **Data** tab, select one or more columns that can be used to limit the data to include.

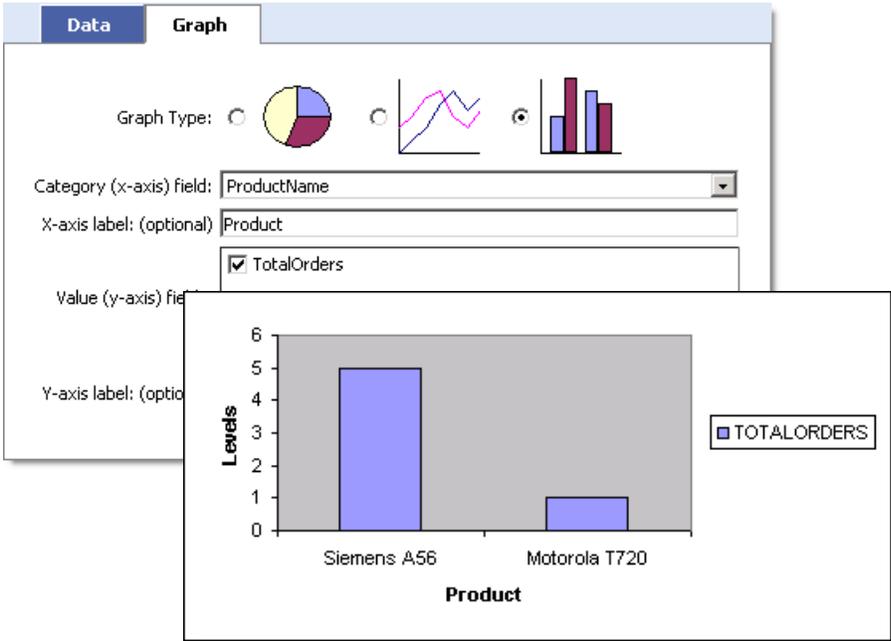
For example, choosing ProductName and TotalOrders in the illustration below allows you to later limit the report to specific inventory levels or products only. The report is currently limited to two products.

The screenshot shows the FileNet BAM Workbench interface. The **Data** tab is active, displaying a spreadsheet with columns for **PRODUCTNAME** and **TOTALORDERS**. A dropdown menu is open for **PRODUCTNAME**, showing the following options: **SonyEricson T61z**, **Sony Ericsson TE**, and **Samsung N625**. The **Graph** tab shows a bar chart titled **Orders** with **Products** on the x-axis and **Orders** on the y-axis. The chart displays two bars: one for **SonyEricson T61z** with a value of 4, and one for **Samsung N625** with a value of 2. A yellow callout box labeled **Filtered view details** points to the filtered data in the spreadsheet.

PRODUCTNAME	TOTALORDERS
SonyEricson T61z	4
Sony Ericsson TE	3
Samsung N625	2

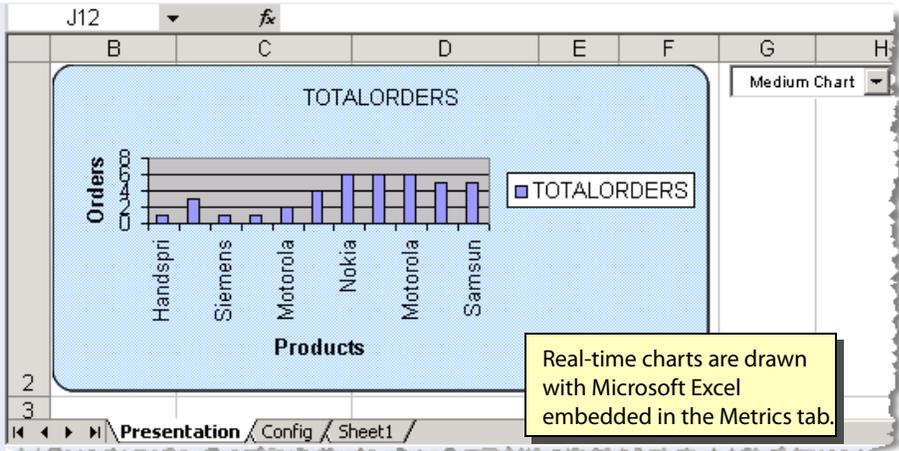
The Graph tab defines the chart properties including the type of chart, source fields, and axis labels. You must identify the category field that provides the x-axis values. The illustration below shows two products,

each from their own row in the view. You must also identify the values to chart. The chart below graphs TotalOrders. The labels are optional. If you don't define a label, the chart shows the column name.

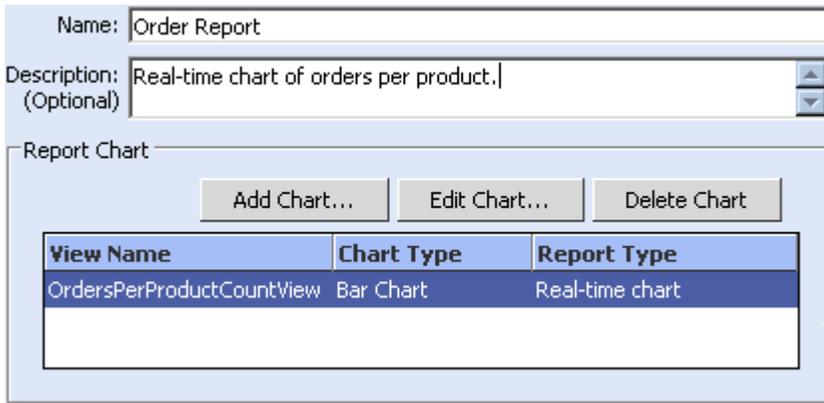


Managing real-time charting

Real-time charts use Microsoft Excel 2002, or later version, to automatically draw updated charts in the Metrics tab whenever the underlying view updates.

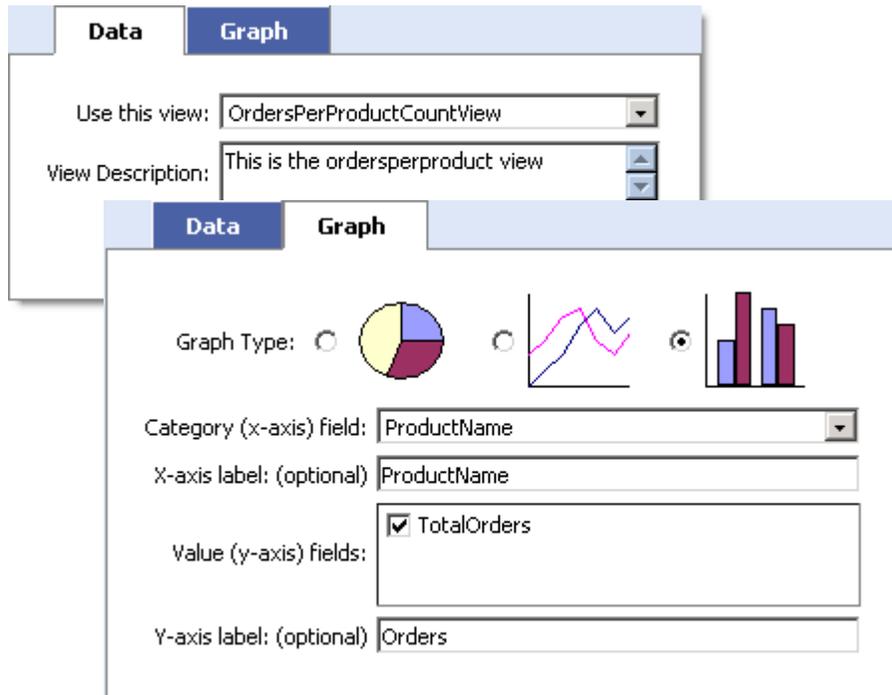


The Real-Time Report Manager is where you add, edit, and delete real-time active reports. Each report you have access to appears in the list of Report Charts. A real-time active report may have multiple charts on the same report.



To add or edit a chart:

1. In the Real-Time Report Manager, click **Add Chart** or **Edit Chart**.
 The Active Report Chart Editor displays.
2. To specify view and add a view description, click the **Data** tab.
3. To select a graph type and its parameters, click the **Graph** tab.



Troubleshooting active reports charts

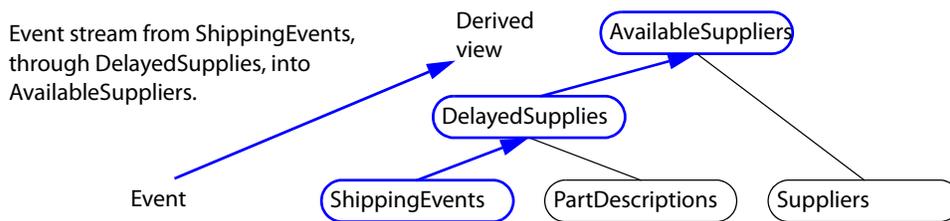
If you encounter problems viewing active reports, consider these situations:

- If you receive a File Download warning similar to “Some files can harm your computer” regarding an .xls file when loading the active report, your Internet Explorer browser is generating this message. Load the active report by choosing Open. Further, you can suppress this warning in the future by turning off the “Always ask before opening this type of file.”, or by changing your browser’s security settings.
- If Microsoft Excel warns that the file contains macros, and that macros may contain viruses, select Enable Macros to load the active report. Further, you can disable this message by changing the Microsoft Excel Security Level setting to “Low” in the Tools>Macro>Security menu.
- For real-time charts, you need Microsoft Excel 2002, or later. Otherwise, the spreadsheet displays a #NAME? error and the formula shows =RTD("RTD.Celequest",...).
- FileNet Business Activity Monitor server must have a JDK installed and in the system path.
- The Visual Basic runtime-error message “DataObject:GetTextinvalid FORMATETC structure” occurs when FileNet Business Activity Monitor RTD server is not registered as a COM server. This should happen automatically when the server starts up.
- Do it manually by running vcRTD -R from the command prompt on the server.

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.

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



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

NOTE: Views created in the FileNet 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 FileNet BAM Workbench.

In this Chapter::

[“Working with the FileNet BAM Workbench” on page 39](#)

[“Working with events and context tables” on page 42](#)

[“Working with views” on page 46](#)

[“Working with cubes and dimensions” on page 59](#)

[“Working with UDFs and JAR files” on page 65](#)

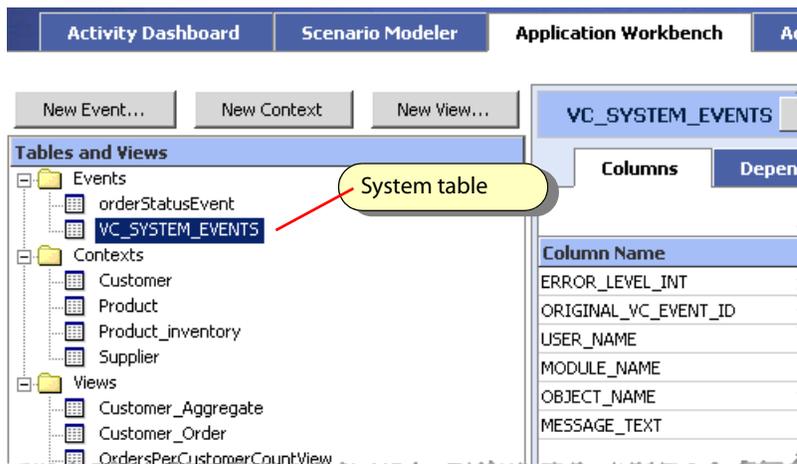
[“Working with external links” on page 67](#)

[“Working with process definitions” on page 68](#)

Working with the FileNet BAM Workbench

The left side of the FileNet 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 FileNet Business Activity Monitor. (See “Monitoring the logs” on page 92 for more information about this event.)



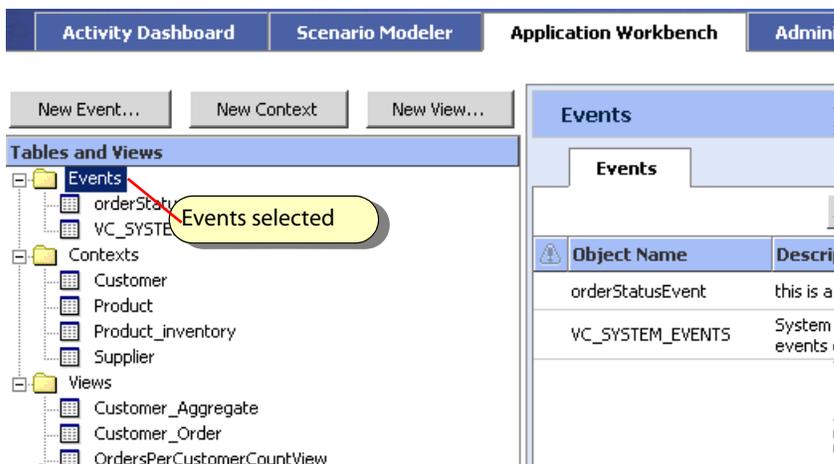
To create a new object:

- 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 10 for a description of the Status indicator.

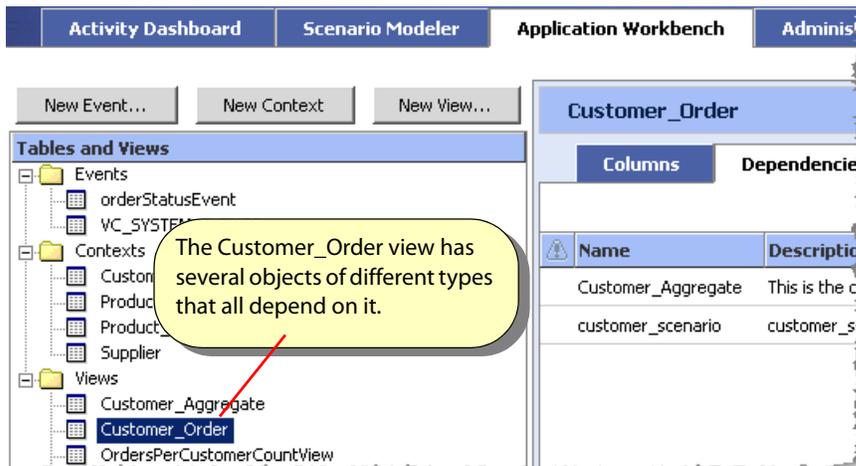


To display object details and access the object definition:

- 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.



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 10 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 FileNet 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 14](#) 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 this:

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

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

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:

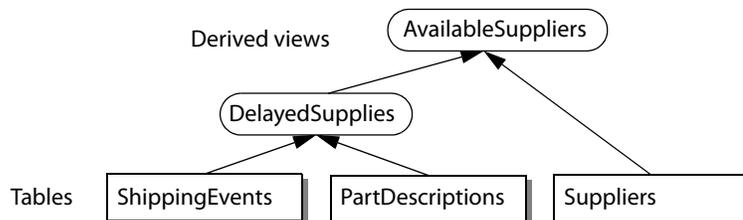
1. Select one of the event, context, or view objects to be the left-side of the join.
2. Click the **Join Relationships** tab.
3. Click **Add Join...**
4. Select the Context table to be the right-side of the join.
5. On the New Join dialog:
 - Name the join relationship.
 - Include a description of the 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 context are the data that drive FileNet Business Activity Monitor.

- Events feed FileNet Business Activity Monitor 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.
- Context support event processing and provide meaningful information about events. Context 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 FileNet BAM Workbench, and the tables get their information from agents defined in the Administration Console. (See “[Viewing the agents list](#)” on page 74 for details.) This illustration 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 FileNet 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 for details.

Working with events

The Events folder in the FileNet 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 FileNet Business Activity Monitor activity. This stream is used by system administrators to monitor the system. See [“Understanding logging” on page 87](#) for details.

To create an event table:

1. Select **New Event**.
2. Select the type of event. Usually you can pick Single event, unless you are merging multiple event sources into one stream.

In the later case, pick Consolidated Event. See [“Working with consolidated events” on page 45](#) for more information.

3. Select the type of event source.

Only those types with previously defined agents will be available. All agents are defined in the Administration Console. See [“Viewing the agents list” on page 74](#) for information about creating agents.

4. Identify the agent you want to use. Additionally, 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” on page 72](#) for complete details. Optionally, you can identify a sample data to load to assist with the 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 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 in Creating event tables.

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 FileNet BAM Workbench object tree lists all context tables that you have access to. Click the Context 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 74](#) 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](#) 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, etc. 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](#) 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.

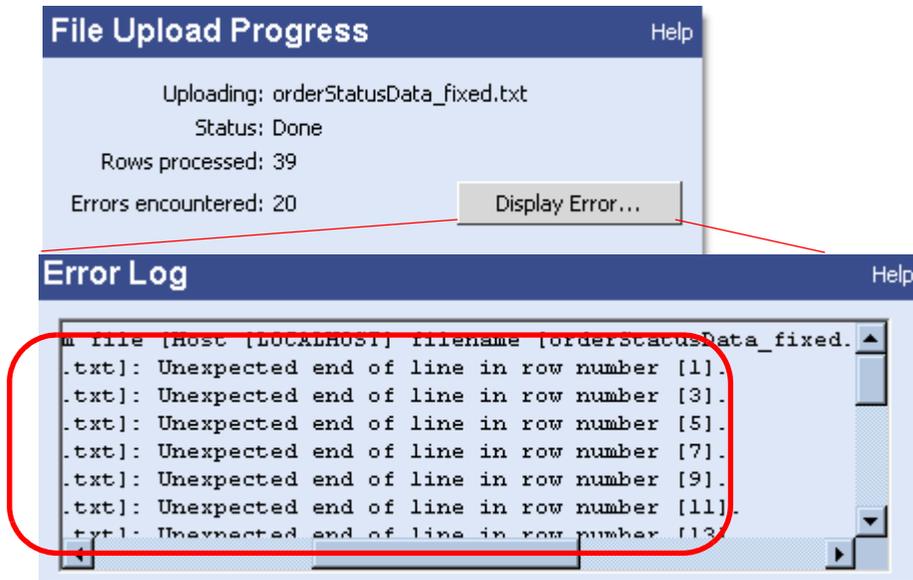
To upload event text into an event stream:

1. Select the event in the event tree in the FileNet 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.

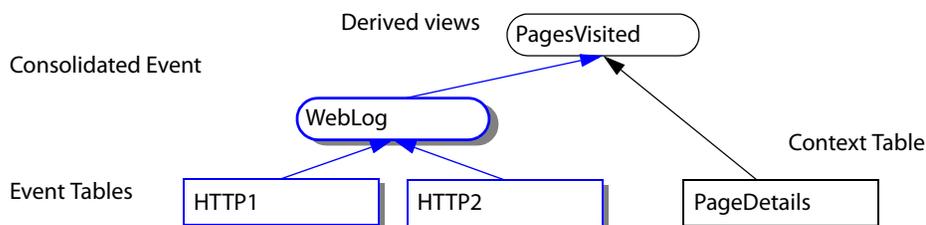
The upload is complete when the Status indicator says “Done”. If the source file contains erroneous data, the Display Error button activates. Click the button to see the descriptions of the errors in the Error Log.



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 combing events from multiple, 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 this 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 FileNet BAM Workbench. See Working with events,” above, 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.
- You cannot be a stateful view. When choosing the source for the view, stateless views are not available:

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

Select tables to consolidate:

<input type="checkbox"/>	Name	Type	Description
<input type="checkbox"/>	WarehouseQtyChangeFF	Event	Warehouse change in qu
<input type="checkbox"/>	WarehouseQtyChangeHTTP	Event	Warehouse change in qu
<input type="checkbox"/>	30DayOrders	View	Tracks order averages f
<input type="checkbox"/>	8WeekOrders	View	Tracks orders totals for
<input type="checkbox"/>	8WeekOrdersAvg	View	Averages orders for the
<input type="checkbox"/>	InventoryChangeDetails	View	Detailed information abo
<input type="checkbox"/>	OrderChangeDetails	View	Wrap-up of an order ev

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 FileNet Business Activity Monitor. 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 FileNet 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.
- And more.

NOTE: You can also create views in the Scenario Modeler, but those new views are dependant on join relationships defined in the FileNet BAM Workbench (see “[Viewing join relationship definitions](#)” on page 41 for details). Views created in the FileNet 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 40](#) for details).
- Join relationships for this view (see [“Viewing join relationship definitions” on page 41](#) for details).
- Reference data for plans on this view ([“Viewing plan relationships” on page 57](#)).
- Current values in the view (if any).
- Access filters applied to this view, as described above.

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

- Creating and editing views in the Application Workbench
- [“Creating view fields” on page 49](#)
- [“Working with clauses” on page 53](#)
- [“Maintaining events in a stateless view” on page 55](#)
- [“Persisting view data” on page 55](#)
- [“Displaying SQL expressions” on page 56](#)

Creating and editing views in the FileNet 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 FileNet 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 FileNet BAM Workbench:

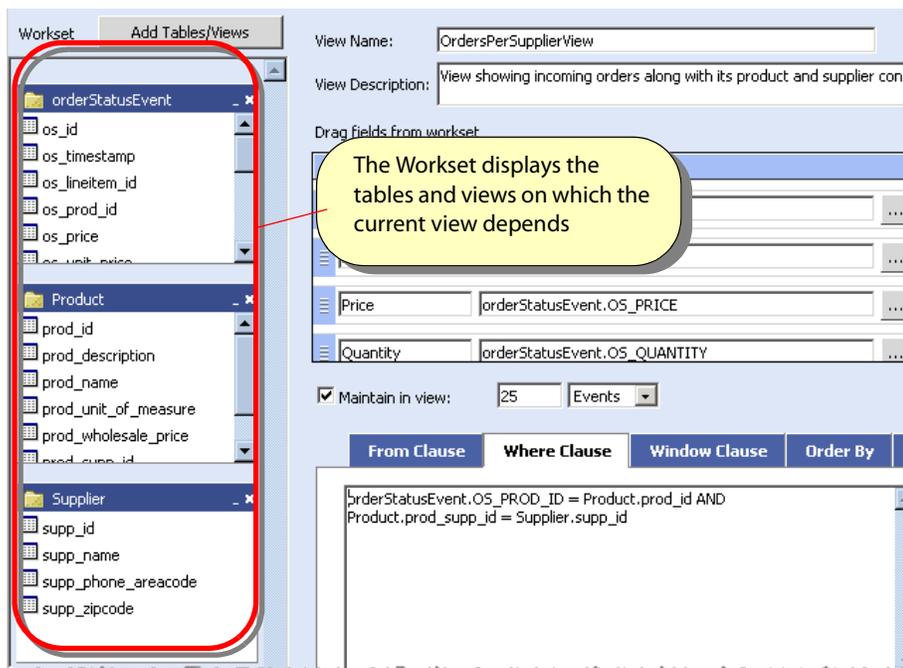
1. Select the view's name in the object browser
2. Click **Edit This View**.

3. Change the definition in the View Editor.

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

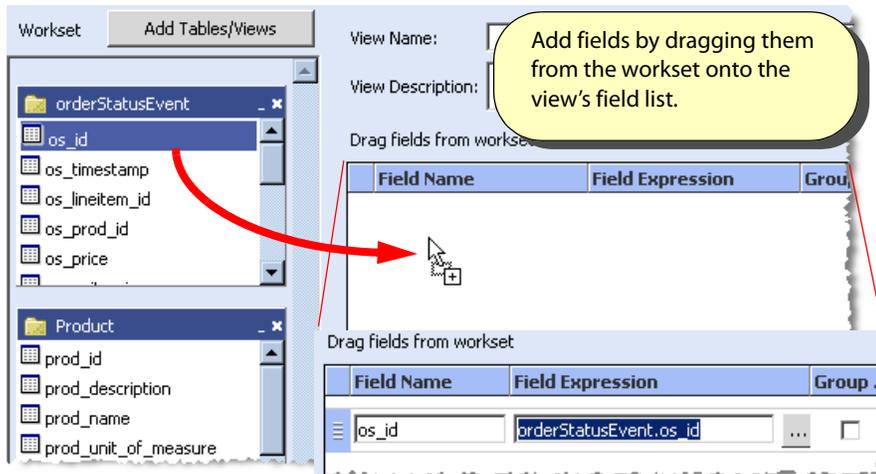
Workset

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.



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.



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



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 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. Here 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 these:

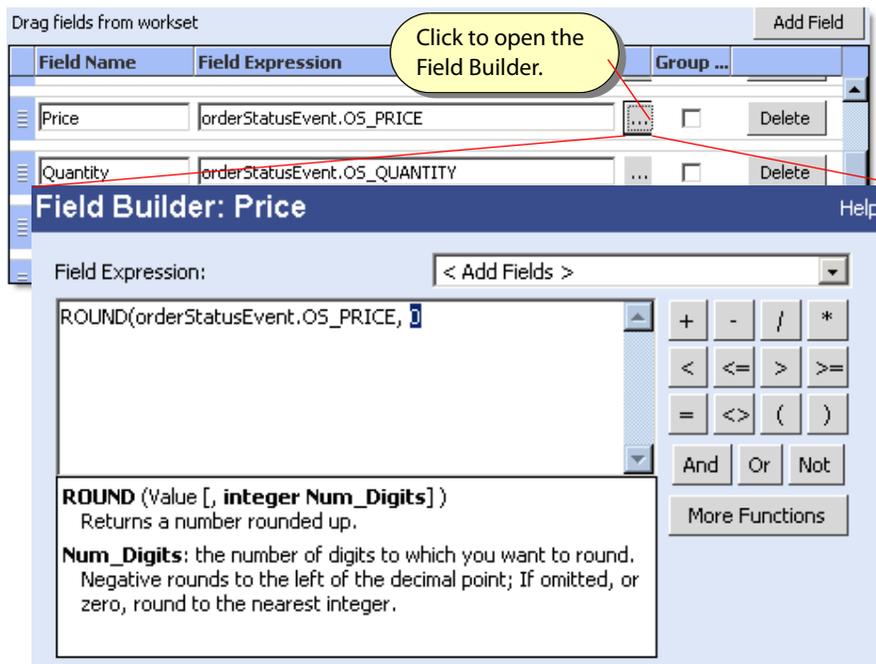
- `DISPLAY_MONEY(AVG(OS_PRICE))` formats the average price as a currency value, such as '\$1,234.56'.
- `CAST(((OS_PRICE/OS_COST)-1)*100 AS DECIMAL(5,0))||'%'` determines the percentage markup on the price of an item over its cost, strips out the decimals, and concatenates a percent symbol. The result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

NOTE: To learn more about functions and expressions, see Functions.

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.

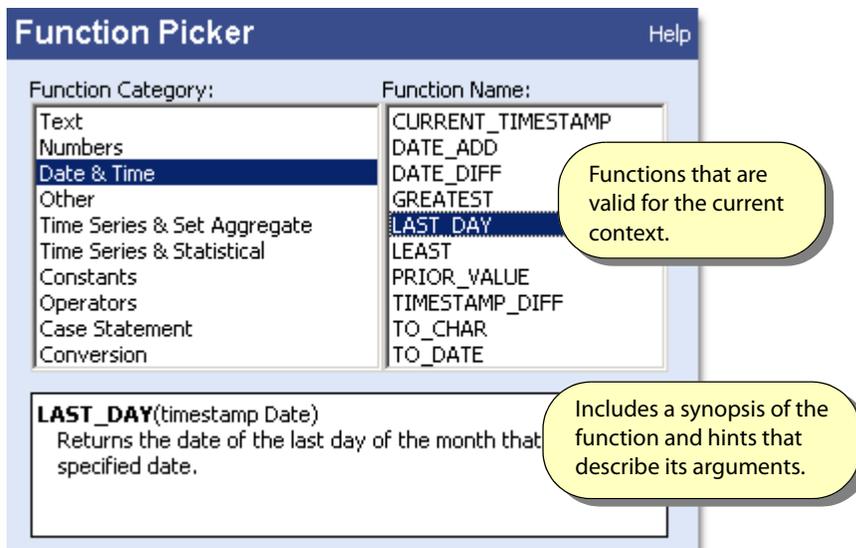


With the field builder you can:

- 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.



For detailed descriptions about the click **Help** on the dialog or see Functions.

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 this:

```
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, like this:

```
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

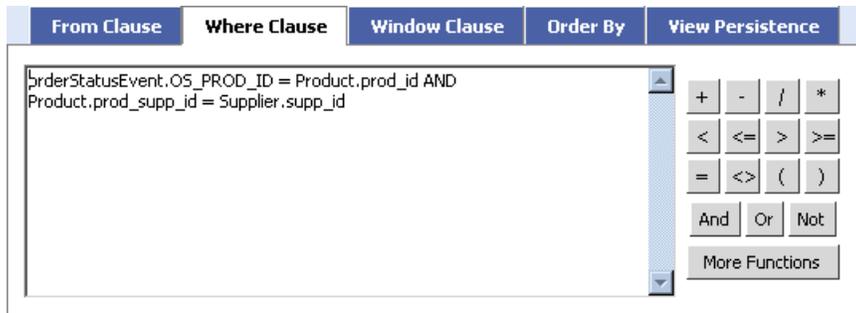
NOTE: When you use the Group by option, every field must either be part of the Group by or its definition must include a set function, such as SUM() or AVG().

For detailed information about Group by semantics, see GROUP BY clause.

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 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 55 for details.



NOTE: These tabs correspond to the clauses in the C-SQL SELECT statement: FROM clause, WHERE clause, and . 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. Here is an example:

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

For a detailed discussion of join relationships, see FROM clause.

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, 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 — that satisfies the view condition — to maintain. 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 for a means of performing aggregations on sets of recent events.

Persisting view data

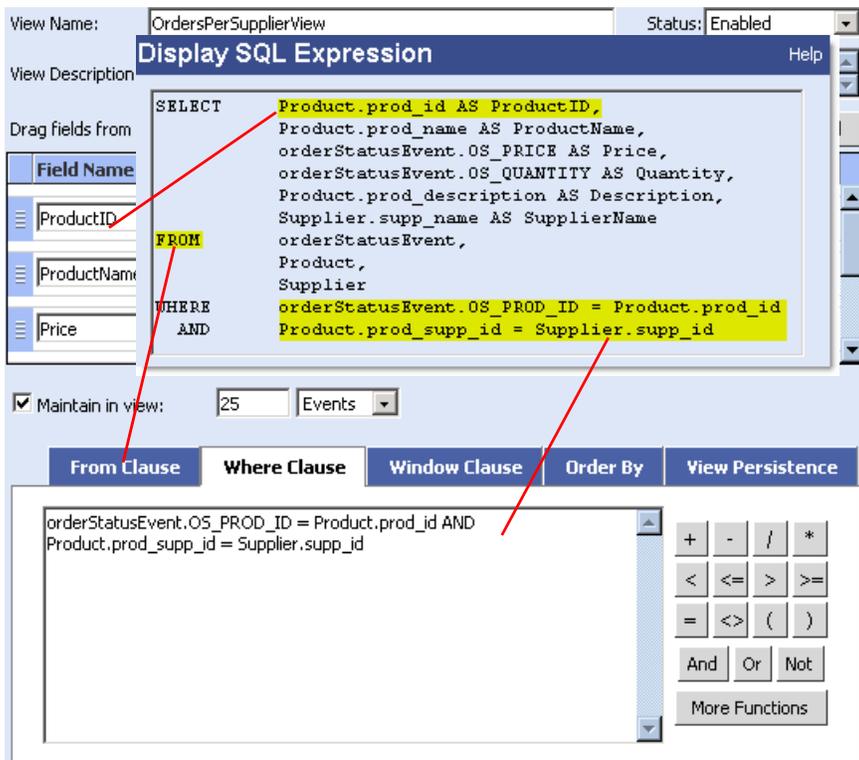
FileNet Business Activity Monitor can persist business view data to an external RDBMS for future reporting by third party tools. The information in the table is sufficient for the reporting tools to recreate a complete snapshots of the view. When persisting, the view information is written to a table 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.

Displaying SQL expressions

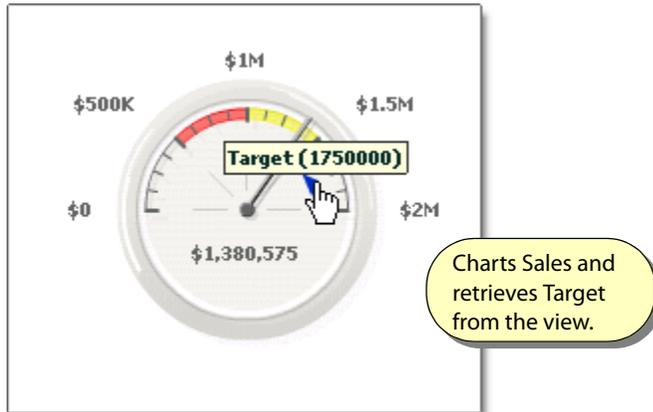
You create views with the view editor in FileNet BAM Workbench; however, behind the scenes, the view is defined as an SQL SELECT statement. From the editor you can see the SQL statement by choosing the Display SQL Expression option.



For a detailed description of the SELECT statement, see [SELECT](#).

Viewing plan relationships

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



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 the FileNet 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, this view has three references.

Reference Data		
Manage Reference Data		Delete Reference Data
Name	Description	Status
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. This 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:

	Column Name	Alias Name	Description
<input type="checkbox"/>	Qty		
<input type="checkbox"/>	Sales		
<input checked="" type="checkbox"/>	Target max	Target upper	
<input checked="" type="checkbox"/>	Target min	Target lower	Target minium range
<input checked="" type="checkbox"/>	Target	Target	Target sales

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

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	
<input type="radio"/>	No Plan Line		
<input type="radio"/>	Target upper	Plan	<input type="button" value="..."/>
<input type="radio"/>	Target lower	Plan	<input type="button" value="..."/>
<input checked="" type="radio"/>	Target	Plan	<input type="button" value="..."/>

References available to a dashboard object.

Or enter a numeric value for your Plan Line:

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):

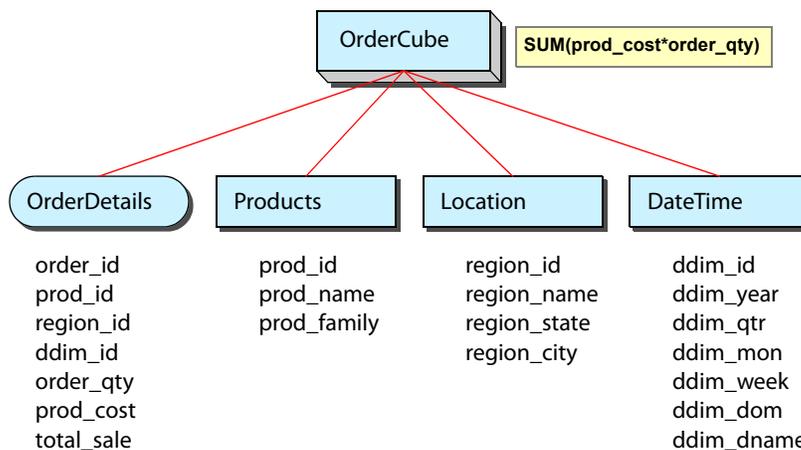
Cubes are similar to business views in that they aggregate event data, but they do so across different dimensions. The view that a cube aggregates is a *fact table*: a view or event table in an event stream that contains one or more columns to measure (aggregate), and which also contains columns that 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. This 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. Here 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. (Note that while the mathematical term “cube” implies three dimensions; however, 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](#) for information about controlling the cache.

To create a dimension:

1. Open the Workbench tab of FileNet 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. 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.

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 61](#) for details.

To create a cube:

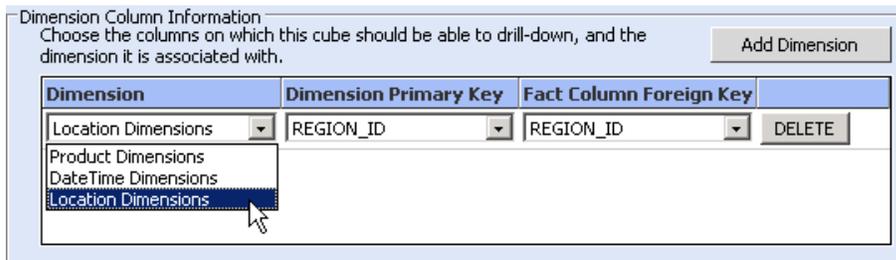
1. Open the **Workbench** tab of FileNet BAM Workbench, select the **Cubes** folder, and click **New Cube**.
2. Identify a name, and optionally provide a description of the cube.
3. Select the Fact Table that contains the data to measure, and which contains columns that identify the dimension elements.
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.

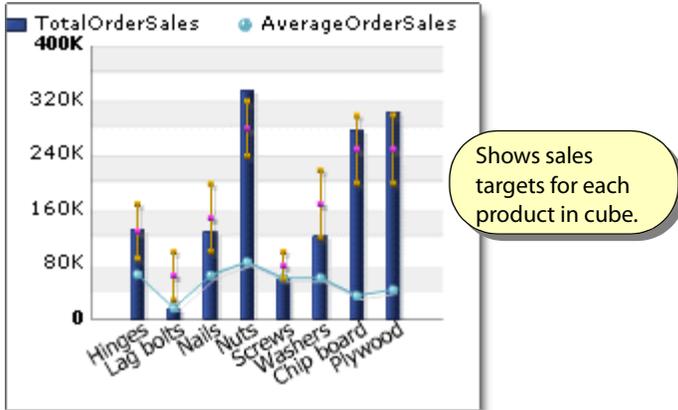


- Identify the key columns in the dimension and in the fact table. See Dimensions for more details about the keys.
 - The data type for the key in the fact table must be the same for the key in the dimension (context table). You cannot, for example, mix integer and decimal types; both must be either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



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 the FileNet 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, this cube has four references.

Dependencies		Reference Data	Access Filters	Results
		Manage Reference Data		De
Name	Description			
City sales target	City sales target.	Five references associated with the current cube.		
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.			

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. This is some of ten data from the Product_Targets context table that is an example:

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:

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
<input checked="" type="checkbox"/>	PRODUCT_MAX	Product upper target range	Maximum targets for products
<input checked="" type="checkbox"/>	PRODUCT_MIN	Product lower target range	Minimum targets for products
<input checked="" type="checkbox"/>	PRODUCT_TARGET	Product sales targets	Sales targets 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)

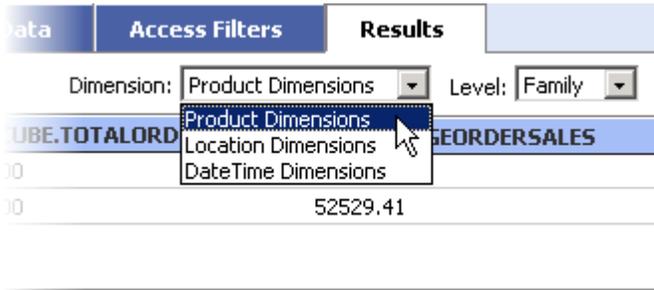
	Name	Type	
<input type="radio"/>	No Plan Line		
<input type="radio"/>	Target upper	Plan	<input type="button" value="..."/>
<input type="radio"/>	Target lower	Plan	<input type="button" value="..."/>
<input checked="" type="radio"/>	Target	Plan	<input type="button" value="..."/>

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 FileNet 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 like this:

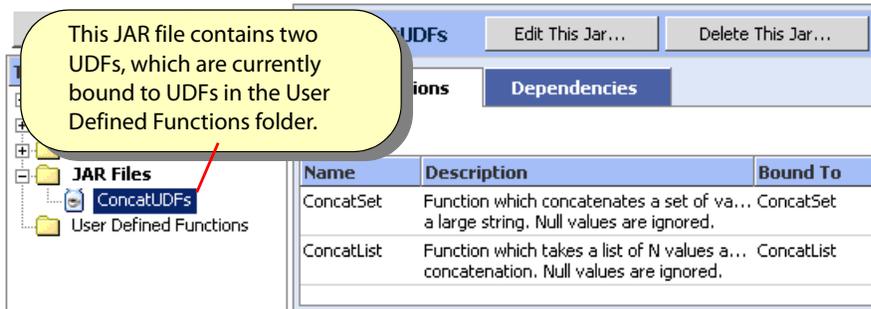
```
ConcatSet (Product .Name)
```

UDFs are small Java programs that take arguments and return a value, just like the internal FileNet Business Activity Monitor functions. See User-Defined Functions 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 FileNet Business Activity Monitor servers, and then identify the functions to make available.

JAR files folder

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



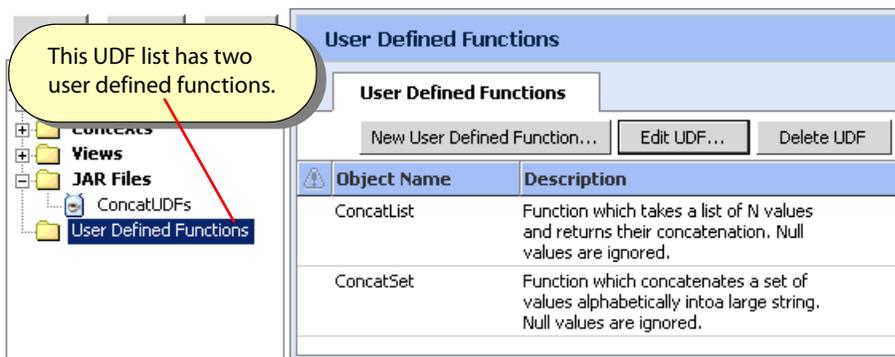
To upload a JAR file:

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 FileNet Business Activity Monitor. Select the folder to see the list of functions available, and to upload or identify new functions, or to edit or delete functions already in the list.



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 66](#), 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 Function**.
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, and which are inserted into Reportlets for communicating to external report mechanisms, such as Business Objects or Cognos.

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.
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.

External link URL

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 this:

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

See “[Working with reportlets](#)” on page 30 for information about creating reportlets in FileNet BAM Workbench, and Reportlets for detailed information about reportlets.

Creating an external link

To create an external link:

1. In the FileNet BAM Workbench, select the External Links folder.
2. Click **New External Link**.
3. Name and describe the link, and define the URL link.

Save the link and it is immediately available to reportlets.

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. The FileNet 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 FileNet Business Activity Monitor, see Processes.

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 FileNet 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 FileNet 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:

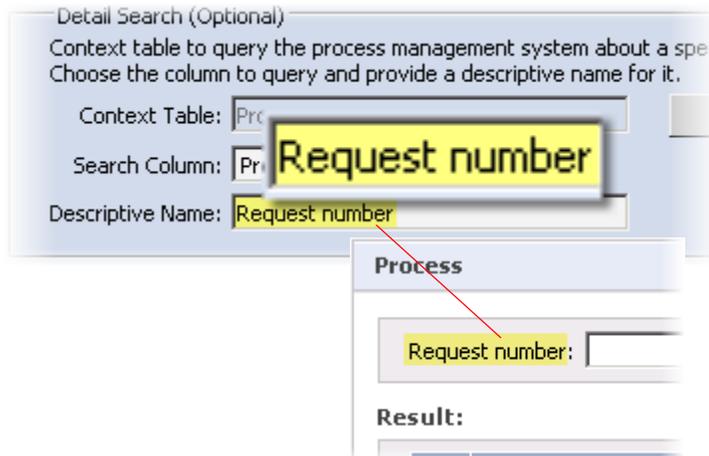
1. Open the Process Definitions folder in the FileNet BAM Workbench and select **New Process Definition**.
2. Name the object and upload the process definition file generated by the BPM.
 The process name from the definition file appears in the Process Name field after uploading the file.
3. Select the 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. This illustration shows the Step Name column in the Process Request Approval statistics view.



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

This illustration shows how the descriptive name appears in the search dialog:



5. Save the process definition.

You can now create process diagrams based on this definition.

Managing System Administration

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

In this Chapter:

[“Managing users” on page 72](#)

[“Managing roles” on page 73](#)

[“Viewing the agents list” on page 74](#)

[“Viewing the external processes list” on page 74](#)

[“Working with system settings” on page 75](#)

[“Importing/exporting metadata” on page 85](#)

[“Understanding logging” on page 87](#)

Managing users

Each user that interacts with FileNet Business Activity Monitor 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.

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.
 - 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.

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.

NOTE: For information about permissions, see Permissions.

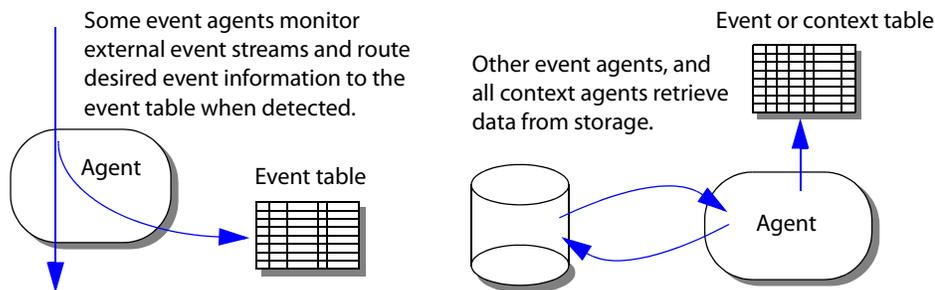
To administer roles:

1. In the Administration Console, click on **Roles** to see a list of the roles in the system. From this list you can do the following:
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 in the "[Managing users](#)" on page 72.
6. To see which roles you belong to, view your user account by clicking Account Settings.

NOTE: See Users for more information about user accounts.

Viewing the agents list

Agents are FileNet Business Activity Monitor 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.



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 agents attributes, see Agents.

- 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.

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 FileNet BAM Dashboard. FileNet Business Activity Monitor publishes the alert message, and any attached reportlets as an XML document to the service. See Web service external processes 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.
- 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 FileNet Business Activity Monitor is configured for normal operation.

To change system settings:

1. Open the **Administration** tab.
2. Click **Edit System Settings**.
3. Choose the tab of settings to alter.

Each tab's settings are described in detail below.

- Working with checkpoint and recovery, controls the system checkpoint and recovery mechanism.
- [“Working with custom branding” on page 74](#) identifies the logo images that appear on various pages throughout FileNet Business Activity Monitor.
- [“Working with execution throttling” on page 78](#) monitors and adjusts the rate that events are published and propagated into the view system.
- [“Working with the LDAP” on page 79](#) configures LDAP user authentication.
- [“Setting logging” on page 79](#) is how the system logs messages, and which messages to log.
- [“Setting logging levels” on page 81](#) are the messages to log, and which system components to generate the messages.
- [“Configuring memory monitoring” on page 81](#) configures how the system behaves when the amount of virtual memory on the server becomes low.
- [“Working with single sign-on settings” on page 82](#) allows users to bypass the login pages on FileNet Business Activity Monitor applications.
- [“Viewing SMTP configuration” on page 83](#) is the e-mail transport that the system uses to deliver e-mail notifications.
- [“Viewing system controls” on page 83](#) administer the operation of the installation.
- [“Viewing thread configurations” on page 84](#) changes how FileNet Business Activity Monitor processes information internally.
- [“Viewing user interface” on page 84](#) 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 Event properties” of Events 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 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 this 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 this option on the command-line that starts the application server:

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

Viewing checkpoint and recovery options

The 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 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 FileNet Business Activity Monitor 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 81 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.
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 FileNet Corporation.
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 FileNet Corporation.
Backlog Weight	Influences how previous throttling actions affect current and future actions. Do not change this setting without specific direction.

Working with the LDAP

In addition to allowing you to manually create users and user permissions, FileNet Business Activity Monitor enables you to import user information from supported LDAP providers. FileNet Business Activity Monitor 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 FileNet Business Activity Monitor, a default email UserProfile (called "PrimaryEmail") is created and which cannot be modified. When FileNet Business Activity Monitor synchronizes with the LDAP server, it uses this email 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 FileNet Business Activity Monitor does not recognize a role, it creates the role and assigns a set of default permissions. You can later modify the permissions as necessary.

For detailed information about LDAP configuration in FileNet Business Activity Monitor, see ["Setting up LDAP settings"](#) on page 73 in the *Installing and Configuring* documentation.

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 87. To identify which system modules generate log messages, see the description of the ["Setting logging levels"](#) on page 81.

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 FileNet Business Activity Monitor. May be redirected to a console. See "Understanding logging levels" on page 89 for details.
Summary File Log Level	General log file named celequest.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 89 for details.

Setting	Description
Detailed File Log Level	Use this log when trying to track a problem where you don't want to add to the messages in the Summary file. This file is celequest_detailed.log in the log directory. Typically this is set to OFF. See "Understanding logging levels" on page 89 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 FileNet Business Activity Monitor. 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 92 for more information.
Logging synchronicity	<p>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.</p> <p>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.</p>
Log file roll over period	<p>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.</p> <p>The choices are Monthly, Weekly, Biweekly, Daily, Bidaily, Hourly, and Minutely.</p> <p>Note: Bidaily logs are closed at noon and midnight.</p>
Additional Log4j properties	<p>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.</p> <p>For details about this file, consult the "Class PropertyConfigurator" documentation in the Log4j product from the Apache Jakarta Project (http://jakarta.apache.org).</p>

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 of the [“Understanding logging modules” on page 89](#). For detailed information about system logging, see [“Understanding logging” on page 87](#).

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 FileNet Corporation (see note below); rather, you should increase the memory on the server or off-load non-FileNet Business Activity Monitor 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 FileNet Business Activity Monitor applications. When enabled, users login only to their network and when they later connect to a FileNet Business Activity Monitor application, the access management software passes the user's name to the application in the HTTP request that opens the application in the user's browser. The application then verifies that the user has a FileNet Business Activity Monitor account before giving them access to the application.

All users must have an account in FileNet Business Activity Monitor, 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 FileNet Business Activity Monitor servers to affect and changes in this status.
Sign-off URL	Where to send the user's browser when they sign-off from FileNet Business Activity Monitor application. Normally the application sends the user to the application's login page.
Request header	<p>HTTP request header that contains the user's FileNet Business Activity Monitor user name. This header is provided in the request from the access management software when the user navigates to a FileNet Business Activity Monitor application. See the access management software documentation to determine the header name. For example, Netegrity 3.0 uses SMUSER.</p> <p>If the header is missing or invalid in the HTTP request, the user is directed to the application login page.</p>

Viewing SMTP configuration

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

Viewing system controls

System controls, located in the System Control tab, administer the operation of FileNet Business Activity Monitor 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 76 for details about checkpoints.
Restart Now	Stops and restart the servers.
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.
Shutdown Now	Stops FileNet Business Activity Monitor servers.

Viewing thread configurations

Thread configurations can be viewed in the Thread Configuration tab.

FileNet Business Activity Monitor uses execution threads that allow the system to processes views, alerts, rules, context queries, etc. 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 FileNet Corporation.

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 rows for view snapshot	Greatest count of rows to display when showing the contents of a view. Zero (0), the default, is unlimited.
Maximum days to retain alert notifications in the FileNet BAM Dashboard	Greatest count of 24-hour days to retain alerts in the FileNet BAM Dashboard. Timing begins when the alert is generated. For example, if the expiration period is 1 day, the alert is removed from the FileNet 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 FileNet BAM Dashboard

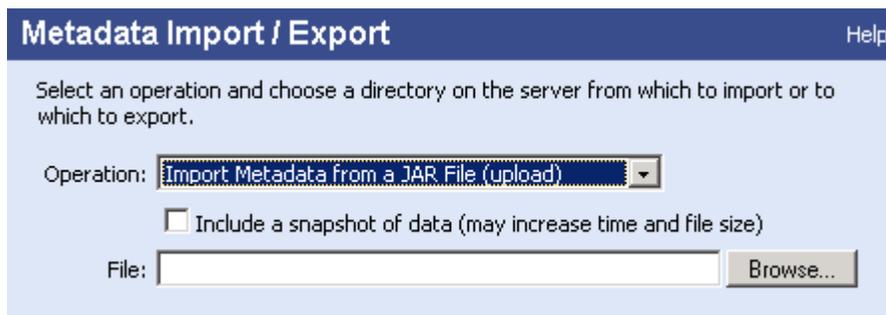
Importing/exporting metadata

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 either:

- XML files in a directory on the server, or
- 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 too that using this option can greatly increase the time to perform the operation. See [“Working with checkpoint and recovery” on page 76](#) for details about snapshots.



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 76](#) 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 FileNet Business Activity Monitor server. Further, the files are not encrypted. As such, all user account information are 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.
 - 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 FileNet Business Activity Monitor subsystems (*Understanding logging modules*) 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.



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.

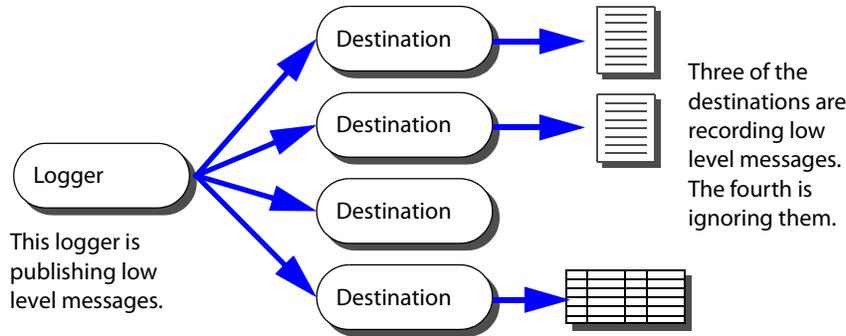
Messages have associated *logging levels* that indicates the importance and urgency of the message. Messages with higher levels indicate higher levels of importance. Using the levels you can tell the loggers which messages to publish, and you can tell the destinations which messages to include. Destinations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



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.

FileNet Business Activity Monitor 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.



Viewing logging destinations

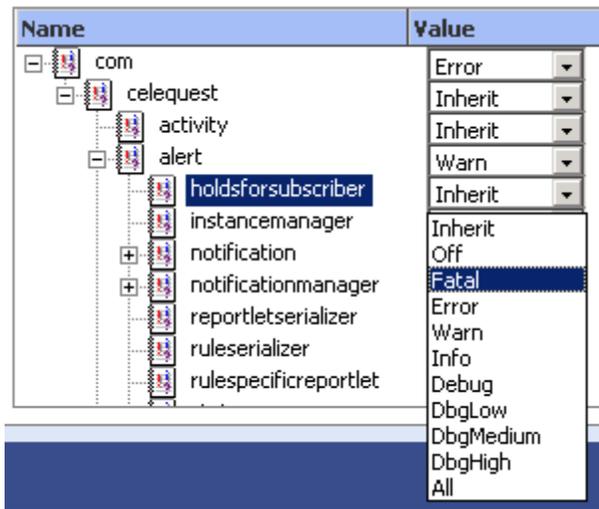
Logging destinations identify where log messages are recorded when the destination receives messages of the level it is configured to receive. The logging destinations are:

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 FileNet Business Activity Monitor. May be redirected to a console.
Summary File	General log file named celequest.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 don't want messages to go to the summary file. The file celequest_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 FileNet Business Activity Monitor. 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 92 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 FileNet Business Activity Monitor 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.



NOTE: Do not turn on Debug logging for modules without direction from FileNet Corporation. Turning on this excessive logging for high volume module 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 lead 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.
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.

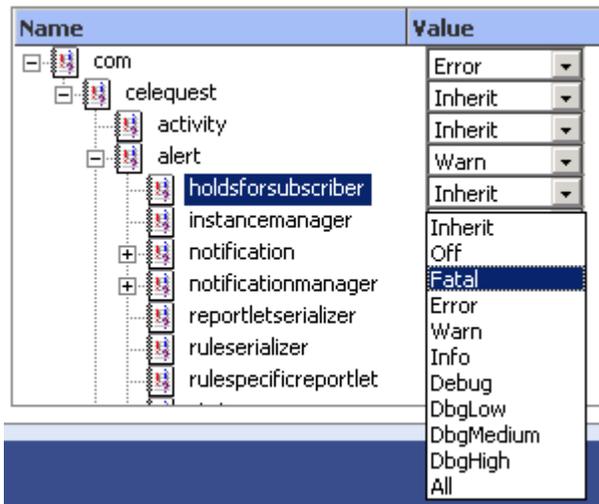
Do not turn on Debug or higher logging without direction from FileNet Corporation. Turing on this excessive logging for high volume module like system, query, or exec can rapidly flood your system and kill the servers.

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(s).
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.



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 the reports them in an alert to key administrators. When the System Appender Logging destination is capturing log messages, it writes them to the VC_SYSTEM_EVENTS event, which has these 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” 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 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 89](#).
 - 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

A

- acknowledgements 24
- active reports 31, 32, 33
 - ad-hoc queries 34
 - real-time charts 35
- Activity Dashboard
 - alert expiration 84
- ad-hoc queries 34
- administration 71
- Administration Console 71
- agents 74
- alerts
 - acknowledgements 24
 - creating in Scenario Modeler 19
 - deleting automatically 84
 - expiration from Activity Dashboard 84
 - Rule Filter of Event Data 25
 - testing 44
- Application Workbench 38
- authentication via LDAP 79

B

- backup 85
- business data modeling 8
- Business Objects reports 67
- business views
 - creating in the Scenario Modeler 18

C

- celequest.log file 88
- celequest_detailed.log file 88
- checkpoints 76
- Cognos reports 67
- columns
 - Scenario Modeler, editing in 15
 - View editor, editing in 49
- consolidated events 45
- context tables
 - Workbench, working with 44
- cubes
 - creating 61
 - dimensions, creating 61
 - plan relationships 63
 - results 65
- cubes, working with 59

D

- data modeling 8

- DEFAULTRECOVERYLOGGER_ file 77
- dependencies 40
- dimensions
 - creating 61
- dimensions, working with 59
- directories
 - log files 88
- disabled state 10
- Display SQL expression option 56

E

- e-mail transport 83
- enabled state 10
- errors
 - "DataObject GetTextinvalid FORMATETC structure" 37
 - "file contains macros" 37
 - "macros may contain viruses" 37
 - "#NAME?" 37
 - "Some files can harm your computer" 37
- event file for uploading 44
- events
 - consolidated 45
 - maintaining in the Scenario Modeler 17
 - uploading from text 44
- Execution Throttling tab 78
- export metadata 85
- _export.jar file 86
- external links 67

F

- fields
 - Scenario Modeler, editing in 15
 - View editor, editing in 49
- filtering views in the Scenario Modeler 16
- From clause in View editor 53

I

- import metadata 85
- invalid object 10

J

- JAR Files folder 66
- join relationships
 - creating 41
 - Scenario Modeler, in the 14

L

LDAP 79
 log4j properties 80
 logging 87

- destinations 88
- directory for log files 88
- levels 89
- levels, assigning 91
- log4j properties 80
- loggers 89
- module 89
- monitoring with rules 92
- roll over period 80
- synchronicity 80

 login

- bypassing 82

M

Maintain in View

- Scenario Modeler, events in the 17

 Maintain in View in Workbench View editor 55
 Memory Monitoring tab 81
 metadata 85
 metrics 31

- ad-hoc queries 34
- managing 31
- real-time charts 35
- receiving 31

 Metrics tab 31
 Microsoft Excel 31
 modeling business data 8

N

notifications

- deleting automatically 84

O

objects

- dependencies, determining 40
- invalid 10

 Order by clause in View editor 54
 ordering views in the Scenario Modeler 17

P

parameters, rule template 28
 performance

- event publication rate 78

 performance tuning

- memory low conditions 81

 Persist view data

- View editor, in 55

persistence 76
 plan relationships

- cubes 63
- views 57

 process definitions 68

- creating 69

R

real-time charts 35
 real-time data 31

- active reports 33

 recovery 76
 reference data

- views 57, 63

 relationships

- creating 41
- Scenario Modeler, in 14

 reportlets

- Business Objects 67
- Cognos 67
- data based on 25
- external (3rd party) 67
- Rule Filter of Event Data 25
- Scenario Modeler 30
- testing 44

 reports 32
 roles 73
 RTD 31

- active reports 33

 Rule Filter of Event Data 25
 rule templates 25

- details 25
- parameters 28

 rules

- creating in Scenario Modeler 19
- rule templates 25
- testing 44

 running state 10

S

Scenario Modeler 8

- working with 9

 scenarios

- creating in the Scenario Modeler 18
- testing 44

 shutdown 83
 sign-on 82
 single sign-on 82
 SMTP configuration 83
 spreadsheet active reports 31
 SQL, displaying a view's definition 56
 stand-by state 10

- stopped state [10](#)
- stopping the servers [83](#)
- system administration [71](#)
- system controls [83](#)
- system logging [87](#)
 - destinations [88](#)
 - levels [89](#)
 - levels, assigning [91](#)
 - loggers [89](#)
 - modules [89](#)
 - monitoring with rules [92](#)
- system settings [75](#)

T

- testing scenarios [44](#)
- text file, uploading to event stream [44](#)
- thread configuration settings [84](#)
- throttling execution [78](#)
- tuning
 - event publication rate [78](#)
 - memory low conditions [81](#)

U

- Upload Event File option [44](#)
- User Defined Functions folder [66](#)
- user interface
 - configuration settings [84](#)
- users
 - creating [72](#)
 - list of [72](#)

V

- VC_SYSTEM_EVENTS event [92](#)
- View editor [46](#)
 - fields [49](#)
 - From clause [53](#)
 - Maintain in View [55](#)
 - Order by clause [54](#)
 - Persist view data [55](#)
 - Where clause [54](#)
 - Window clause [54](#)
 - workset [48](#)
- view fields in the Scenario Modeler [15](#)
- views
 - creating in the Scenario Modeler [13](#), [18](#)
 - filtering in the Scenario Modeler [16](#)
 - maintaining events in the Scenario Modeler [17](#)
 - ordering in the Scenario Modeler [17](#)
 - results in Scenario Modeler [12](#)
 - Scenario Modeler, in [12](#)
 - SQL, displaying definition as [56](#)
 - testing [44](#)

- View editor [46](#)

W

- “what if” scenarios [44](#)
- Where clause in View editor [54](#)
- Window clause in View editor [54](#)
- workbench [38](#)
- workset [48](#)