



Note!

Before using this information and the product it supports, be sure to read the general information under “Notices and Trademarks” on page 71

Sixth Edition (December 2005)

This edition applies to Version 6 of IBM WebSphere Business Monitor product (5724-M24) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. You can send to the following address:

Cairo Technology Development Center (CTDC)
Business Integration Product Development
IBM WTC – Egypt Branch
Pyramids Heights Office Park, Building C10
Cairo – Alexandria Desert Road, km. 22
P.O. Box 166 El-Ahram, Giza, Egypt

Include the page number or topic related to your comment.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2005. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Administering WebSphere Business

Monitor 1

Administration overview. 3

WebSphere administrative console extension.	3
Monitor Server administration	3
Adaptive Action Manager administration.	4
Schema Generator administration	4
General administration	5

Administering Monitor Server 7

Server configuration	7
Business measures model management	7
On time Situations	8
Exception handling	10
Situation events description	12
Editing system properties.	13
Importing a business measures model	13
Unloading a business measures model	14

Administering Adaptive Action Manager 15

Adaptive Action Manager overview	15
Adaptive Action Manager components	16
Adaptive Action Manager handlers	16
Action logging service.	20
Action services templates.	20
Situation events binding	21
Action Catalog data export and import	23
Configuring Adaptive Action Manager properties	24
Configuring general properties	24
Configuring LDAP properties	25
Registering an action service.	25
Creating a new notification template	26
Creating a new web services template	26
Updating action services template definitions	27
Updating a notification template	27
Updating a Web services template	27
Binding a situation event with action services	28
Creating a new situation event binding	28
Adding a template to situation event binding	28
Updating a situation event binding	29
Using the Action Catalog database export and import tool	29

Managing schema generation 31

Overview	31
Database DDL artifacts	31
DB2 Cube Views metadata	32
Database replication scripts	33
Usage scenario	33
Configuring the Schema Generator	34
General configuration	34

State to Runtime databases replication	35
Runtime to Historical databases replication.	36
Generating schema	36

Performing general administration tasks 39

Starting and stopping enterprise applications	39
Managing the logging and tracing.	39
Configuring log files	39
Enabling the Trace option.	39
Changing log details levels	40
Increasing the databases connections pool size for Dashboard Client	40
Enabling and disabling the CEI store	41
Changing the data sources names	42
Changing DB2 Alphablox data source user name and password	42
Adjusting the maximum search results for the Organizations view.	43
Setting the work request queue size	43
Setting the destination queue size	44
Changing the exception destination for the destination queue	45
Configuring CEI Bus on a remote WebSphere Application Server cell	45
Configuring the remote CEI Bus	46
Resetting the remote CEI Bus Configuration	46
Setting the default DB2 Alphablox theme on an AIX System to the WebSphere Business Monitor theme	47

Administrative Console 49

Business measures model files	49
Notification data fields	49
Supported expression functions.	50
Data types.	59
Java primitive data types and precision	60
DB2 data types range and precision	60

Administration troubleshooting 63

Changing the user ID and password for the created authentication alias.	63
Changing user passwords on the Monitor Server components' machine	64
The imported business measures model runs successfully, but no data is stored in the databases	66
Business measures model import failure due to the usage of user defined metric name similar to a predefined metric name	67
DB2 start on AIX 5.2 and AIX 5.	68
Increasing shared memory segments	68

Notices and Trademarks 71

Administering WebSphere Business Monitor

Before you begin administering WebSphere® Business Monitor, you should become familiar with its administrative functions, key concepts, and tasks.

Administration overview

This overview describes the administrative utility of the WebSphere Business Monitor. It explains the administrative functions that the administration utility provides as an extension to the WebSphere administrative console.

Administrators can manage WebSphere Business Monitor components by using the administrative utility. This utility is added as an extension to the administrative console of the WebSphere Application Server as a group of plug-ins. After the successful installation of the components, the **Monitor Administration** node becomes available on the WebSphere Application Server administrative console, and each installed component is shown as a child node under it. Through these nodes, you can access and administer different functions using your login information (access rights) to the WebSphere Application Server.

WebSphere administrative console extension

You administer WebSphere Business Monitor through an extension to the WebSphere Application Server administrative console.

The administrative console of WebSphere Application Server is a Web-based tool that you use to manage the administrative activities of the IBM® WebSphere Application Server. The administrative console can be extended by adding plug-ins that help you to manage WebSphere Business Monitor and its components.

The plug-ins and their functions are:

- **Monitor Server plug-in:** You can configure the Monitor Server configuration properties and import and unload business measures models that are exported from the Business Measures editor in the WebSphere Business Modeler.
- **Adaptive Action Manager plug-in:** You can configure the Adaptive Action Manager properties, register and update action services, and bind situation events with action services.
- **Schema Generator plug-in:** You can specify the settings of the Schema Generator component to generate the required database scripts needed for creating the database tables, the DB2® Cube Views™ metadata, and the replication metadata XML file needed for the Replication Manager

The extension to the WebSphere Business Monitor administrative console is used to create a set of administration usage scenarios covering Monitor Server administration tasks, Adaptive Action Manager administration, and Schema Generator administration.

Monitor Server administration

The administrative functions of the Monitor Server are handled through the Server node under WebSphere Business Monitor extension on the WebSphere Application Server administrative console.

The **Server** node helps you to configure administrative functions that you use to edit configuration properties, such as the On-time situations checking interval, through the **System properties** page. As well, you can perform business measures model administrative tasks, such as importing a business measures model into

WebSphere Business Monitor and unloading a business measures model by deleting all of its versions from WebSphere Business Monitor.

Adaptive Action Manager administration

The administrative functions of the Adaptive Action Manager are handled through the Adaptive Action Manager node under WebSphere Business Monitor on the administrative console of WebSphere Application Server.

The Adaptive Action Manager is one of the key components in WebSphere Business Monitor. Adaptive is the characteristic that allows the Adaptive Action Manager to modify its behavior in response to changes in the target business processes and the character of the environmental stimulus. The Adaptive Action Manager receives situation events, which are notifications of business situations, emitted by the Monitor Server or other applications. It then selects the actions to take, based on predefined rules and policies set by the user. Finally, the Adaptive Action Manager invokes a selected action or a set of actions.

The Adaptive Action Manager helps you to perform the following:

- **Add or modify static configuration data** to a table in the action catalog database, which is the configuration table that contains the static information for setting up the SMTP Server and the LDAP Server. When you set up or update the general configuration properties and the LDAP Server configuration properties, the information is stored and updated in the action catalog table.

Important: You must restart the Adaptive Action Manager from the WebSphere Application Server administrative console after you update any values in the general configuration properties for your values to take effect.

- **Add or modify template definitions** When you create and modify templates, such as notification templates and Web Services templates, the information is stored and updated in the action catalog database. A template is information that is used by a particular action service (such as e-mail). When you create a template, you also bind a particular action service to the template.
- **Bind situation events** by linking the situation events with action services (target actions that you intend to invoke, for example, Web service). You do this by binding a situation-event name with a particular template. Because a template is already bound to a particular action service, this binds an event to an action service. Multiple action services (templates) can be bound to a single situation-event name.

Schema Generator administration

The administrative functions of the Schema Generator component are handled through the Schema Generator node under WebSphere Business Monitor on the administrative console of WebSphere Application Server.

The Schema Generator is a WebSphere Business Monitor component. It receives a .zip file which contains the business measures model only. This model is created by the Business Measures editor. The Schema Generator uses the business measures model to generate the scripts needed for:

- Creating tables for the State, Runtime, and Historical databases of WebSphere Business Monitor. These tables contain important information about the KPIs as defined in the business measures model. For more information about business measures model, WebSphere Business Modeler documentation

- Producing Cube Views metadata that is used for the dimensional analysis.
- Creating the data movement services artifacts to move information across the State, Runtime and Historical databases.

Before you can import the .zip file (containing the business measures model XMI file, the Cube Views metadata XML file) into WebSphere Business Monitor, you must generate the artifacts and then deploy the artifacts.

The Schema Generator node under Monitor Administration on the administrative console of WebSphere Application Server helps you to:

- Modify the Schema Generator settings which include general configuration and data movement services settings, such as the path and name of the business measures model XML file and the output directory.
- Generate Schema Generator artifacts for each business measures model.

General administration

You can perform administrative actions on WebSphere Business Monitor directly from the WebSphere Application Server administrative console.

For all general administrative actions, for example, starting and stopping components (such as, Monitor Server, Event Emitters, Adaptive Action Manager), you can use the WebSphere Application Server administrative console. From the Applications node, you select **Enterprise Applications**. The Enterprise Application page lists all installed applications. You can select enterprise applications for WebSphere Business Monitor components from the list, and apply the required action.

For WebSphere Business Monitor logging and tracing options, use the Troubleshooting node on the WebSphere Application Server administrative console to:

- Specify how the server will handle log records
- Select WebSphere Business Monitor to enable or disable a system log
- Specify where log data will be stored
- Choose a format for log content
- Specify a log detail level for components and groups of components

Administering Monitor Server

This section provides tasks to administer Monitor Server from the WebSphere Business Monitor Administrative Console.

Server configuration

The Monitor Server component has some system properties (configuration properties) that must be set with the proper values through the WebSphere Business Monitor administrative console.

You should define specific values for the two WebSphere Business Monitor system properties on the System Properties page. These properties are:

- **On-time situations checking interval (min.):** Specify the interval at which all on-time situations of the system are checked for occurrence. The interval is in minutes. For example, if the interval is 5, then every 5 minutes the on-time situations are checked for occurrence. The interval range is 1-1440 minutes.
- **Event processing batch size:** The number of events consumed and processed in a single batch.

Note: For better performance, set the **Event processing batch size:** value to correspond with the **Work request queue size** value for the **DeserializationWorkManager**. Initial setting of 100 is recommended for both values. Refer to the topic named Setting the work request queue size for details.

Important: After you edit one or more properties on the System Properties page, you must restart the WebSphere Application Server to apply these changes.

Business measures model management

You manage business measures models through the WebSphere Business Monitor administrative console. Managing includes importing a business measures model into WebSphere Business Monitor and unloading all the versions of a previously imported business measures model.

You can perform the following actions on a business measures model through the WebSphere Business Monitor administrative console:

- **Import a business measures model:** You can import a business measures model that has been exported by the Business Measures editor to the Repository database and the model execution engine. Before you can import a business measures model file, you must prepare the databases to receive the imported model data by creating the required database tables. This is done by running the Data Definition Language (DDL) script files that have been generated by the Schema Generator in the WebSphere Business Monitor administrative console. The imported business measures model file can be either for a new business measures model or for a new version of an existing business measures model.

Important: In order to import business measures models successfully to WebSphere Business Monitor that has been installed in a secured WebSphere Application Server environment with enabled global

security feature, you have to add the following properties with their values to the file *soap.client.props* in the location `<WAS_Home_Dir>\profiles\<Server_Name>\properties\` directory using any text editor:

- `com.ibm.SOAP.securityEnabled=true`
- `com.ibm.SOAP.loginUserId=<UserName>` (Where `<UserName>` is the user name of an authorized user that has access on the WebSphere Application Server in security mode).
- `com.ibm.SOAP.loginPassword=<Password>` (Where `<Password>` is the password of the authorized user).

For more details about this feature, you can refer to the section named *Configuring security with scripting* in the WebSphere Application Server documentation.

Important: The import may fail if the imported business measures models contains a user defined metric that its name is similar to any predefined metric name. Refer to the topic named Business measures model import failure due to the usage of user defined metric name similar to a predefined metric name for details.

- **Unload a business measures model:** You can unload a previously imported business measures model. The unload action will remove all versions of this model.

The unloaded model information is deleted from both the model execution engine and the Repository database. You can select to delete the model information from only the execution engine and keep the model data in the Repository database for dashboard reporting.

Important: You must restart the WebSphere Application Server after importing a new version of an existing business measures model, or after unloading a specific model that will be imported again.

Important: On the AIX® platform, the import of the business measures model could fail if CPU Guard is enabled. Depending on your system, you may want to disable CPU Guard before importing a business measures model.

On time Situations

On time situations are situations whose evaluation strategy is based on time. That means the situation triggers are evaluated and triggered at regular intervals that you defined in the business measures model, for example the situation is triggered every 30 minutes.

The on time situation triggering time is based on the Monitor Server clock time not on the clock time of the runtime engine where the process or activity instance on which the situation is modeled has been created. For example, if the process instance was created at 9:20 based on the runtime engine time, and the Monitor Server received the event of the process instance creation at 10:20 based on the Monitor Server clock, then the on time situation will be triggered and evaluated based on this Monitor Server time (10:20).

The Monitor Server evaluates if there is any on time situations that should be triggered periodically based on the value of the server property named On-time Situation Checking Interval. This property is configured through the System

Properties page in the WebSphere Business Monitor administrative console. The value of this property determines the frequency (in minutes) of checking the existence of any on time situations that should be triggered. If this interval is greater than the recurring duration for an on time situation, then the situation will effectively be triggered when the former interval has elapsed. However, if this interval was smaller than the recurring duration for the situation, then the latter will have precedence and the situation will only be evaluated when the recurring duration has elapsed. To illustrate, suppose the on time situation checking interval is 30 minutes, and a model has two on time situations defined, which have recurring durations of 10 and 50 minutes respectively.

If the system had started at 10 am, and the MC instance was created at 10:40 am, then at 11:00 the first on time situation will be evaluated and potentially fired. At 11:30, both the first and second on time situations will be evaluated.

The administrator can set the interval with the value that best suites his needs with regard to the performance. You cannot set the On-time Situation Checking Interval with a value greater than 1 day.

For example, providing that we have these values: There is an on time situation with a 30 minute recurring duration, and the first triggering time of this situation was on 10:20, the administrator set the On-time Situation Checking Interval property to 60 minutes, and the server started on 10:00, In this case, the on time situation will be evaluated is 11:00.

The triggering of an on time situation is also controlled by a gating condition that can optionally be modelled with the situation, for example an on time situation is triggered every 30 minutes if the value of a specific metric is true. Considering this with the example above, if the gating condition of the on time situation that its trigger time is at 10:50 was supposed to be evaluated to true, but since the checking interval was set to 1 hour, and for some reason, the gating condition of this situation was not true when it was evaluated at 11:00. If the Administrator sets the On-time Situation Checking Interval to 50 minutes then the situation would have been fired at 10:50.

Note that you may choose an On-time Situation Checking Interval that is greater than the recurring duration of the situation, if you are aware that the gating condition for this situation will not change frequently. Meaning that, the evaluation of the gating condition of the on time situation in the example above will not change from 10:50 to 11:00 and thus the situation will then be fired normally. This will thus boost up the Monitor Server performance to consider only those processes which have a fire time less than the server current time.

Moreover, if an on time situation has multiple recurring duration points in time at which the gating condition evaluates to true within the On-time Situation Checking Interval, then at most one firing of the situation will occur within this interval.

In Addition there may be multiple on time situations with a different recurring duration for each of them in the same process. The Monitor Server guarantees that the on time situation with the least fire time gets triggered first. This guarantees an accurate evaluation of an on time situation condition that may be affected by another on time situation.

Exception handling

The Monitor Server component produces three types of exceptions in WebSphere Business Monitor.

- **Soft exceptions:** Which are exceptions that are modeled by the user in the business measures model (such as correlation matches exceptions and parent not found exceptions). These exceptions are predetermined by the model and thus expected. Such exceptions will be logged by the Monitor Server and a common based event will be sent to the Adaptive Action Manager to be displayed in the Alerts view in the Dashboards Client to inform the administrator with the exception's occurrence. The description of this event is elaborated in the topic named *Situation Events description*. After such a soft exception, processing of other events is continued normally.
- **Hard exceptions (Runtime exceptions):** Which are exceptions that are thrown as a result of runtime errors while retrieving and processing the events of a business measures model. These exceptions are not predetermined by the model and thus are not expected. These exceptions are logged and traced in the log files and the events that caused these hard exceptions are rolled back with all their triggered maps and situations. In addition a common based event will be sent to the Adaptive Action Manager in order to take a proper action (e-mail, alert, cell phone message,...etc.) to the administrator to inform him with the exception's occurrence. The description of this event is elaborated in the topic named *Situation Events description*.

The rolled back event will be iteratively processed and rolled back in an infinite scenario, which may cause the blocking of the Monitor Server. The reason for this behavior is to avoid processing the events that come after the event that caused the exception leading to an out-of-order event processing, which will result to the loss of sequence of event processing.

Alternatively, you can prevent the Monitor Server from being blocked by any runtime exception by changing the Exception Destination for the destination queue `Monitor_Bus_Queue_Destination` which is used by the Monitor Server to **System** instead of **None**. This way the events causing runtime exceptions will be ignored. In this case, it is the responsibility of the administrator to configure the WebSphere Business Monitor to either be blocked when a runtime exception occurs, to preserve data consistency and event sequencing, or to ignore the event that caused the error to avoid the blocking of the server but allow data inconsistency and the out of order of events. Refer to the topic named *Changing the exception destination for the destination queue* for the detailed steps of how to change the exception destination for the destination queue.

A special case for this behavior is implemented for the hard exceptions caused by processing the on time situation. As long as these situations are generated and owned by the Monitor Server and are independent from the runtime engine events, then there is no need to treat these exceptions in the same manner by forcing the Monitor server to retry processing the event and block the system. In this case the exceptions caused by processing on time situation events are handled differently as follows: The processing of the on time situation event is handled within the batch event processing cycle transaction boundary. Thus given that the processing of the on time situation event threw an exception, the batch of processed events are rolled back. Then the monitor server resets the last fire time value such that when the next on time event is created, it will initialize again the last fire time to the current monitor time. This has the effect of delaying the on time situation event to the next on time situation event interval, hoping that the events that will be processed in between will eliminate the cause of error.

- **In doubt exception:** If, for whatever reason, the WebSphere Application Server crashes the state of some events will be set to in doubt. The Monitor Server cannot determine if in doubt events were successfully processed or not. On Startup of the Monitor server, in doubt events are checked. If found exception is logged and a common based event is sent to the Adaptive Action Manager in order to send a proper action (e-mail, alert, cell phone message,...etc.) to the administrator to inform him with the exception's occurrence. The description of this event is elaborated in the topic named *Situation Events description*. It is the responsibility of the administrator to determine if these events are to be processed again or to be deleted.

Situation events description

There are two types of situation events emitted by the Monitor Server and received by the Adaptive Action Manager. Both events types conform to the common base events specifications.

- **User Defined Business Situation Events** which are defined in the business measures model through the Business Measures editor. These situation events are generated due to a specific business situation that occurred while the execution of the model as designed by the user.
- **Monitor Defined Situation Events** which are the events that are created by the Monitor Server as a result of any exception (soft, hard, and in doubt exceptions).

The common created situation events contain a set of fields that are categorized to three groups: The first group is the fields that are mandatory to be exist in any common base event. The second group is the fields that are added to any event that should be sent to the Adaptive Action Manager. The third group is the fields that are added to the events that are created as a result of an exception (soft, hard, and in doubt exceptions). The events fields are described below:

The mandatory common base events fields

Field	Value	Comment
CreationTime	currentEvent.getCreationTime() = the creation time of the event being processed	
SourceComponentID.Application	BaseMonitoringContextInterface.AM_COMPONENT_APPLICATION = "Websphere Business Monitor Version 6.0"	
SourceComponentID.Component	BaseMonitoringContextInterface.AM_COMPONENT_COMPONENT = "com.ibm.wbimonitor"	
SourceComponentID.Subcomponent	BaseMonitoringContextInterface.AM_COMPONENT_SUB_COMPONENT = "com.ibm.wbimonitor.observationmgr"	
SourceComponentID.ComponentType	BaseMonitoringContextInterface.AM_COMPONENT_COMPONENT_TYPE = "engine"	
SourceComponentID.ComponentIDType	ComponentIdentification.COMPONENT_ID_TYPE_PRODUCT_NAME	
SourceComponentID.Location	getHostAddress() = Current [®] Machine IP Address or if no network interfaces present 127.0.0.1	
SourceComponentID.LocationType	ComponentIdentification.LOCATION_TYPE_IPV4	
Situation	"Monitor Situation"	
Situation.reportSituation	"External" "ecode"	
Severity	10-model outbound events 40- soft exceptions 70- hard exceptions and in doubt events	10- info 40- error 70- fatal
ElapsedTime	12000	
Priority	50-model outbound events 50- soft exceptions 70- hard exceptions and in doubt events	50- medium 70- high
ExtensionName	Type of Event	

The fields that should exist in any situation event that is sent to the Adaptive Action Manager

Attribute Name	Content	Example
BusinessSituationName	The name of the business situation. For User Defined Business Situation Events it is the name specified by the user. For the Monitor Defined Situation Events it is the predefined soft exception name.	UserSituation1 Or com.ibm.wbimonitor.ParentNotFound
ContextID	Represents the MCIID to be overridden by the Monitor server	1233344
ContextDef	Represents the monitoring context physical name to be overridden by the Monitor server	Ndsoijh29832498

The fields that exist in any situation event caused by an exception

Attribute Name	Content	Example
Business Measures Model Name	Fully qualified name of Business Measures Model ignoring hidden business measures models	BMM1, BMM2
Monitoring Context Name	Fully qualified Monitoring Context business name	MC1/MC2
Original Event	The Event that caused the exception. It could not be contained in a context data element nor an extended data element and thus added as an ANY element	
Exception Message	A Globalized exception Message	No Correlation Matches for event entry EVENTENTRY in context CONTEXT

For the monitor defined situation events, all possible fields will be populated by the Monitor Server. The following table lists the Monitor-defined situations names with the corresponding event attributes for each situation. The Monitor situation names will have a prefix "com.ibm.wbimonitor" to avoid name collisions with other existing Adaptive Action Manager business situations.

The Monitor-defined situations names with the corresponding event attributes for each situation

Situation Name	Business Measures Model Name	Monitoring Context Name	Original Event	ContextID	ContextDef	Exception Message	Action in Adaptive Action Manager
MultipleParentFound	✓	✓	✓				email
ParentNotFound	✓	✓	✓				email
NoCorrelationMatches	✓	✓	✓				email
MultipleCorrelationMatches	✓	✓	✓	✓	✓	✓	email
OneCorrelationMatch	✓					✓	email

The Monitor-defined situations names with the corresponding event attributes for each situation

Situation Name	Business Measures Model Name	Monitoring Context Name	Original Event	ContextID	ContextDef	Exception Message	Action in Adaptive Action Manager
IndoubtEventInMonitorQueue						✓	email
RuntimeException	✓	✓	✓	✓	✓	✓	email

Editing system properties

Follow these steps to set the WebSphere Business Monitor system properties related to event processing and on-time situations.

1. To access the **System Properties** page from WebSphere Application Server administrative console, go to **WebSphere Business Monitor** → **Server** → **Configuration** → **System Properties**.
2. In the **On-time situations checking interval (min)** field, type the interval (in minutes) at which all on-time situations of the system are checked for occurrence.
3. In the **Event processing batch size** field, type the number of events consumed and processed in a single batch.
4. Click **OK** or **Apply** to accept and save the settings. Click **Cancel** to ignore the changes, or **Reset** to retrieve the last saved values.

Note: For the new values to be effective in the **System Properties** page, you must restart the WebSphere Business Monitor.

Importing a business measures model

Follow these steps to import a business measures model into WebSphere Business Monitor.

Important: On AIX platform, import of the business measures model will fail if the **CPUGaurd** option in the SMIT is enabled, You must make sure that this option is disabled before importing a business measures model.

1. To access the **Model Import** page from WebSphere Application Server administrative console, go to **WebSphere Business Monitor** → **Server** → **Business Measures Model** → **Model Import**.
2. To select the compressed file (.zip) that contains the business measures model XMI file you want to import, click **Browse**. The file path and name are displayed in the **File Name** field.

Important: The selected .zip file should be the one which has been generated by the Schema Generator not the one that was generated by the Business Measures editor.

3. To import the selected file, click **Import**.
4. If you are importing a new version of an existing business measures model, you must restart the WebSphere Application Server.

Unloading a business measures model

Follow these steps to unload (delete) a business measures model that has been previously imported to the WebSphere Business Monitor.

1. To access the **Business Measures Models cleanup** page from the WebSphere Application Server administrative console, go to **WebSphere Business Monitor** → **Server** → **Business Measures Model** → **Model Unload**.
2. To select the business measures model that you want to unload, select the check box next to the business measures model name in the **Business Measures Models** list. You can select only one model at a time.
3. To delete the selected business measures model, click either **Delete** or **Delete and keep for reporting**. If you click **Delete and keep for reporting**, the information about the selected business measures model is kept in the Repository database for dashboard reporting. The information is deleted only from the execution engine.
4. Click **OK** to confirm the deletion, or click **Cancel** to cancel the deletion. The unload action will remove all versions of the selected business measures model.
5. If you intend to import again the business measures model that has just been unloaded then you must restart the WebSphere Application Server.

Administering Adaptive Action Manager

Adaptive Action Manager is a component in WebSphere Business Monitor. It receives situation events emitted by applications, selects appropriate actions based on rules and policies defined by users, and invokes one or more actions.

An administrator for WebSphere Business Monitor binds situation events to action services by associating an action template with situation events, through the Adaptive Action Manager administration console. The Adaptive Action Manager receives situation events, parses the received events, and selects an appropriate action by looking up the action in the action catalog, where information about bindings are stored. Finally, the Adaptive Action Manager invokes the selected action.

Important:

- You must stop the Adaptive Action Manager before you make any changes.
- The changes that are made while the Adaptive Action Manager is running are not effective until you stop and then restart the Adaptive Action Manager services.
- If you stopped the Runtime database, the Adaptive Action Manager application must be stopped from the WebSphere Application Server administrative console first. Otherwise, incoming Alert situation events may be lost by Adaptive Action Manager.

Adaptive Action Manager overview

The Adaptive Action Manager is a key component of WebSphere Business Monitor. It receives situation events emitted by the WebSphere Business Monitor; and potentially, by other applications; selects appropriate actions, based on predefined rules and policies by the business user; and invokes a selected action or a set of actions.

Assume there is a process in your business where you want to detect a particular business situation when it happens. A business situation is a condition where one or more actions needs to be taken. For this condition, a business situation needs to be defined to be triggered when the condition occurs. For example, a situation is defined to occur when a metric exceeds its acceptable threshold. You further define one or more events to be emitted when this situation is triggered. These are defined in the Business Measures editor in WebSphere Business Modeler.

After the business measures models is imported to WebSphere Business Monitor, each time the situation occurs (a metric exceeds its threshold limit or a situation occurs on time), an event is emitted. To be aware of this business situation, you need to be notified when it happens. The emission of an event alone does not notify you. The Adaptive Action Manager dispatches a notification of the business situation and takes one or more actions to address the situation.

To ensure the notifications and the actions take place, you have to perform the following tasks:

- Identify the situation event.

- Define templates for the action services that you want to invoke. The templates are used by an action handler like an alert handler or mail handler to invoke the action services that send a dashboard alert or a mail event.
- Bind the situation events with the templates. The binding defines the configuration used at runtime so that Adaptive Action Manager can determine what action services should be invoked upon receiving the situation event.

When the Adaptive Action Manager receives those situation events, it applies the definitions and the bindings that you have set for those events, invokes the action services identified by the templates bound to the situation event, and notifies you with the situation in the manner you specified.

The Adaptive Action Manager:

- Accepts situation events and invokes one or more action services based on the configuration set up against the value of the business situation name. This business situation name is defined in the Business Measures editor.
- Administers the configuration of templates and event-to-action service bindings.
- Supports the following action service types:
 - Notification
 - Invocation of a Web Service
 - Invocation of a BPEL process that has been exposed as a Web Service
- Emits Common Base Event-formatted events. before and after action invocation.

Adaptive Action Manager components

The Adaptive Action Manager components interact with each other and with other WebSphere Business Monitor components to respond to business situations.

The Adaptive Action Manager consists of the following components:

- **Adaptive Action Manager handlers:** Are a set of action handlers that are used to launch the appropriate action service.
- **Adaptive Action Manager logging service:** Provides logging and tracing information about action invocations.

The Adaptive Action Manager components perform the following functions by interacting with other components in WebSphere Business Monitor:

- The WebSphere Business Monitor administrator binds situation events to action services by defining the action handler and the action template and associating them with situation events.
- The Adaptive Action Manager receives situation events from the Common Event Infrastructure (CEI), which are emitted to CEI by the Monitor Server component.
- The Adaptive Action Manager parses the received situation events and determines an appropriate action by looking up the action in the Action Catalog database, where the information about bindings are stored.
- The Adaptive Action Manager invokes each action handler with a situation event and an action template to invoke the required action service.

Adaptive Action Manager handlers

The Adaptive Action Manager includes action handlers that invoke services when it receives a situation event. Situation events are defined in WebSphere Business Modeler. When the events are triggered, they are sent by the Observation Manager to the Common Event Infrastructure (CEI) where the Adaptive Action Manager

processes the events. Each action handler is then called to initiate the associated action services within the Adaptive Action Manager.

The Adaptive Action Manager obtains configuration information for each action handler from the action catalog service. Action handlers are responsible for executing action services. The following types of action handlers are supported:

- **Notification handlers:** Send alert, e-mail, pager, or cell-phone notifications based on a template. The e-mail data is stored in the Common Base Event.
 - **Alert Handler:** Sends alert notifications based on a given template. They are represented as records in the WebSphere Business Monitor Runtime database. The alert data needed to submit the records is stored in the situation events. The situation events are sent by the Observation Manager to CEI. The stored records are picked up by the alert dashboard to be viewed by the dashboard client in the Alerts view.
 - **E-mail handler:** Sends e-mail, pager, and cell-phone notifications based on a template.
- **Web services handler:** Invokes Web services as an action. The web service handler uses a user defined template that contains the Web service parameters necessary for a Web services invocation.

An executable action is an internal entity specific to the Adaptive Action Manager. It is created by a combination of a Common Base Event, an action handler, and a template. For example, an e-mail handler consumes a Common Base Event and a template to send an e-mail. The template identifier contains e-mail-related information, such as subject, addressees, body, and variables. The e-mail handler substitutes those variables with values from the Common Base Event data.

Notification handlers

Notification handlers carry out the steps to send notifications of different types to users. The notifications are sent based on the action service templates that are configured through the administrative console panels of the Adaptive Action Manager.

There are four types of notifications: alert, e-mail, cell phone, and pager. Each of these action services requires the same type of template data to be stored in the action service catalog database using the Adaptive Action Manager administration panels. Notification handlers are divided into two types:

- The alert handler, which handles alert notifications
- The e-mail handler, which handles e-mail, pager, and cell-phone notifications

The required data in the notification template is as follows:

- LDAP root and LDAP query data that allows for queries to an LDAP database for a list of users who will receive the notification. For alerts, the information obtained is a user ID. For cell phone, pager, and e-mail notifications, it is the e-mail addresses of the users who will receive the notification.
- Subject of the notification that will be sent to the users.
- Body of the notification that contains detailed information on the emitted situation event and the metric values. The metric values are displayed by substituting the defined metric variables with the corresponding values of the incoming situation events. The metric variables are defined within the body and the subject of the notification.

There may be variables defined within the subject or body of a notification. They are mapped to field values within the Common Base Event or the business

situation event, which are read by the handler. The data values within the Common Base Event are substituted into the corresponding variables. E-mail, cell phone, and pager notifications are sent as e-mails to an SMTP server that the user has defined. The Adaptive Action Manager alert handler sends to the Runtime database the subject, body, events, and list of user IDs to be stored. The emitted alert notifications can be displayed through the Alerts view in the dashboards component of WebSphere Business Monitor.

Web service handler

The Web service handler invokes Web services as an action.

The Web service is initiated through fields defined in the Web service template. These fields relate to a typical Web Services Description Language (WSDL) file used by Web services. A Web service template contains the parameters for a Web service invocation. Only Web services that take a Common Base Event as an input message can be supported. Business Process Execution Language (BPEL) processes are supported through the Web service handler. The BPEL process must be exposed through a WSDL file definition.

The Web service can be invoked asynchronously or synchronously depending upon whether the user needs a response from the web service.

Action logging service

The Adaptive Action Manager enables action logging using Common Event Infrastructure (CEI) as the action-logging mechanism.

The Adaptive Action Manager emits events in Common Base Event structure on the Common Event Infrastructure before and after an action is executed. You can track action execution based on the logging data in the emitted events.

The following tables describe the logging events before the actions are executed.

Before Action Handler Logging Event

Common Base Events Property Name	Usage description
globalInstanceId	xsd: ID Globally unique primary identifier of the event, generated by CEI when event is emitted.
creationTime	xsd:dateTime The date and time when the event was issued. Must be a dateTime data type.
sourceComponentId	cbe: ComponentIdentification location = Fully qualified hostname like host.raleigh.ibm.com locationType="FQHostname" application=put nothing in this field component= "IBMWBIMonitor#6.0" subComponent="AdaptiveActionManager#6.0" componentIdType="ServiceName" componentType= "http://www.ibm.com/namespaces/autonomic/ WebSphereApplicationServer/IBMWBIMonitorComponent"
Situation	cbe: Situation situationType cbe: SituationType reasoningScope="EXTERNAL" is assigned. categoryName="OtherSituation"

Before Action Handler Logging Event

Common Base Events Property Name	Usage description
contextDataElement	AAM puts a situation event global identifier here. type = "SituationEventGlobalIdentifier" name="IBMWBIMonitor" (comply with standards) contextId=value of the situation event global identifier use addContextDataElementWithId(String arg0, String arg1, String arg2)method
extensionName	"IBMWBIMonitorActionHandlerLoggingEvent" is stored.
ExtendedDataElements	Data from the extended data element in a Common Base Event (situation event data).

Attributes in ExtendedDataElements

Name	Type	description
IBM_AAM_Logging_Type	xsd: string	"BeforeInvokingActionService"
IBM_AAM_Situation_Extension_Name	xsd:string	Situation Event Extension Name
IBM_AAM_Monitor_Context_Data_Element_Name	xsd:string	Situation Event ContextDataElement Name
IBM_AAM_Monitor_Context_Data_Element_Id	xsd:string	Situation Event ContextDataElement Id
IBM_AAM_Action_Handler_Name	xsd:string	Name of the handler that will be executing

The following tables describe the logging events after the actions are executed.

After Action Handler Logging Event

Common Base Events Property Name	Usage description
globalInstanceId	xsd:ID. Globally unique primary identifier of the event, generated by CEI when event is emitted.
creationTime	xsd:dateTime The date and time when the event was issued. Must be a dateTime data type.
sourceComponentId	cbe: ComponentIdentification location = Fully qualified hostname like host.raleigh.ibm.com locationType="FQHostname" application=put nothing in this field component= "IBMWBIMonitor#6.0" subComponent="AdaptiveActionManager#6.0" componentIdType="ServiceName" componentType= "http://www.ibm.com/namespaces/autonomic/ WebSphereApplicationServer/IBMWBIMonitorComponent"
Situation	cbe: Situation situationType cbe: SituationType reasoningScope="EXTERNAL" is assigned. categoryName="OtherSituation"

After Action Handler Logging Event

Common Base Events Property Name	Usage description
contextDataElement	AAM puts before event global identifier here. type = "BeforeInvokingActionServiceLoggingEventGlobalIdentifier" name="IBMWBIMonitorActionHandler" contextId=value of the before logging event global identifier AAM puts a situation event global identifier here. type = "SituationEventGlobalIdentifier" name="IBMWBIMonitor" contextId=value of the situation event global identifier use addContextDataElementWithId(String arg0, String arg1,String arg2) method
extensionName	"IBMWBIMonitorActionHandlerLoggingEvent" is stored.
ExtendedDataElements	Data from the extended data element in a Common Base Event (situation event data).

Attributes in ExtendedDataElements

Name	Type	Description
IBM_AAM_Logging_Type	xsd: string	"AfterInvokingActionService"
IBM_AAM_Situation_Extension_Name	xsd:string	Situation Event Extension Name
IBM_AAM_Monitor_Context_Data_Element_Name	xsd:string	Situation Event ContextDataElement Name
IBM_AAM_Monitor_Context_Data_Element_Id	xsd:string	Situation Event ContextDataElement Id
IBM_AAM_Action_Handler_Name	xsd:string	Name of the handler that will be executing
IBM_AAM_Action_Invocation_Disposition	xsd:string	"ActionInvocationSuccessful" or "ActionInvocationUnSuccessful"
IBM_AAM_Action_Service_Name	xsd:string	"Email" or "Page" or "SMS" or "Alert" or "Web Service"
IBM_AAM_Template_Name	xsd:string	Name of the template used for the action service
IBM_AAM_Exception_Name	xsd:string	Name of the exception thrown (if there is a one)
IBM_AAM_Application_Message	xsd:string	Name of the application message to the common logs (if there is a one)

Action services templates

Action services templates define information to be used by the Adaptive Action Manager in invoking a particular action service.

Action services templates are defined by the Adaptive Action Manager administration panels from the administration node on the WebSphere Application Server administrative console.

Notification templates, such as the alert notification template, typically contain the subject, body, addressee list, a user-defined LDAP query that determines the user ID or e-mail addresses of those who can display the alert or to whom the notifications are sent, and variables for the notification. Other templates, such as Web services templates, contain the information necessary to initiate their services.

The notification templates define the alert, e-mail, cell-phone, and pager contents. Variables can be inserted into the LDAP query, the subject, and the body. They are delimited by `%VariableName%`, which is embedded inside their entry field. The *VariableName* is substituted from the event, into its respective field similar to a variable in a batch file. The `#` character can be used as an escape character to allow `%` characters to be evaluated as literals. For example, a subject field with value "We are at %UsagePercent%#" will undergo substitution and could be "We are at 97%" when the notification is sent.

The Web services template contains the web service parameters that are required for invoking a Web service.

To define a template, you need to define the configuration parameters for each template in the template definitions pages in the administrative console. You can also modify those parameters later or delete the entire template.

Templates are further bound to situation events so that, when the event is received, the appropriate action services are invoked

Situation events binding

An action service template by default is bound to a particular action handler when the template is created. Binding a situation event to an action service template that is associated with a particular action handler allows the Adaptive Action Manager to invoke the appropriate action service.

When a situation event is detected, the Adaptive Action Manager invokes one or more action service templates according to the binding configuration. A situation event is a Common Base Event that represents a notification of a business situation. Situation events are business measures model elements that are defined in the Business Measures editor as outbound events. When the situation is triggered, the Observation Manager emits the situation event. For example, a situation event can be defined to trigger a situation when a certain Key Performance Indicator (KPI) exceeds a threshold. The situation event is then published into Common Event Infrastructure (CEI) and is routed to a WebSphere activation specification that is linked to a queue that the Adaptive Action Manager is listening to.

Situation events are defined into Adaptive Action Manager by a name and description. They are then bound to an action services template and thus a corresponding action handler through the Administration node on the administrative console of the WebSphere Application Server. This binding at runtime allows the Adaptive Action Manager components to use the action handler and template pairing to invoke the appropriate action service.

When the binding is set, the Adaptive Action Manager can parse the received situation event and select an appropriate action through the action catalog service. The selection is done by looking up the appropriate action in the action catalog, where information about bindings is stored. Based on this information, the Adaptive Action Manager creates entities called executable actions. An event-to-action determiner functionality determines what executable actions should be invoked for an incoming situation event, by matching it with the situation name stored in an extended data element in the Common Base Event.

Action Catalog data export and import

The Adaptive Action Manager provides a command-line tool you can use for exporting the defined actions and templates data stored in the action catalog database to an XML file. This file can be imported again into another action catalog database. You do not have to redefine the actions and templates, for instance, when you are moving from a testing environment to the production environment.

The export and import is performed using a command-line tool, `Importer.java`, that is initiated by running a batch file, `runAAMExport.bat`. This tool uses a properties file, `config.properties`, that contains required properties and the values needed for the export and import operations. By editing these properties, you can determine whether the tool will export the data in the action catalog database to the XML file or import the XML file to the action catalog database. You can change the properties that indicate the source and destination of the tool. You can export data from an existing action catalog to an XML file with one run of the tool. Later, you can change the configuration parameters in the properties file and run the tool again to import the data from the XML file to a new action catalog database. If this new action catalog database contains any actions data, it will be replaced with the imported data for the identical records (identical templates, identical situation-event binding names).

You can override the configuration parameters in the `config.properties` file by using the `-D` command in the command line from which you are running the `runAAMExport.bat`, and then include the parameters with different values in this command. You can use this command to invoke the tool with new values for the parameters without having to change these values in the `config.properties` file.

The action catalog data export and import tool provides logging and tracing. The logging and tracing feature is configured using the `logging.properties` file that is located in the same directory that contains the `runAAMExporter.bat` file. The `logging.properties` file conforms to the JSR47 specification format for controlling logging. By default, this file is set to log to the command prompt window from which you run the tool. You can reconfigure this file to use different handlers, log levels, and formats. Refer to the JSR47 documentation before editing the `logging.properties` file for details about its contents and how to change the values of its properties.

The following table contains the available configuration parameters in the `config.properties` file, with their possible valid values.

Note: If you specified the XML file as the target of the tool (in the `ActionManager.to.type` property), then you only need to set the value for the `ActionManager.to.XmlFilename` property. All the target database-related properties are not needed and will be bypassed. If you have specified the XML file as the source of the tool (specified in the `ActionManager.from.type` property), you only need to set the value for the `ActionManager.from.XmlFilename` property. All source database-related properties are not needed and will be bypassed. You only need to fill in database information when using a database and to fill in XML information when using XML.

Action Catalog data export and import tool configuration properties

Property name	Description	Valid values
ActionManager.from.moveGlobalConfig	Determines whether to export the static configuration, which is in the action catalog database.	true, false
ActionManager.to.type	Determines the output type of the tool (action catalog database or XML file).	CM_DB_LAYER, XML
ActionManager.to.serverName	The server name or IP address of the server machine where the target action catalog database resides.	For example, localhost
ActionManager.to.portNumber	The port number of the action catalog database to which you are moving the data.	For example, the DB2 default port number is 5000.
ActionManager.to.driverType	The DB2 driver type for the action catalog database to which you are moving the data.	2 or 4. (The default value is 4.)
ActionManager.to.dbName	The name of the action catalog database to which you are moving the data.	The default database name is AAMCAT.
ActionManager.to.schemaName	The Schema of the action catalog database to which you are moving the data.	The default Schema name is AAMCAT.
ActionManager.to.username	The user name of a user with enough authorizations on the action catalog database to which you are moving the data.	
ActionManager.to.password	The password of the given user name on the action catalog database to which you are moving the data.	
ActionManager.to.XmlFilename	The path and name of the XML file to which you are moving the data. (On the Windows® platform, use a double backslash in the file path.)	For example, C:\AAMCAT.XML
ActionManager.to.overwriteXML	Determines whether you want to overwrite the XML file, if it exists.	true, false
ActionManager.from.type	Determines the input source type of the tool (action catalog database or XML file).	CM_DB_LAYER, XML

Action Catalog data export and import tool configuration properties

Property name	Description	Valid values
ActionManager.from.serverName	The server name or IP address of the server machine where the source action catalog database resides.	For example, localhost
ActionManager.from.portNumber	The port number of the action catalog database from which you are moving the data.	For example, the DB2 default port number is 5000.
ActionManager.from.driverType	The DB2 driver type for the action catalog database from which you are moving the data.	2 or 4. (The default value is 4.)
ActionManager.from.dbName	The name of the action catalog database from which you are moving the data.	The default database name is AAMCAT.
ActionManager.from.schemaName	The Schema of the action catalog database from which you are moving the data.	The default Schema name is AAMCAT.
ActionManager.from.username	The user name of a user with enough authorizations on the action catalog database from which you are moving the data.	
ActionManager.from.password	The password of the user name on the action catalog database from which you are moving the data.	
ActionManager.from.XmlFilename	The path and name of the XML file from which you are moving the data. (On the Windows platform, use a double backslash in the file path.)	For example, C:\AAMCAT.XML

Configuring Adaptive Action Manager properties

To configure the Adaptive Action Manager, you configure the general properties and the LDAP properties.

Configuring general properties

To configure the general properties of the Adaptive Action Manager component, you perform the following steps:

All the fields below are optional

1. To access the **General** tab from the WebSphere Application Server administrative console, go to **WebSphere Business Monitor > Adaptive Action Manager > Configuration**.

2. Select the **General** tab.
3. In the **SMTP Sender Name** field, type the default name of the sender or the e-mail address of the sender of notifications.
4. In the **SMTP Host Name** field, type the SMTP server host name for the e-mail server.
5. In the **SMTP Port Number** field, type the listener port number used by the SMTP server.
6. In the **CEI Action Logging** options, select to turn CEI action logging **On** or **Off**. The default is **Off**.
7. Click **Apply** or **OK** to save the settings you entered.

Note: You must save the current values before you can change the panes or panels. Otherwise, changes will not be saved.

Configuring LDAP properties

To configure LDAP properties for the Adaptive Action Manager, you perform the following steps:

All the fields below are optional

1. To access the **LDAP** tab from WebSphere Application Server administrative console, go to **WebSphere Business Monitor > Adaptive Action Manager > Configuration > LDAP**.
2. In the **LDAP URL** field, type the LDAP URL that is used to connect to the user registry.
3. In the **LDAP User ID** field, type the LDAP user ID that is used to log in the to the user registry.
4. In the **LDAP Password** field, type the LDAP user password that is used to log in to the user registry.
5. In the **LDAP Alert** field, type the LDAP attribute for alert information.
6. In the **LDAP Cell Phone** field, type the LDAP attribute for cell phone information.
7. In the **LDAP Email** field, type the LDAP attribute for the e-mail address.
8. In the **LDAP Pager** field, type the LDAP attribute for pager information.
9. Click **Apply** or **OK** to save the settings you entered.

Note: You must save the current values before you can change the panes or panels. Otherwise, changes will not be saved.

Registering an action service

The system administrator uses the Adaptive Action Manager administrative console to register each action service. An action service notifies defined users about a situation and/or invokes a web service application.

The registration of the action service involves the creation of a template for this action service. Each template is created specifically for one type of action service only. Afterwards, the system administrator binds a situation event to an action service.

Creating a new notification template

To create a new notification template definition, you perform the following steps.

All steps are optional unless noted as mandatory.

1. To access the **Notification Template Configuration** tab from WebSphere Application Server administrative console, go to **WebSphere Business Monitor > Adaptive Action Manager > Template Definitions > Notification**.
2. Click **New**.
3. In the **Template name** field, type a unique template name. This attribute is mandatory.
4. In the **Description** field, type a template description.
5. From the **Action service Type** options, select the required action service: **Alert, Cell Phone, Email, or Pager**.
6. In the **To (LDAP Query)** field, type the LDAP formatted query to retrieve the set of users intended to receive this notification, using variables.
7. In the **Subject** field, type the notification subject, using variables. These variables map to Common Base Event data fields and are identified with surrounding percent signs (%CBEVARIABLE%). The escape character to display a percent sign is #. (%% is evaluated as %, and ## is evaluated as #.)
8. In the **Body** field, type the notification body, using variables.
9. In the **LDAP Root** field, type the distinguished name for the root of the LDAP search query.
10. Click **Apply** or **OK** to save the entered settings.

Note: You must save the current values by clicking **Apply** or **OK** before you can change the panes or panels. Otherwise, changes will not be saved.

Creating a new web services template

To create a new web-services-template definition, you perform the following steps.

All steps are optional unless noted as mandatory.

Creating a new web services template can only invoke an operation that takes a single string variable as input. The string variable will be filled in by an XML string representation of the incoming event.

1. To access the **WebServices Template Configuration** tab from WebSphere Application Server administrative console, go to **WebSphere Business Monitor > Adaptive Action Manager > Template Definitions > Web Services**.
2. Click **New**.
3. In the **Template Name** field, type a unique template name. This attribute is mandatory.
4. In the **Description** field, type the template description.
5. In the **Target Namespace** field, type the target name space attribute of the definitions element.
6. In the **Service Name** field, type the service name attribute of the service element.
7. In the **End Point address** field, type the location attribute of the service, port, or address element.
8. In the **Port Type** field, Type the name attribute of the port type element.

9. In the **Operation Name** field, type the operation name attribute of the port type or operation element.
10. In the **Input Message Name** field, type the name attribute of the port type, operation, or input element.
11. Click **Apply** or **OK** to save the entered settings.

Note: You must save the current values by clicking **Apply** or **OK** before you can change the panes or panels. Otherwise, the changes will not be saved.

Updating action services template definitions

Updating the action services template definitions consists of updating defined notification templates and web service templates.

Updating a notification template

To update a notification template, you can modify the template attributes, create a new template definition, or delete a template definition.

1. To access the **Notification Template List** page from WebSphere Application Server administrative console, go to **WebSphere Business Monitor > Adaptive Action Manager > Template Definitions > Notification** page. The page contains a list of already created notifications templates.
2. To modify the template attributes, in the **Template Name** column, click the notification template name. The **Notification Template Configuration** page opens displaying the template attributes that you can change.
3. To create a new notification template definition, click **New**. An empty **Notification Template Configuration** page opens where you can create a new notification template configuration.

To delete a notification template definition, select the template by selecting the check box next to the template name, and then click **Delete**.

Note: All checked templates are deleted when you click **Delete**.

Updating a Web services template

To update a Web services template, you can modify the template attributes, create a new Web template definition, or delete a template definition.

1. To access the **Installed Web Services Templates** page from WebSphere Application Server administrative console, go to **WebSphere Business Monitor > Adaptive Action Manager > Template Definitions > Web Services** page. The page contains a list of already created Web services templates.
2. To modify template attributes, in the **Template Name** column, click the Web services template name. The **WebServices Template Configuration** page opens displaying the template attributes that you can change.
3. To create a new Web service template definition, click **New**. An empty **WebServices Template Configuration** page opens where you can create a new Web service template configuration.

To delete a Web service template definition, select the check box next to the template name, and then click **Delete**.

Note: All checked template definitions are deleted when you click **Delete**.

Binding a situation event with action services

After creating the action templates, the user binds the templates to a situation event. Each template is associated with only one action service.

The action service is bound to the situation event by the value found in the `BusinessSituationName` field, within the situation event being passed to the Adaptive Action Manager from the server. This value is used by Adaptive Action Manager at runtime so that it can determine what action services should be invoked upon receipt of the situation event. The situation event can be bound to more than one action service. For example, the situation event could be bound to both an e-mail notification and a web-service invocation.

Creating a new situation event binding

To create a binding among the template definition, action service type, and the required situation events, you perform the following steps:

1. To access the **New Situation Event Binding** tab from WebSphere Application Server administrative console, go to **WebSphere Business Monitor > Adaptive Action Manager > Installed Situation Event Binding**.
2. Click **New**.
3. In the **Situation Event Name** field, type a unique situation-event name. This attribute is mandatory. The situation event name must be the same as what is specified for the attribute of *BusinessSituationName* of corresponding situation events in Business Measures editor. If they were different, the notification event will not be sent. The maximum length for this field is 64 bytes.
4. In the **Description** field, type a description of the situation event.
5. To add a new table row representing a binding to a specific template and specific action service, click **Add**. The **New Situation Event Binding** page opens.
6. To associate an action service template to a situation event, see “Adding a template to situation event binding.”
7. Click **Apply** or **OK** to save the entered settings.

To delete the bind between the situation event and the associated templates, select one or more templates, and then click **Remove**. The remove operation does not delete the actual template, it only removes the binding between the action service template and the situation event.

Adding a template to situation event binding

To add template definitions, which are associated with an action service type, to situation event binding, perform the following steps:

1. To access the **Add Template to Situation Event Binding** tab from WebSphere Application Server administrative console, go to **WebSphere Business Monitor > Adaptive Action Manager > Installed Situation Event Binding** page.
2. In the **Situation Event Name** column, click the name of situation-event-binding template. The **New Situation Event Binding** page opens, displaying the binding attributes that you modify.
3. Click **Add**.
4. From the **Template Name** list, select the template name. The list box contains all sorted existing template definitions. This attribute is mandatory.

Note: If no templates are already defined, there are none to select. Templates must have been previously defined.

5. To save and apply the settings, click **Apply**. A new row is created for the new template in the **Installed Situation Event Binding** table. You are returned to the **Installed Situation Event Binding** page.

Updating a situation event binding

To update the installed situation event binding templates, you perform the following steps:

1. To access the **Installed Situation Event Binding** page from WebSphere Application Server administrative console, go to **WebSphere Business Monitor > Adaptive Action Manager > Installed Situation Event Binding** page. The page contains a list of bindings.
2. In the **Situation Event Name** column, click the name of the situation-event-binding template. The **New Situation Event Binding** page opens, displaying the binding attributes you can modify.
3. To create a situation event-binding-template definition, click **New**. An empty **New Situation Event Binding** page opens where you can create a new situation event binding.
4. To delete the binding template definitions, select one or more check boxes next to the situation event binding names, and then click **Remove**.

Note: The template definitions are not deleted when clicking **Remove**. Only the bindings are deleted.

Using the Action Catalog database export and import tool

You use the Action Catalog data export and import tool to export the defined actions and templates data, which are stored in the Action Catalog database, to an XML file. You also use the tool to import this XML file into another Action Catalog database.

1. Open the batch file named `runAAMExport.bat` in edit mode.
2. Make sure that the `%JAVA_HOME%` variable is set and points to the correct Java™ home directory. You can use the IBM JDK that is shipped with IBM WebSphere Application Server.
3. Save and close the file.
4. Open the file named `config.properties` in edit mode.
5. Modify the values for each property in this file, to set the source and target of the tool.
6. Save and close the file.
7. Make sure that `db2jcc.jar` file exists in the classpath environment variable. This file is installed with DB2. If this file does not exist in the classpath environment variable, then add it with its location.
8. Run the `runAAMExport.bat` file as follows:
 - a. Start a command-line window.
 - b. Call the batch file by typing `runAAMExport.bat`, and press Enter.
 - c. If you want to specify a different location for the `config.properties` file, add the following: `"-DActionManager.mover.configFile=<The config.properties new file name and path>"` to the command line after the batch file name. For example: `runAAMExport.bat -DActionManagerMover.configFile=C:\temp\example.properties`

- d. You can also add any property in this command line with its value to override the value that exists in the config.properties file. Type: -D
<property_name>=<property_value>

Note: After data is imported into the Action Catalog database, if the user is logging in to the administrative console, the user must log out and log in again to see the imported data.

Managing schema generation

Managing schema generation consists of configuring and generating the database schemas.

Overview

The Schema Generator is a key component in WebSphere Business Monitor. It is part of the WebSphere Business Monitor administrative console. It uses the business measures model that is produced by the Business Measures editor to generate the appropriate artifacts for this model, based on a particular configuration.

The Schema Generator generate the needed artifacts to create the dynamic database tables schemas. Dynamic databases tables are based on business measures models, which are imported into the administrative console; their schemas are unique for each model. The dynamic databases tables must be updated and regenerated if any change occurs in the business measures model.

The dynamic databases tables are created in the State, Runtime, and Historical databases. The Schema Generator produces the following artifacts for each business measures model:

- Database DDL scripts: Are the required definitions for creating database tables and indexes.
- DB2 Cube Views metadata: Is the required star schema for generating the multidimensional views on the DB2 OLAP center.
- Database replication scripts: Are the required scripts for synchronizing databases by enabling replication for the following databases:
 - State
 - Runtime
 - Historical

The replication scripts generated by the Schema Generator make use of DB2 Replication utility.

Database DDL artifacts

You use the Schema Generator to generate the database artifacts, scripts, and associated DDLs, to support a business measures model.

The Schema Generator, using the annotated business measures model, generates DB2 specific DDL files. These files create the actual tables and columns in the various databases. The database administrator (DBA) must manually run the DDL files to create the database tables and their columns in the State, Runtime, and Historical databases.

The generated DDL files are saved into a directory location that is specified in the General Configuration page under the Schema Generator node in the WebSphere Business Monitor administrative console. One DDL file and one description file, in text format, are created for each database. The description file contains the description of the DDL file contents.

Each DDL file contains the SQL statements necessary to:

- Create or update the database tables
- Set any appropriate configuration parameters for a given table
- Create the required indexes

A ZIP file is also generated. It contains the following two files:

- The business measures model XML metadata interchange (XMI) file.
- Cube Views metadata XML file.

This ZIP file is the one that will be imported in the WebSphere Business Monitor using the Import Model page of the Business Measures Model administrative functions, under the Server node on the administrative console. For more information about importing the Business Measures Model, refer to “Importing a business measures model” on page 13.

DB2 Cube Views metadata

A key benefit of using the Schema Generator is that the database administrator (DBA) does not have to manually create the Cube Views schema using the online analytical processing (OLAP) center. It automatically generates a Cube Views XML file, which contains cube models and cubes to support the business measures model.

The Schema Generator uses the business measures model to produce a DB2 Cube Views XML file. This XML file contains information that describes the star schema associated with this business measures model. After the generation of the schema, the DBA must deploy the generated DDL scripts, which create additional database tables in the Historical database. Next, the DBA uses the OLAP center to manually import the Cube Views schema.

Important:

- All databases and related static tables are created before the Cube Views schema is deployed.
- The DDL that is generated to represent the running database tables must be executed before the Cube Views metadata is executed.

The Schema Generator maps the business measures model to a simple star schema representation, which will be deployed in the OLAP center. Some key points to note about how the Cube Views schema is generated are:

- A cube model is created for each business measures group. The name of the cube model is derived from the business measures model.
- A fact table exists for each business measures group. The fact table name is defined in the business measures model.
- For each metric, there may be one or more aggregation types (such as SUM, AVG, COUNT, MAX, or MIN) and a corresponding measure name.
- The schema name is specified at the time the product is installed. The Launchpad collects the schema name during the installation process.
- The cube model and cubes are based on the most current business measures model. Any change in the business measures model requires regenerating and redeploying the Cube Views metadata.

Database replication scripts

The Schema Generator generates replication scripts that are used to handle the replication from the State to the Runtime database and from the Runtime to the Historical database.

The purpose of the replication services is to provide an infrastructure that copies business measures model events data stored in the State database to the Runtime database and from the Runtime database to the Historical database. This infrastructure is dynamically built during the deployment phase of the business measures model. The Schema Generator creates the replication scripts. These scripts, once deployed on the systems hosting the State, Runtime, and Historical Databases, provide the services necessary to move the data from one database to another. The general activity flow can be described as follows:

- Create scripts that implement data movement for a business measures model using the Schema Generator
- Deploy replications scripts on the machine in which the State, Runtime, and Historical Databases are hosted.
- Enable replication services

To enable the Schema Generator to create the required replication scripts, the DBA must configure the replication operation through the State to Runtime and Runtime to Historical pages in the Schema Generation Configuration panel before generating the scripts. The defined configuration parameters can be changed later. For more information about changing data movement service configuration parameters, refer to “Configuring the Schema Generator” on page 34. The most important parameter is the replication population interval, which determine the time period in minutes between two replication cycles. This interval is determined according to the required frequency of updating the information in runtime. This time period must be defined for the replication between State to Runtime databases and between Runtime to Historical databases. The default replication interval for the State to Runtime database replication is 10 minutes, while the default replication interval for the Runtime to Historical database replication is 24 hours.

Subsequent changes to the business measures model have an impact on the generated replication scripts and can lead to a re-creation and redeployment of previously generated scripts. The Schema Generator maintains the mapping among State, Runtime, and Historical database tables; it generates the required replication scripts. A new mapping is created, reflecting the latest state of the databases. A user will have to go through a series of activities before starting replication services. The basic activity flow is:

1. Configure replication parameters through WebSphere Business Monitor administrative console
2. Generate replication scripts using information derived from the Schema Generator and other sources
3. *Optional.* Customize generated replication scripts
4. Deploy replication scripts.
5. Start replication services

Usage scenario

The Schema Generator usage scenario describes how the Schema Generator is used to deploy a business measures model.

Before using the Schema Generator component to generate artifacts, you must perform some tasks. The following tasks describe the usual scenario for the creation and installation of WebSphere Business Monitor databases.

1. The business measures model is created, edited, modified, and annotated in the Business Measures editor. The business measures model file is in XML metadata interchange (XMI) format. It is exported from the Business Measures editor in .zip file format.
2. The following databases should be created:
 - The Repository database is created either during the installation of WebSphere Business Monitor or manually by the database administrator (DBA). Because the Repository database is static, the DBA creates all table spaces, tables, and indexes at this time. The required scripts for creating the Repository database are saved in *monitor_installation_dir\install\mondb*.
 - The State, Runtime, and Historical databases are created either during the installation of WebSphere Business Monitor or manually by the DBA. Only the databases and some static tables are created at installation. The required scripts for creating the State, Runtime, and Historical databases are saved in *monitor_installation_dir\install\mondb*.

The following steps describe the usual scenario for the deployment of databases artifacts:

1. The business measures model is imported through the Schema Generator panel in the WebSphere Business Monitor administrative console.
2. The Schema Generator generates databases artifacts.
3. The DBA executes the generated Database DDL files on the appropriate databases.
4. The DBA executes the replication setup scripts.
5. The DBA imports the generated Cube Views XML file into the DB2 OLAP Center.
6. The business measures model can be deployed through the WebSphere Business Monitor administrative console. Deployment is done by importing the generated .zip file, which contains the business measures model XMI file and the Cube Views metadata XMI file, into the Import Model page.

The artifacts generated by the Schema Generator are based on the business measures model. When you change the business measures model in the Business Measures editor, you must regenerate and redeploy this model again.

Configuring the Schema Generator

There are some configuration settings that pertain to the Schema Generator. The various configuration settings are divided among three configuration tabs.

General configuration

To set the general configuration of the Schema Generator component, you perform the following steps:

1. To access the **General Configuration** tab from WebSphere Application Server administrative console, go to **WebSphere Business Monitor** → **Schema Generator** → **Configuration**.
2. Select the **General Configuration** tab.

3. In the **Table Space Properties File** field, type the path of the table space properties file. This file contains the definition and configurations of the table spaces that are created and assigned to database tables that will be created by the output DDL scripts.
4. In the **Business Measures Model** field, type the path and name of the .zip file that contains the business measures model XMI file. The output of the Business Measures editor is a .zip file: monitor.zip.
5. In the **Output Directory** field, type the path to the directory where the output artifacts will be saved.

Note: AIX users should ensure that the user doing the deployment has access to the output directory specified here.

6. If you want to generate all artifacts for a business measures model, select **Ignore older deployments and generate all artifacts**. To generate artifacts that only handle the differences between the two models, do not select this check box. Typically, you will want to generate the differences, and not all the artifacts.
7. Click **Apply** or **OK** to save the entered settings.
8. Click **Save** in the messages box to apply changes to the master configuration.

Note: You must save the current values before you can move to other panes or panels. Otherwise, changes will not be saved.

State to Runtime databases replication

To set the replication configuration between the State and Runtime databases, you perform the following steps:

1. To access the **State to Runtime Configuration** tab from WebSphere Application Server administrative console, go to **WebSphere Business Monitor** → **Schema Generator** → **Configuration**.
2. Select the **State to Runtime Configuration** tab.
3. In the **Capture Log Path** field, type the path of the general log file. The Capture log path will be specified as the *CAPTURE_PATH* parameter to each of the Capture component servers that will be servicing the State to Runtime database movement service. This path must exist on the server containing the State database. This path must be accessible to the user that runs the Capture component servers. The choice of this path may affect performance. Refer to DB2 documentation for more information about *CAPTURE_PATH* parameter and performance.
4. In the **Apply Log Path** field, type the path of the apply log file. The Apply log path will be specified as the *APPLY_PATH* parameter to each of the Apply component servers that will be servicing the State to Runtime database movement services. This path must exist on the server containing the Runtime Database. This path must be accessible to the user that runs the Apply component servers. The choice of this path may affect performance. Refer to DB2 documentation for more information about *CAPTURE_PATH* parameter and performance.
5. In the **Runtime Database Population Interval** field, type the required time interval between replication cycles.
6. Click **Apply** or **OK** to save the entered settings.
7. Click **Save** in the messages box to apply changes to the master configuration.

Note: You must save the current values before you can move to other panes or panels. Otherwise, changes will not be saved.

Runtime to Historical databases replication

To set the replication configuration between Runtime and Historical databases, you perform the following steps:

1. To access the **Runtime to Historical Configuration** tab from WebSphere Application Server administrative console, go to **WebSphere Business Monitor** → **Schema Generator** → **Configuration**.
2. Select the **Runtime to Historical Configuration** tab.
3. In the **Capture Log Path** field, type the path of the general log file. The Capture log path will be specified as the *CAPTURE_PATH* parameter to each of the Capture component servers that will be servicing the Runtime to Historical database movement service. This path must exist on the server containing the Runtime database. This path must be accessible to the user that runs the Capture component servers. The choice of this path may affect performance. Refer to DB2 documentation for more information about *CAPTURE_PATH* parameter and performance.
4. In the **Apply Log Path** field, type the path of the apply log file. The Apply log path will be specified as the *APPLY_PATH* parameter to each of the Apply component servers that will be servicing the Runtime to Historical database movement services. This path must exist on the server containing the Historical Database. This path must be accessible to the user that runs the Apply component servers. The choice of this path may affect performance. Refer to DB2 documentation for more information about *CAPTURE_PATH* parameter and performance.
5. In the **Historical Database Population Interval** field, type the required time interval between replication cycles.
6. Click **Apply** or **OK** to save the entered settings.
7. Click **Save** in the messages box to apply changes to the master configuration.

Note: You must save the current values before you can move to other panes or panels. Otherwise, changes will not be saved.

Generating schema

After you have configured the Schema Generator component, you next invoke the Schema Generator to generate the required artifacts of the business measures model.

To generate the databases artifacts, you perform the following steps.

1. To access the **Schema Generate Profile** page from WebSphere Application Server administrative console, go to **WebSphere Business Monitor** → **Schema Generator** → **Generate**.
2. To generate the artifacts according to the configuration you entered, click **Generate**. The generated artifacts are saved in the output directory that is specified in the Schema Generator configuration panel. The contents of the output directory are:
 - Three DDL files for each database (State, Runtime, and Historical)
 - Three text files describing the tables of each database

- Three zip file containing the replication scripts and procedures required to manage replication among databases (one for State, one for Runtime and one for Historical database). On AIX system, JAR archives will be created instead of zip files.

On the AIX system, the Schema Generator files that are created are owned by the user ID under which the WebSphere Business Monitor server is running. Because they are created with general read-and-write access, you should take the following precautions when configuring and running the Schema Generator:

- Ensure that the specified output directory is accessible to the user who will deploy the generated scripts.
- The generated artifact files are created with general access. It is recommended that, immediately after generation, the user should make copies of the files and remove the generated artifacts that are accessible to anyone. This activity should be done before the database artifacts are deployed.

Performing general administration tasks

There are several administrative tasks that you can perform on WebSphere Business Monitor outside the WebSphere Business Monitor Administration extension in the WebSphere Application Server administrative console.

Starting and stopping enterprise applications

Follow these steps for starting and stopping one or more of the installed enterprise applications of WebSphere Business Monitor through the WebSphere Application Server administrative console.

1. From the WebSphere Application Server administrative console, go to **Applications** → **Enterprise Applications**.
2. From the **Enterprise Applications** list, select the check box next to the application name as follows:
 - a. To start or stop the **Monitor Server**, select the **IBM_WB_MONITOR_SERVER** enterprise application.
 - b. To start or stop the **Adaptive Action Manager**, select the **IBM_WB_ACTIONMANAGER** enterprise application.
3. Click **Start** to start the selected applications, or click **Stop** to stop them.

Managing the logging and tracing

Follow these steps to manage the logging and tracing feature in WebSphere Business Monitor using the WebSphere Application Server administrative console. Tasks include enabling the logging and tracing for the application server on which the WebSphere Business Monitor components are deployed and setting the logging and tracing levels for each of the WebSphere Business Monitor components.

Configuring log files

Follow these steps to configure the log files.

1. From the navigation tree on the WebSphere Application Server administrative console, select **Troubleshooting** → **Logs and Trace**.
2. In the **Logging and Tracing** page, from the **Application Servers** table, click the name of the application server whose logging files you want to configure.
3. Click **JVM Logs**.
4. In the **Configuration** tab, specify the name and path for the system out (SystemOut.log) and system error (SystemErr.log) log files, or you can accept the default name and location.
5. Specify the maximum file size for each log file, or you can accept its default maximum size.
6. Click **OK**.
7. In the **Message** information box, click **Save** to apply your changes. The **Save** page appears.
8. Click **Save**.

Enabling the Trace option

Follow these steps to enable the trace option and to configure the trace file.

1. From the navigation tree on the WebSphere Application Server administrative console, select **Troubleshooting** → **Logs and Trace**.
2. In the **Logging and Tracing** page, from the **Application Servers** table, click the name of the application server on which you want to enable the tracing option.
3. Click **Diagnostic Trace**.
4. In the **Configuration** tab:
 - a. Select the **Enable Log** check box to enable the trace option.
 - b. Specify the name and path for the trace out file (trace.log), or you can accept the default name and location.
 - c. Specify the maximum file size for the trace file, or you can accept its default maximum size.
 - d. Specify the maximum number of historical files to a reasonable number. Otherwise, the trace file will be recycled after reaching the provided size limit, and the trace information will be lost.
5. Click **OK**.
6. In the Message information box, click **Save** to apply your changes. The Save page appears.
7. Click **Save**.

Changing log details levels

Follow these steps to set logging levels for WebSphere Business Monitor components.

1. From the navigation tree on the WebSphere Application Server administrative console, select **Troubleshooting** → **Logs and Trace**.
2. In the **Logging and Tracing** page, from the **Application Servers** table, click the name of the application server on which the components whose logging levels you want to set are deployed.
3. Click **Change Log Level Details**.
4. If you want to change the logging levels for the current session only, select the **Runtime** tab. If you want to set the logging levels across all sessions, select the **Configuration** tab.
5. In the selected tab:
 - a. Expand the component *com.ibm.wbimonitor*.
 - b. To specify the logging level for each component, click the component name and select the logging level from the pop-up menu that appears.
 - c. If you selected the **Runtime** tab, you can select the **Save runtime changes to configuration as well** check box to apply the changes in the logging details levels across all sessions.
6. Click **OK**.
7. In the Message information box, click **Save** to apply your changes. The Save page appears.
8. Click **Save**.
9. If you made the changes in the **Configuration** tab, restart the server.

Increasing the databases connections pool size for Dashboard Client

Dashboard Client communicates with the Repository, Runtime and Historical databases using Java Database Connectivity (JDBC), which manages reusable pools of database connections.

There must be sufficient connections in these pools to accommodate multiple simultaneous Dashboard sessions. If there are no sufficient connections in each of these pools, you may experience long, unexpected and unpredictable delays in their Dashboard activities. The recommended initial maximum connection pool size is 100. If long delays are experienced, check that the maximum size of each pool is 100. If delays persist, try increasing the maximum size of each pool. If delays persist after increasing the pool sizes, look for solutions elsewhere

There are five relevant data sources that their corresponding maximum connection sizes should be increased. Each of these data sources belongs to a specific JDBC Provider as follows:

- **JDBC Provider name:** MonitorDB2XADriver1
 - **Data source name:** HistoricalDBDataSource
 - **Data source name:** RepositoryDBDataSource
 - **Data source name:** RuntimeDBDataSource
- **JDBC Provider name:** wpsdbJDBC
 - **Data source name:** wmmDS
 - **Data source name:** wpsdbDS

The data pool maximum sizes are set from the WebSphere Application Server administrative console. To increase the maximum size of the databases connection for each of the above data sources:

1. From the WebSphere Application Server administrative console navigation tree, select **Resources** → **JDBC Providers**.
2. From the **JDBC Providers** list, select **<JDBC_Provider_Name>**.
3. Click the **Data Sources** link.
4. From the **Data Sources** list, select **<Data_Source_Name>**.
5. Click the **Connection pool properties** link.
6. In the Maximum connections field, type the value of the maximum connections you need.
7. Click **OK**.
8. Click **Save** in the **Message** information box to apply your changes. The **Save** page appears.
9. Click **Save**.

Enabling and disabling the CEI store

The emission of events from the runtime engine to the WebSphere Business Monitor occurs as message transmissions using the Common Event Infrastructure (CEI).

There is an option in the CEI configuration to use CEI store or not. If this option is turned on, the CEI persists every message it receives before sending it to the Monitor Server. Enabling this option can serve as a backup in case of failures; lost messages can be recovered and sent again to the Monitor Server. On the other hand, the persistence of messages affects performance, resulting in a lower total throughput of events. You can choose to enable or disable this feature depending on the priority of performance against messages backup and also based on your hardware configuration. To enable this option:

1. From the left pane of WebSphere Application Server administrative console, select **Resources** → **Common Event Infrastructure Provider**.

2. Select the appropriate level. (If you are using default server settings, click **Cell**.)
3. From **Additional properties**, click **Event Server Profile**.
4. Click on the profile name you are using. (The default is **Default Common Event Infrastructure server**.)
5. Select the **Enable Data Store** check box to enable this feature or clear the check box to disable the feature.
6. Click **OK** and save your changes.

Changing the data sources names

The data sources names for WebSphere Business Monitor databases are hardcoded. To change those names, perform the following steps:

1. Stop the application whose data source name you want to change. (For example, stop the Monitor Server if you want to change the data source name that is used for recording the state of the process or process aggregate in the State database.)
2. Create a new data source in the resources of the application server on which the application is deployed as follows:
 - a. From the navigation tree on the WebSphere Application Server administrative console, select **Resources** → **JDBC Providers**.
 - b. Change the scope to the server on which the application is installed.
 - c. Select **MonitorDB2XADriver** → **Data Sources** → **New**. Click **New**.
 - d. In the **New Data Source** properties page, enter the information of the new data source. Click **Save** to save your entries.
 - e. Click the **Test connection** button to test the connection with this new data source.
3. Change the data source that the application points to as follows:
 - a. From the navigation tree on the WebSphere Application Server administrative console, select **Enterprise Applications** → **<Enterprise_Application_Name>**.
 - b. Click **Map resource references to resources**
 - c. Go to **javax.sql.DataSource** section, and select your newly created data source JNDI name from the **Specify existing Resource JNDI** name combo box.
 - d. Select both check boxes at the bottom of the **javax.sql.DataSource** section.
 - e. Click **Save** to save your changes.
 - f. Click **OK**.
 - g. In the Message information box, click **Save** to apply your changes. The Save page appears.
 - h. Click **Save**.
4. Start the enterprise application.

Changing DB2 Alphablox data source user name and password

The user name and password of data sources that are created on the DB2 Alphablox for the Dashboard Client can be changed using the DB2 Alphablox administrative console.

During the installation of the Dashboard Client using the WebSphere Business Monitor Launchpad, two data sources are created in DB2 Alphablox, named **<Historical_DB_Name>** (Default name is HISTORY) and

<Historical_DB_Name>_Cube. You can change the default user name and password for the first data source while the second data source (a relational data source) has no user name and password properties to be changed. To Change the user name and password of the <Historical_DB_Name> data source using the DB2 Alphablox administrative console:

1. In the DB2 Alphablox administrative console, select the **Administration** tab.
2. Click **Data Sources**.
3. From the **Data sources** list, select the data source named <Historical_DB_Name> and then click **Edit**. The default data source name is HISTORY unless you have used a different name for the Historical database, then the data source will have the same different name.
4. In the data sources properties, change the value of the **Default Username** and **Default Password** fields with the new values you want to set.
5. Click **Save**.

Adjusting the maximum search results for the Organizations view

In the Organization View, you can configure the maximum number of search results that are returned when you are searching for employees or organizations.

The default number of the maximum search result is 200. Only 200 or fewer results should be returned and displayed in the results table. If the search query returns more than this number, an error occurs and a message informs you that your results exceed the maximum. In this case, you have two options: either narrow your search query to return fewer results or increase the maximum number of search results.

You use the WebSphere Member Manager (WMM) to configure the maximum search results property. WMM is used by the WebSphere Portal on which the Dashboard Client component is installed to communicate with the LDAP Server (or any user registry server) that contains the employees and organizations information.

To increase the maximum search results, perform the following steps:

1. Edit the file named **wmm.xml** that is located under the <WebSphere_Portal_Installation_Directory>\wmm\ directory using a text editor.
2. Locate the line that contains: *maximumSearchResults='200'*.
3. Change this value to your search results limit. If you type 0 or a negative number, there will be no limit, and all search results will be returned.
4. Save and close the file.
5. Restart the WebSphere Portal.

Setting the work request queue size

For better performance, you need to set the **Work request queue size** value for the **DeserializationWorkManager** to correspond with the **Event processing batch size** system property of the WebSphere Business Monitor.

To set the **Work request queue size** with the value of the **Event processing batch size** system property:

1. In the WebSphere Application Server administrative console, select **Resources** → **Asynchronous beans** → **Work managers**

2. Select the **Server** scope.
3. Click **Apply**.
4. Click **DeserializationWorkManager** in the table.
5. In the **Work request queue size** field, type the value that you specified for the **Event processing batch size** system property of the WebSphere Business Monitor through the System Properties page of the WebSphere Business Monitor administrative console.

Note: Initial setting of 100 is recommended for both values.

6. Click **OK**.
7. In the Message information box, click **Save** to apply your changes. The **Save** page appears.
8. Click **Save**.

Setting the destination queue size

Follow these steps to increase the WebSphere Business Monitor destination queue size.

You may need to increase the maximum size of the destination queue that receives the events messages from the event bus used by the Monitor Server. The need for this is to avoid any event loss in case the Monitor Server application is stopped or events consumption is blocked due to any runtime exceptions, and thus the events messages are not consumed from the bus destination queue so that the queue will be filled of messages and any incoming event message will not be saved in the queue. Also this situation will happen if the rate of the incoming messages is significantly grater than the rate of the message consumption by the Monitor Server, and consequently the queue may get full of messages and event loss will occur.

In order to increase the WebSphere Business Monitor destination queue size:

1. From the WebSphere Application Server administrative console's navigation tree, select **Service Integration** → **Busses**
2. From the **Busses** table, select the bus named **MONITOR.<Cell_Name>.Bus**, where the <Cell_Name> is the name of the cell on which the Monitor Server is installed.
3. From the **Additional Properties** links, click the **Destination** link.
4. From the **Destinations** table, select **Monitor_Bus_Queue_Destination**.
5. From the **Message points** links, click the **Queue points** link.
6. From the Queue points table select **Monitor_Bus_Queue_Destination@<Node_Name>.<Server_Name>-MONITOR.<Cell_Name>.Bus**
7. In the **High message threshold** field, type the value of the maximum queue size you want to use. The default value is 50000 messages.
8. Click **OK**.
9. In the **Message** information box, click **Save** to apply your changes. The **Save** page appears.
10. Click **Save**.

Changing the exception destination for the destination queue

Perform the following steps to change the exception destination for the destination queue that is used by the Monitor Server.

You may need to do this in order to avoid blocking the Monitor Server in case of any runtime exception occurred. You should consider that by performing this task, the events causing runtime exceptions will be ignored.

1. From the WebSphere Application Server administrative console's navigation tree, select **Service Integration** → **Busses**
2. From the **Busses** table, select the bus named **MONITOR.<Cell_Name>.Bus**, where the <Cell_Name> is the name of the cell on which the Monitor Server is installed.
3. From the **Additional Properties** links, click the **Destination** link.
4. From the **Destinations** table, select **Monitor_Bus_Queue_Destination**.
5. In the **Exception destination** group box, select the **System** option button instead of **None**.
6. Click **OK**.
7. In the **Message** information box, click **Save** to apply your changes. The **Save** page appears.
8. Click **Save**.

Configuring CEI Bus on a remote WebSphere Application Server cell

WebSphere Business Monitor performs its operation by reading the messages that contain the events from the queue destination named **Monitor_Bus_Queue_Destination** that gets these events from the Common Events Infrastructure (CEI) Bus. The events are emitted to the CEI Bus by the applications that runs on the BPEL engine in WebSphere Process Server.

WebSphere Business Monitor runs on WebSphere Process Server 6.0.0, and it only supports applications running on the BPEL engine in WebSphere Process Server 6.0.1. Based on this topology, the events are emitted in a CEI Bus that is located in remote WebSphere Application Server cells, which means the event messages must be delivered from this CEI Bus to the foreign queue destination named **Monitor_Bus_Queue_Destination** that is located in the WebSphere Application Server cell that hosts the Monitor Server component.

You can configure this environment by:

1. Set up a Service Integration Bus Link between the bus containing the **Monitor_Bus_Queue_Destination** and a bus in the remote WebSphere Application Server cell that contains the event emitting application (the runtime engine).
2. Define the **Monitor_Bus_Queue_Destination** as a foreign queue destination on the remote WebSphere Application Server cell.
3. Configure the CEI on the remote WebSphere Application Server cell to deliver event messages to the foreign queue destination **Monitor_Bus_Queue_Destination**.

The detailed instructions of performing the above steps can be found on the topic named *Foreign destinations* in the WebSphere Application Server and the WebSphere Process Server documentation.

You can use a provided sample of script files in order to configure the cross cell monitoring environment instead of performing these instructions manually. You can also use these scripts to reset the configuration that you made. The following are the steps of running the cross cell monitoring configuration scripts to configure the CEI Bus in a remote WebSphere Application Server cell and to reset the configuration.

Configuring the remote CEI Bus

Follow these steps to run the cross cell monitoring configuration scripts in order to configure the CEI Bus in a remote WebSphere Application Server cell.

1. On the machine on which the Monitor Server is installed do the following:
 - a. Locate the required script files on the following location:
<Monitor_install_Dir>\install\monsrv\configuration\crosscell. There are six script files with .tcl extension.
 - b. Edit the crossCellParameters.tcl file using any text editor.
 - c. Replace the values of each parameter in this file with the actual value according to your environment settings. The values are between double quotes. All parameters are required.
 - d. Save and close the file.
 - e. Start the WebSphere Application Server on which the Monitor Server is installed if it was stopped.
 - f. From the directory that contains the cross cell configuration scripts, execute the script file named configureMonitorCrossCell.tcl as follows:
<WAS_Home_Dir>\bin\wsadmin -f configureMonitorCrossCell.tcl
 - g. Restart the WebSphere Application Server.
2. On the machine on which the remote WebSphere Application Server exists do the following:
 - a. Copy the cross cell configuration script files from the Monitor Server machine. You must use the exact same crossCellParameters.tcl file.
 - b. Start the WebSphere Application Server in the remote cell if it was not started.
 - c. From the directory containing the cross cell configuration scripts, execute the script file named configureCrossCell.tcl. as follows:
<WAS_Home_Dir>\bin\wsadmin -f configureCrossCell.tcl
 - d. Restart the WebSphere Application Server.

Resetting the remote CEI Bus Configuration

Follow these steps to run the cross cell monitoring configuration scripts in order to reset configuration that has been previously made to configure the CEI Bus in a remote WebSphere Application Server cell.

1. On the machine on which the Monitor Server is installed do the following:
 - a. Locate the required script files on the following location:
<Monitor_install_Dir>\install\monsrv\configuration\crosscell. There are six script files with .tcl extension.
 - b. Edit the crossCellParameters.tcl file using any text editor.
 - c. Replace the values of each parameter in this file with the actual value according to your environment settings. The values are between double quotes. All parameters are required.
 - d. Save and close the file.

- e. Start the WebSphere Application Server on which the Monitor Server is installed if it was stopped.
 - f. From the directory that contains the cross cell configuration scripts, execute the script file named `configureMonitorCrossCell.tcl` as follows:
`<WAS_Home_Dir>\bin\wsadmin -f unconfigureMonitorCrossCell.tcl`
 - g. Restart the WebSphere Application Server.
2. On the machine on which the remote WebSphere Application Server exists do the following:
- a. Copy the cross cell configuration script files from the Monitor Server machine. You must use the exact same `crossCellParameters.tcl` file.
 - b. Start the WebSphere Application Server in the remote cell if it was not started.
 - c. From the directory containing the cross cell configuration scripts, execute the script file named `configureCrossCell.tcl`. as follows:
`<WAS_Home_Dir>\bin\wsadmin -f unconfigureCrossCell.tcl`
 - d. Restart the WebSphere Application Server.

Setting the default DB2 Alphablox theme on an AIX System to the WebSphere Business Monitor theme

Follow these steps in order to set the default theme of the DB2 Alphablox on an AIX machine to the WebSphere Business Monitor theme.

1. In the DB2 Alphablox administrative console, select the **Administration** tab.
2. Click **General**.
3. Click **System** under the **General Properties** section.
4. From the **Default HTML Client Theme** drop down list, select **wbmonitor**.
5. Click **Save**.

Administrative Console

This section provides reference material for the WebSphere Business Monitor administrative console.

Business measures model files

The business measures model files are imported to the Repository database and deployed to the Monitor Server for processing.

To import a business measures model file, you must prepare the State database to receive the imported model data. You create a database table by running the Data Definition Language (DDL) script files that have been generated by the Schema Generator.

The business measures model files that you have to import are imported in a .zip file. This file is verified to ensure correct contents. The file contains the following:

- The EMF file of the business measures model (model.xmi)
 - The XML file of the DB2 Alphablox cube-views metadata (model_cv.xml)
-

Notification data fields

Notification data fields are required to define a notification template.

Notification Data Fields

Data Field	Description
Template name	A template unique name.
Description	A description of the template.
Action service type	Notification types: alert, e-mail, pager, cell phone. Only one type can be chosen.
To (LDAP Query)	A query to retrieve one or more users who are intended to receive the alert. Example: "select users.user_id from users where (&(objectClass=top)(uid=%userid%))" where userid is the field in the common base event that is being replaced.
Subject	The notification subject text with variables that will be substituted with Common Base Events extended-data-element values. Example: "Situation occurred: %BusinessSituationName%"
Body	The notification body text with variables that will be substituted with Common Base Events extended-data-element values. Example: "Gold customer order response time %averageResponseTimeGold% has exceeded the threshold %threshold%"
LDAP Root	The distinguished name for the root of an LDAP query. Example: "cn=root" Leave this field blank if you want to make the top-level distinguished name the default distinguished name.

Supported expression functions

The expression functions described here are supported and handled by the Monitor Server in WebSphere Business Monitor.

The following tables list all expression functions that can be used in the modeled business measures model and that are handled by the Monitor Server to calculate the values of the metrics and KPIs that will be displayed in the dashboards. The tables also list the output behavior of each data-type variation of the function's argument list.

- The output values marked with asterisk (*) cause an overflow exception to be thrown to the log file.
- For all the scalar functions that deal with date, time, and DateTime data types, the time zone used in calculations is GMT. If you defined a date or time with your machine's current local time, the stored date will be different from the specified one. This difference is according to the difference between the local time zone and GMT time zone.
- After the evaluation of any expression, a check is made on the double and float types to assert that an overflow did not occur. If an overflow occurs, an overflow exception will be thrown.

absolute-value (com.ibm.btools.expression.absoluteValue)

Argument list: Long

Variations	Output
A long value	The absolute value of this long value

Argument list: Double

Variations	Output
A double value	The absolute value of this double value
Double.POSITIVE_INFINITY	(*)Double.POSITIVE_INFINITY
Double.NEGATIVE_INFINITY	(*)Double.POSITIVE_INFINITY
Double.NaN	(*)Double.NaN

integer-part (com.ibm.btools.expression.integerValue)

Argument list: Double

Variations	Output
A double value	The integer cast value of the double value
Double.NaN	0
Double.MAX_VALUE	Integer.MAX_VALUE
Double.MIN_VALUE	0
Double.NEGATIVE_INFINITY	Integer.MIN_VALUE
Double.POSITIVE_INFINITY	Integer.MAX_VALUE

decimal-part (com.ibm.btools.expression.fractionalValue)

Argument list: Double

Variations	Output
A double value	The fractional part of the double value
Double.NaN	(*)Double.NaN
Double.NEGATIVE_INFINITY	(*)Double.NEGATIVE_INFINITY
Double.POSITIVE_INFINITY	(*)Double.POSITIVE_INFINITY

if-then-else (com.ibm.btools.expression.ifthenelse)

Argument list: bBoolean: test, String: thenTerm, String: elseTerm

Variations	Output
Boolean, String, String	If test is true, thenTerm is returned; otherwise, elseTerm is returned.

Argument list: boolean: test, double: thenTerm, double: elseTerm

Variations	Output
Boolean, double, double	If test is true, thenTerm is returned; otherwise, elseTerm is returned.

Argument list: boolean: test, boolean: thenTerm, boolean: elseTerm

Variations	Output
Boolean, boolean, boolean	If test is true, thenTerm is returned; otherwise, elseTerm is returned.

attribute-is-set (com.ibm.btools.expression.bom.ome.isSet)

Argument list: EventWrapper: cbe, String: propertyName

Variations	Output
EventWrapper, String	Verifies whether the given property is found in the CBE, and if found, whether it is set. This method will not return true unless the property is found and set in the CBE. The user must check whether a property is set. It can have an alternative path if it is not set by using the if-then-else function. A property used in an expression and not set in an event causes the Monitor Server to throw a hard exception and to be suspended.

greater-value (com.ibm.btools.expression.maxValue)

Argument list: long: var1, long: var2

Variations	Output
long, long	Returns the maximum value.

Argument list: double: var1, double: var2

Variations	Output
double, double	Returns the maximum value.
Double.NaN, double Or double, Double.NaN	(*)Returns Double.NaN
Double.POSITIVE_INFINITY, double Or double, Double.POSITIVE_INFINITY	(*)Returns Double.POSITIVE_INFINITY
Double.NEGATIVE_INFINITY, Double.NEGATIVE_INFINITY	(*)Returns Double.NEGATIVE_INFINITY
Double.NEGATIVE_INFINITY, double Or double, Double.NEGATIVE_INFINITY	Returns the other double value.

Argument list: Double: var1, long: var2

Variations	Output
double, long	Returns the maximum value.
Double.POSITIVE_INFINITY, long	(*)Double.POSITIVE_INFINITY
Double.NaN, long	(*)Double.NaN
Double.NEGATIVE_INFINITY, Long.MAX_VALUE	((double) Long.MAX_VALUE)
Double.NEGATIVE_INFINITY, Long.MIN_VALUE	((double) Long.MIN_VALUE)

Argument list: long: var1, double: var2

Variations	Output
long, double	Returns the maximum value.
long, Double.POSITIVE_INFINITY	(*)Double.POSITIVE_INFINITY
long, Double.NaN	(*)Double.NaN
Long.MAX_VALUE, Double.NEGATIVE_INFINITY	((double) Long.MAX_VALUE)
Long.MIN_VALUE, Double.NEGATIVE_INFINITY	((double) Long.MIN_VALUE)

lesser-value (com.ibm.btools.expression.minValue)

Argument list: long: var1, long: var2

Variations	Output
long, long	Returns the minimum value.

Argument list: double: var1, double: var2

Variations	Output
double, double	Returns the maximum value.
Double.NaN, double Or double, Double.NaN	(*)Returns Double.NaN
Double.NEGATIVE_INFINITY, double Or double, Double.NEGATIVE_INFINITY	(*)Returns Double.NEGATIVE_INFINITY
Double.POSITIVE_INFINITY, Double.POSITIVE_INFINITY	(*)Returns Double.POSITIVE_INFINITY

Argument list: Double: var1, long: var2

Variations	Output
Double.MIN_VALUE, Long.MIN_VALUE	((double) Long.MIN_VALUE)
Double.MAX_VALUE, Long.MAX_VALUE	((double) Long.MAX_VALUE)
Double.NaN, long	(*)Double.NaN
Double.NEGATIVE_INFINITY, long	(*)Double.NEGATIVE_INFINITY
Double.POSITIVE_INFINITY, long	(double) long

Argument list: long: var1, double: var2

Variations	Output
Double.MIN_VALUE, Long.MIN_VALUE	((double) Long.MIN_VALUE)
Double.MAX_VALUE, Long.MAX_VALUE	((double) Long.MAX_VALUE)
Double.NaN, long	(*)Double.NaN
Double.NEGATIVE_INFINITY, long	(*)Double.NEGATIVE_INFINITY
Double.POSITIVE_INFINITY, long	(double) long

includes-specified-text (com.ibm.btools.expression.containsText)

Argument list: String: text, String: subText

Variations	Output
String, String	Looks up subText in the text and returns true if subText is contained in the text; otherwise, it returns false.
"" , String Or String, ""	false
null, String Or String, null	Throws NullPointerException

starts-with-specified-text (com.ibm.btools.expression.startsWithText)

Argument list:String: text, String: subText

Variations	Output
String, String	If text starts with subText, true is returned; otherwise, false is returned.
"" , String	false
String, ""	true
null, String Or String, null	Throws NullPointerException

to-uppercase (com.ibm.btools.expression.toUpperCase)

Argument list:String: text

Variations	Output
String	Returns text in uppercase letters.
""	""
null	Throws NullPointerException

to-lowercase (com.ibm.btools.expression.toLowerCase)

Argument list:String: text

Variations	Output
String	Returns text in lowercase letters.
""	""
null	Throws NullPointerException

Concatenate (com.ibm.btools.expression.concat)

Argument list:String: s1, String: s2

Variations	Output
String, String	Returns s1 + s2
null, String	"null" + s2
String, null	s1 + "null"
null, null	"nullnull"
"", String	s2
String, ""	s1

current-event-creation-time (com.ibm.btools.expression.currentTime)

Argument list:EventWrapper: currentEvent

Variations	Output
EventWrapper	Returns the creation time of the event in milliseconds.

current-event-creation-date (com.ibm.btools.expression.currentTime)

Argument list:EventWrapper: currentEvent

Variations	Output
EventWrapper	Returns the creation date of the event in milliseconds.

current-event-creation-datetime (com.ibm.btools.expression.currentTime)

Argument list:EventWrapper: currentEvent

Variations	Output
EventWrapper	Returns the creation date-time of the event in milliseconds.

Add-To-String-List (com.ibm.btools.expression.bom.ome.addToList)

Argument list:String: list, String: addItem, Boolean: duplicate

Variations	Output
String, String, boolean	An element is added to a comma-separated list. If duplicate is true, duplicates are accepted inside the list; otherwise, the list does not accept duplicates.
String, "", (false or true)	The list with a trailing comma, for example, "hhh,kkk,"
"", element, (false or true)	A string containing an element
"" , "" ,(false or true)	"" , empty string
null, String, (false or true)	A string containing an element
String, null, (false or true)	Throws NullPointerException

Remove-From-String-List (com.ibm.btools.expression.bom.ome.removeFromList.string)

Argument list:String: list, String: removedItem, boolean: allOccurrences

Variations	Output
String, String, boolean	Removes an element from a comma-separated list. An option is given whether to remove all occurrences inside the list.
null, String, (false or true)	""
"" , String, (false or true)	""
String, "", (false or true)	String of the list
String, null, (false or true)	Throws NullPointerException

number-to-text (com.ibm.btools.expression.bom.ome.numberToString)

Argument list:double: number

Variations	Output
Double	Returns the string of the given number.
Double.POSITIVE_INFINITY	Infinity
Double.NEGATIVE_INFINITY	-Infinity
Double.NaN	NaN

date-to-datetime **(com.ibm.btools.expression.bom.ome.dateToDateTime)**

Argument list:long: date

Variations	Output
long	Converts the given date to date-time

datetime-to-date **(com.ibm.btools.expression.bom.ome.dateTimeToDate)**

Argument list:long: date

Variations	Output
long	The date time will be truncated to remove the date. The truncated date is then changed into milliseconds.

datetime-to-time (com.ibm.btools.expression.dateTimeToTime)

Argument list:long: date

Variations	Output
long	Converts the given date-time to time. It sets the value of the date to zero, and the time is represented in milliseconds.

text-to-integer **(com.ibm.btools.expression.bom.ome.textToInteger)**

Argument list:String: string

Variations	Output
String	Returns the integer value of the given string text.
An unformatted string	Throws NumberFormatException
null	Throws NullPointerException

text-to-number **(com.ibm.btools.expression.bom.ome.stringToNumber)**

Argument list:String: string

Variations	Output
String	Returns the number found in the string in the form of double.
An unformatted string	Throws NumberFormatException
null	Throws NullPointerException

text-to-date (com.ibm.btools.expression.bom.ome.stringToDate)

Argument list:String: string, String: format

Variations	Output
String, String	Formats the given string into a date format. Because the date format in the system is represented as long, this method returns long.
String, null Or null, String	Throws NullPointerException
String, "" Or "", "" Or "", String	Throws ParseException

text-to-time (com.ibm.btools.expression.bom.ome.stringToTime)

Argument list:String: string, String: format

Variations	Output
String, String	Formats the given string into a date format; because the date format in the system is represented as long, this method returns long.
String, null Or null, String	Throws NullPointerException
String, "" Or "", "" Or "", String	Throws ParseException

Important: This function may cause an IllegalArgumentException. For modelling metrics that are populated from the event, use the text-to-datetime function instead.

text-to-datetime (com.ibm.btools.expression.bom.ome.stringToDateTime)

Argument list:String: string, String: format

Variations	Output
String, String	Formats the given string into a date format; because the date format in the system is represented as long, this method returns long.
String, null Or null, String	Throws NullPointerException
String, "" Or "" , "" Or "" , String	Throws ParseException

text-to-duration (com.ibm.btools.expression.stringToDuration)

Argument list:String: string, String: format

Variations	Output
String, String	Formats the given string into a duration format. Because the duration format in the system is represented as long, this method returns long. The format must be dddd-hh-mm-ss; if any other format, a NoSuchElementException is raised.
"" , String	Throws NoSuchElementException
null, String	Throws NullPointerException

Data types

A list of all supported data types in business measures models, their sizes, and the corresponding data type name in both Java and DB2.

Business Measures ModelTypes	Size	Java Type used	DB2 8 for Linux®, UNIX® and Windows
Boolean	1 Bit	boolean	SMALLINT
String with "IsInternational" descriptor	length in chars	String	VARCHAR Length or (Length * 3)
Date	8 Bytes	long	BIGINT
Time	8 Bytes	long	BIGINT
DateTime	8 Bytes	long	BIGINT

Business Measures ModelTypes	Size	Java Type used	DB2 8 for Linux®, UNIX® and Windows
Duration	8 Bytes	long	BIGINT
Long	8 Bytes	long	BIGINT
Integer	4 Bytes	int	INTEGER
Short	2 Bytes	short	SMALLINT
Byte	1 Byte	byte	SMALLINT
Double	Approx. in 64 Bits	double	DOUBLE
Float	Approx in 32 Bits	float	REAL

Java primitive data types and precision

The following tables list the Java primitive data types, their description, size, and format:

Integers data types

Keyword	Description	Size	Format
byte	Byte-length integer	8-bit twos complement	From +127 to -128
short	Short integer	16-bit twos complement	From +32767 to -32768
int	Integer	32-bit twos complement	From +2147483647 to -2147483648
long	Long integer	64-bit twos complement	From +9223372036854775807 to -9223372036854775808

Real numbers data types

Keyword	Description	Size	Format
float	Single-precision floating point	32-bit IEEE 754	From 3.4028235E38 to 1.4E-45
double	Double-precision floating point	64-bit IEEE 754	From 1.7976931348623157E308 to 4.9E-324

Other data types

Keyword	Description	Size/Format
char	A single character	16-bit Unicode character
Boolean	A Boolean value (true or false)	true or false

DB2 data types range and precision

A list of the DB2 data types, their range and precession.

Data type	Range	Precision
SMALLINT	-32 768 to 32 767	5 digits
BIGINT	-9 223 372 036 854 775 808 to +9 223 372 036 854 775 807	19 digits
INTEGER	-2 147 483 648 to +2 147 483 647	10 digits
DOUBLE	From 1.79769E+308 to -2.225E-307 Or From 2.225E-307 to 1.79769E+308	15 digits after the decimal point
REAL	From -3.402E+38 to -1.175E-37 Or From 1.175E-37 to 3.402E+38	7 digits after the decimal point
DECIMAL	-10**31+1 to 10**31-1	The defined maximum number of digits. For example, the precision of a column defined as NUMERIC(10,3) is 10

Administration troubleshooting

As you administer WebSphere Business Monitor, some problems may occur. The following information will help you to troubleshoot those problems.

Changing the user ID and password for the created authentication alias

If you are planning to enable WebSphere Application Server global security after installing the Monitor Server, you must first update the authentication alias with a valid user ID and password.

On the Monitor Server machine

When you install the Monitor Server in an unsecured WebSphere Application Server environment, the Launchpad creates an authentication alias in the WebSphere Application Server: *MonitorBusAlias*. This alias contains a default user ID called *messaging*.

If you enable the WebSphere Application Server global security after installation without updating the *MonitorBusAlias* alias with a valid User ID and password (with authority to access the WebSphere Application Server), a *ResourceAllocationException* exception will be thrown when you restart the WebSphere Application Server after enabling global security. The Monitor Server application will not start.

To avoid this error:

1. You must change the user ID and password of the *MonitorBusAlias* alias, the *ActionManagerBusAuth* alias, and the SCA alias before enabling the WebSphere Application Server global security. To update these aliases, complete the following steps:
 - a. In the navigation tree of the WebSphere Application Server administrative console, select **Security** → **Global Security**.
 - b. Select **JAAS Configuration** → **J2C Authentication Data**.
 - c. From the table, select **MonitorBusAlias**.
 - d. On the **MonitorBusAlias** information page, in the **User ID** and **Password** fields, type a valid user ID and password.
 - e. Click **OK**.
 - f. In the Message information box, click **Save** to apply your changes. The Save page appears.
 - g. Click **Save**.
 - h. Repeat the steps above to update *ActionManagerBusAuth* alias, and SCA alias
2. Set the authentication alias property of the SIBus Link to the *MonitorBusAlias*. Refer to the topic named *Security considerations for service integration buses* in the WebSphere Application Server for details.
3. Run the provided script files in order to configure the cross cell monitoring environment. Refer to the topic named *Configuring CEI Bus on a remote WebSphere Application Server cell* for details.
4. Restart the WebSphere Application Server.

On WebSphere Process Server machine

1. Set the privilege user ID before enable global security as follows:
 - a. Click **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Local Security Policy**.
 - b. In the Local Security Settings window, select **Local Policies** → **User Rights Assignments**
 - c. Double click the **Act as part of the operating system** policy.
 - d. In the **Act as part of the operating system Properties** dialog box, add the privileged user ID.
 - e. Click **OK**.
2. Run the provided script files in order to configure the cross cell monitoring environment on the WebSphere Process Server machine. Refer to the topic named *Configuring CEI Bus on a remote WebSphere Application Server cell* for details.
3. Create a new authentication alias (e.g. MonitorBusAlias).
Refer to the topic named *Java 2 Connector authentication data entry settings* in the WebSphere Application Server documentation for details.
4. Set the authentication alias property on the WebSphere Business Monitor bus (e.g. to MonitorBusAlias).
Refer to the topic named *Security considerations for service integration buses* in the WebSphere Application Server documentation for details.
5. Set the authentication alias property on the Service Integration Bus link (e.g. to MonitorBusAlias).
Refer to the topic named *Default messaging provider settings* in the WebSphere Application Server documentation for details.
6. Set the authentication alias property on the MonitorQueueFactory JMS queue connection factory (e.g. to MonitorBusAlias).
Refer to the topic named *Adding a service integration bus link* in the WebSphere Application Server documentation for details.
7. Run the following commands using the **wsadmin** command window. You cannot run these commands using the WebSphere Application Server administrative console. You should replace the variable (**{XXXX}**) with the proper values:

```
$AdminTask addUserToForeignBusRole { -bus ${PROCESS_SERVER_BUS_NAME}
-foreignBus ${MONITOR_SERVER_BUS_NAME} -role Sender -user ${USER_NAME} }
$AdminTask addUserToDestinationRole { -type foreignDestination -bus
${PROCESS_SERVER_BUS_NAME} -foreignBus ${MONITOR_SERVER_BUS_NAME}
-destination Monitor_Bus_Queue_Destination -role Sender -user
${USER_NAME} }
$AdminConfig save
```
8. Restart the WebSphere Application Server.

Changing user passwords on the Monitor Server components' machine

When you change the password of a user ID used by WebSphere Business Monitor components on the system without also changing the same password in the components' locations, the components will not respond. You must complete the password change to ensure communication with the WebSphere Business Monitor components.

On the machine on which the Dashboard Client is installed, complete the following steps:

1. Stop the WebSphere_Portal application server on the WebSphere Application Server by running the following command:

On the Windows platform: *stopServer WebSphere_Portal*

On the AIX platform: *stopServer.sh WebSphere_Portal*

Note: If the WebSphere Application Server global security feature is enabled, provide the user ID and password of a valid user who can start and stop the WebSphere Application Server with the above commands. For example, *stopServer WebSphere_Portal user <User_ID> password <Password>*

2. Stop DB2 by running the *db2stop force* command
3. Start DB2 by running the *db2start* command
4. Start the WebSphere_Portal application server on the WebSphere Application Server by running the following command:

On the Windows platform: *startServer WebSphere_Portal*

On the AIX platform: *startServer.sh WebSphere_Portal*

On the machine on which the Monitor Server is installed, complete the following steps:

1. Open the DB2 Control Center.
2. From the menu, select **Tools** → **Configuration Assistant**.
3. In the databases table, right-click each WebSphere Business Monitor database whether it was created locally on the machine or cataloged from a different machine.
4. From the pop-up menu, select **Change Password**.
5. Change the password you have changed for any DB2 user ID (db2admin on the Windows platform or db2inst1 on the AIX platform).
6. Start the WebSphere Application Server administrative console, and do the following:
 - a. From the navigation tree, select **Resource** → **JDBC Provider**.
 - b. Select **MonitorDB2XADriver**.
 - c. Click **Data Sources**.
 - d. Select **MonitorDataSource**.
 - e. Click **J2EE Connector Architecture (J2C) authentication data entries**.
 - f. Select each alias that has the DB2 user ID whose password you have changed (db2admin on the Windows platform or db2inst1 on the AIX platform).
 - g. On the **General Properties** page of each alias, in the **Password** field, change the password to the new password.
 - h. Click OK, and save your changes.
 - i. Log off the administrative console.
7. Stop the server1 application server on the WebSphere Application Server by running the following command:

On the Windows platform: *stopServer server1*

On the AIX platform: *stopServer.sh server1*
8. Restart the server1 application server by running the following command:

On the Windows platform: *startServer server1*

On the AIX platform: *startServer.sh server1*

The imported business measures model runs successfully, but no data is stored in the databases

The problem: A business measures model was imported successfully into WebSphere Business Monitor administrative console. Its corresponding Business Process Execution Language (BPEL) .ear file was successfully installed on the WebSphere Process Server. The process instances that belong to this model were created and are running successfully. The State database tables were created successfully. But no related data has been created and stored in the State database.

The following steps can help you to investigate and solve this problem:

1. Enable the log details level on the finest level. Refer to Managing the logging and tracing for details on how to set the log details level.
2. Enable the Common Event Infrastructure (CEI) data store. Refer to Enabling and disabling the CEI store for details.
3. Create a new process instance in the runtime engine so that new events will be sent to the Monitor Server.
4. Check that the events are generated and sent successfully to the CEI by logging on to the Common Base Event Viewer application Web page and searching for the newly created events. Refer to the topic named *Viewing monitored events* in WebSphere Process Server documentation for details.
5. If the events do not exist in the CEI data store, check the BPEL model deployment on the WebSphere Process Server. Refer to the topic named *Troubleshooting a failed deployment* in WebSphere Process Server documentation for details on troubleshooting BPEL deployment.
6. If the events exist in the CEI datastore, check that the JMS resources are created and configured successfully.
 - a. Go to the WebSphere Application Server administrative console by selecting **Resources** → **JMS providers** → **Default messaging** → **Server** .
 - b. Select **JMS queue connection factory** under the **Connection Factories** section, and verify that the JMS queue connection factory for WebSphere Business Monitor is created with the name *MonitorQueueFactory*.
 - c. Select **Resources** → **JMS providers** → **Default messaging** → **Server** .
 - d. Select **JMS queue** under the **Destinations** section, and verify that JMS queue is created with the name *MonitorQueueDestination*.
7. Check the trace.log, for exceptions. If there are none, trace the filtering and correlation trace of the events. If the trace feature is enabled and the logging level is set to the finest level, you should find that all events received by WebSphere Business Monitor are in the trace file. Follow the trace for the first of these events (or any creation event) to see how it was filtered by each registered process. For each process registered for the event, a message indicates that the filterAndHandle method was started and whether the event passed the filter. If the filter was successful, follow the correlation to see how many instances were returned by the query for this event entry. If it is a creation event (for example, BPC.BFM.PROCESS.STATUS), the query should return 0 instances. The creation of a new record for this process or activity instance should be triggered.

Business measures model import failure due to the usage of user defined metric name similar to a predefined metric name

The import of the business measures model may fail if the imported business measures models contains a user defined metric that its name is similar to any predefined metric name.

You must not define any metric in the business measures model when you are modeling it in the Business Measures editor with the name of any predefined metrics. This will cause the importing of the business measures model in the WebSphere Business Monitor administrative console to fail and the following exception will be thrown:

```
COM.ibm.db2.jdbc.DB2Exception: [IBM][CLI Driver][DB2/NT] SQL0803N One or more values in the INSERT statement, UPDATE statement, or foreign key update caused by a DELETE statement are not valid because the primary key, unique constraint or unique index identified by "2" constrains table "REPOS.META_MONITOR_METRIC" from having duplicate rows for those columns. SQLSTATE=23505
at COM.ibm.db2.jdbc.app.SQLExceptionGenerator.throwSQLException(Unknown Source)
at COM.ibm.db2.jdbc.app.SQLExceptionGenerator.throwSQLException(Unknown Source)
at COM.ibm.db2.jdbc.app.SQLExceptionGenerator.check_return_code(Unknown Source)
at COM.ibm.db2.jdbc.app.DB2PreparedStatement.execute2(Unknown Source)
at COM.ibm.db2.jdbc.app.DB2PreparedStatement.execute(Unknown Source)
at com.ibm.wbimonitor.repository.importer.OMEntity.insertEntity(OMEntity.java:714)
at com.ibm.wbimonitor.repository.importer.Metric.accept(Metric.java:48)
at com.ibm.wbimonitor.repository.importer.RepositoryManager.importModelList(RepositoryManager.java:409)
at com.ibm.wbimonitor.repository.importer.RepositoryManager.repositoryImport(RepositoryManager.java:234)
at com.ibm.wbimonitor.repository.importer.RepositoryManager.repositoryImport(RepositoryManager.java:198)
at com.ibm.wbimonitor.repository.importer.Run.main(Run.java:60)
```

The following is a list of all predefined names that should not be used for the names of any metrics, counters and stopwatches in the Business Measures editor.

Predefined names in the Business Measures editor

Process	Local process and loops
Process Instance Parent ID	
Process Instance ID	
Process Instance Name	
Process Instance Description	
Start Time	Start Time
Elapsed Duration	Elapsed Duration
Working Duration	Working Duration
State	State
Is Delayed	
Caller MC Definition Id	
Caller MCIID	
Caller Table Physical Name	
	Instance Parent ID

Process	Local process and loops
	Instance Description

DB2 start on AIX 5.2 and AIX 5

Problem: You are trying to open the databases from the DB2 Control Center on the AIX Version 5.2 or AIX Version 5 platform, and a communication error message appears. DB2 generates this error if DB2 is not started.

Solution

- Start DB2.
- If an error occurred during DB2 startup, run the following DB2 command (as root) to refresh, renovate, and update DB2 :

```
/usr/opt/db2_08_01/instance/db2iupdt <instance_owner_ID>  
su db2inst1  
db2start
```

Increasing shared memory segments

When WebSphere Business Monitor attempts to process an event on the AIX platform , an SQL error is returned. And also when WebSphere fails to initialize database connection, both during startup or event processing. The failure results from an operating-system-imposed limit on the number of shared memory segments available to IBM DB2.

when a *Test Connection* from WebSphere Admin Console (jdbc resources) fails with the following error:

Test connection failed for data source MonitorDataSource on server server1 at node Node01 with the f

This error likely indicates that the number of shared memory segments allowed by AIX to the database has been exceeded.

When this error occurs, you must change the DB2 configuration to increase the number of shared memory segments. Currently, recovery expert does not support the use of defining a DB2 alias with a loopback to avoid the shared memory problem.

Solution

An extended shared memory feature is used by DB2 that circumvents this problem. To enable this feature, place the following lines near the beginning of both `/etc/rc.db2` and `<instance_home>/sqlib/db2profile`, (where `<instance_home>` is the home directory of each instance user; for example, `/home/db2inst1`) after the block comment, but before any executable lines:

```
# Local addition to fix AIX shared memory problem:  
EXTSHM=ON  
export EXTSHM
```

Note: The entry is case sensitive. Place the entry in the db2profiles of all instance users and the administrator server user. Then log in as each instance user and the administrator server user, and execute the following command:

```
db2set DB2ENVLIST=EXTSHM
```

This sets a DB2 profile variable within each instance that causes the value of the EXTSHM environment variable to be included in the environment of the DB2 daemon processes as they are started. Defining EXTSHM in the db2profiles of each instance, which are executed at login, ensures that the variable is set in any instance-owner environment. Finally, putting it in /etc/rc.db2 ensures that the variable is set when the DB2 processes are started at startup time. Finally, restart the system to ensure that all DB2 processes are started with EXTSHM=ON in their environment.

If you want the fix to be automatically applied to new instances when they are created in the future, add the lines that set and export EXTSHM to the file /usr/lpp/db2_08_02/cfg/db2profile, which is copied to <instance_home>/sqlib/db2profile on instance creation. To ensure that the DB2ENVLIST profile variable is also set, add this code after the variable INSTHOME is set:

```
if [ -x $INSTHOME/sqlib/adm/db2set ]
then if [ "$INSTHOME/sqlib/adm/db2set DB2ENVLIST" != "EXTSHM" ]
then $INSTHOME/sqlib/adm/db2set DB2ENVLIST=EXTSHM
fi
fi
```

This code causes the DB2ENVLIST instance profile variable to be set the first time a new instance is used.

Note:

- It is assumed that DB2Version 8.0 is installed under /usr/lpp/db2_08_02. If your installation is elsewhere, you must work with your install location.
- For DB2 version 8.0, you should add the following lines to the <instance_home>/sqlib/userprofile file, if it exists; otherwise, create a user profile file with permissions 755 . The lines are :
EXTSHM=ON
export EXTSHM

Notices and Trademarks

Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.*

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*Lab Director
IBM RTP Laboratory
3039 Cornwallis Road
P.O. BOX 12195
Raleigh, NC 27709-2195
U.S.A*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not necessarily tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

COPYRIGHT LICENSE

This information may contain sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries, or both:

IBM
IBM (logo)
WebSphere
DB2
Tivoli
MQSeries
AIX
z/OS

Excel, Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, MMX, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

ALPHABLOX is a registered trademark of Alphablox Corporation in the United States, other countries, or both.

Adobe is trademark of Adobe Systems Incorporated in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.