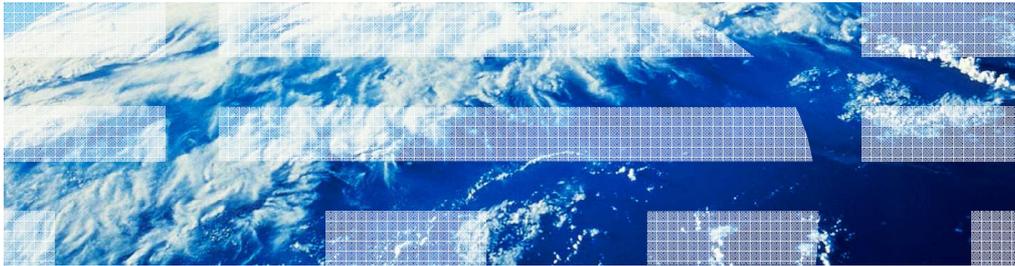


WebSphere Business Monitor

Monitor models



© 2010 IBM Corporation

This presentation should provide an overview of monitor models in WebSphere Business Monitor.

Goals

- Introduce monitor models and the monitor model editor in WebSphere Business Monitor

This presentation will give you an overview of monitor models and the monitor model editor in WebSphere Business Monitor.

Agenda

- What is a monitor model?
- Monitor model elements
- Monitor model editor
- Monitor details model

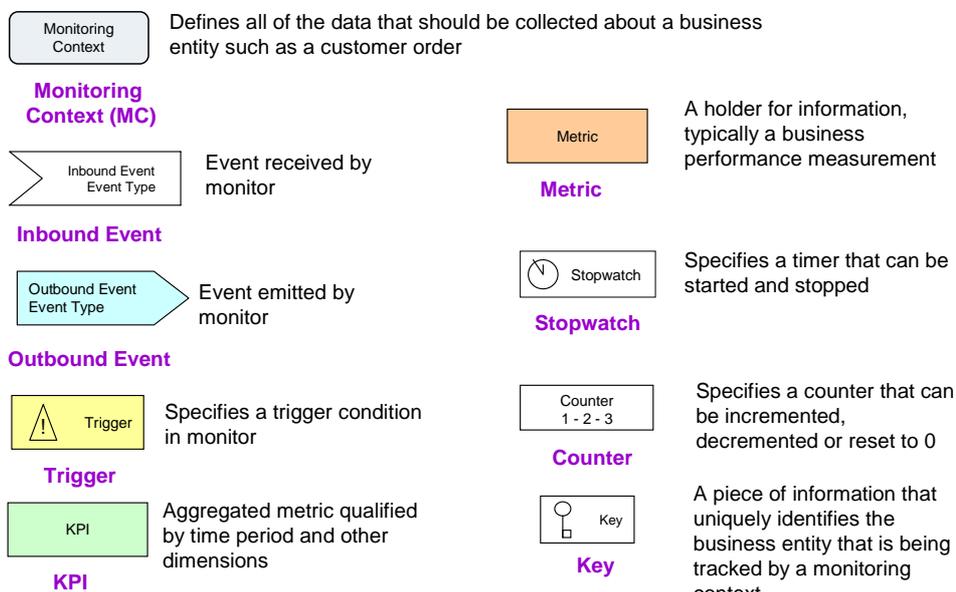
The agenda for this presentation begins with a discussion of the concept of monitor models. Then it provides an overview of the different elements contained in a monitor model and how they interact to monitor your applications. The monitor model editor is reviewed, and then the monitor details model is covered. There are other sub-models in a monitor model, but they are discussed in another presentation.

What is a monitor model?

- An application used for monitoring events emitted by applications or processes
- Monitor model applications are run on WebSphere Business Monitor server
- The monitor model editor is used to create the monitor models

A monitor model is an application used for monitoring events emitted by applications or processes. Monitor model applications are then run on WebSphere Business Monitor server to display business measures in the dashboard using various dashboard widgets. The Monitor model editor is an editor that you use to create monitor models. It is provided as a plug-in to WebSphere Integration Developer. The Monitor model can be based on BPEL processes or any other application that can be instrumented to emit events.

Monitor model elements



5

Monitor models

© 2010 IBM Corporation

A monitor model consists of different elements that are used to define business measures that you want to monitor.

A monitoring context (MC) definition defines all of the data that should be collected about a business entity such as a customer order. An MC instance is created for each unique entity that is being monitored.

An inbound event is used to associate an event definition with an event that Monitor will process.

An outbound event is used to associate an event definition with an event that Monitor will emit. You can use outbound events for business alerts for notifying users of business situations that should be corrected.

Triggers are used to tell Monitor when to capture and calculate values and are used by metrics, stopwatches and counters.

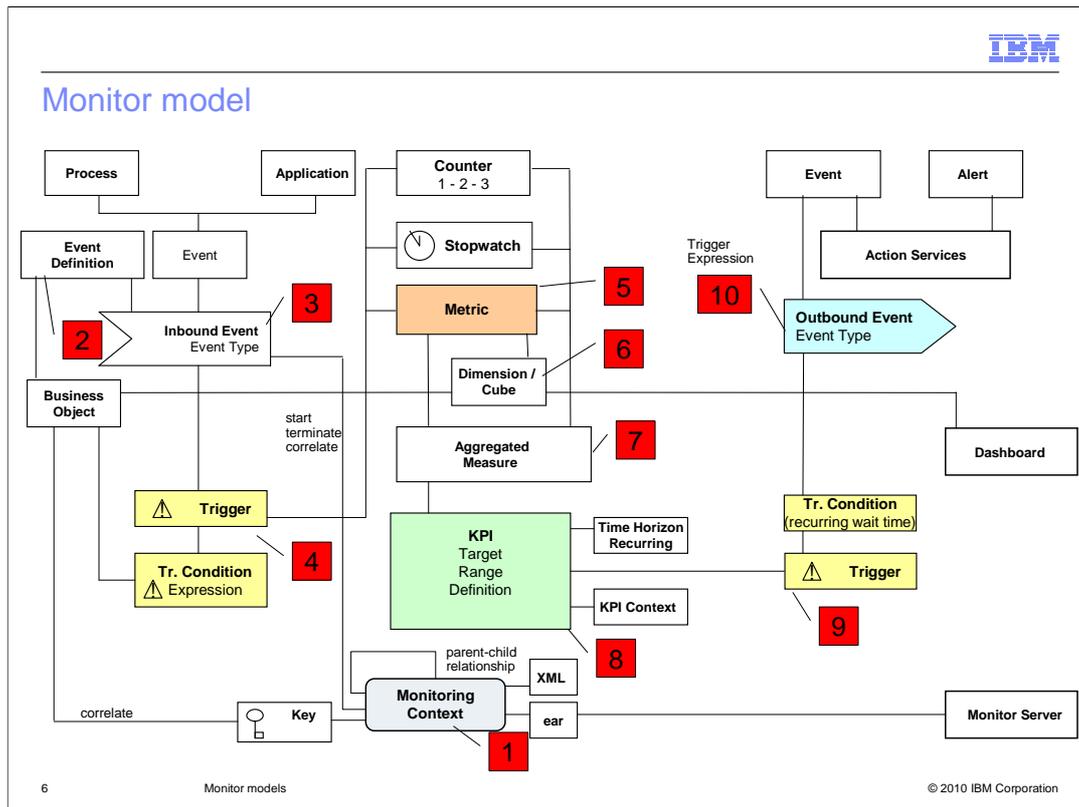
Key performance indicators (KPIs) are quantifiable measurements of the improvement or deterioration in the performance of an activity critical to the success of a business.

A metric is used to store payload information from the inbound events, and as a source for aggregation of other business measures.

Stopwatches provide a pre-built mechanism for timing durations between events. You specify triggers that cause the stopwatch to start, stop or reset.

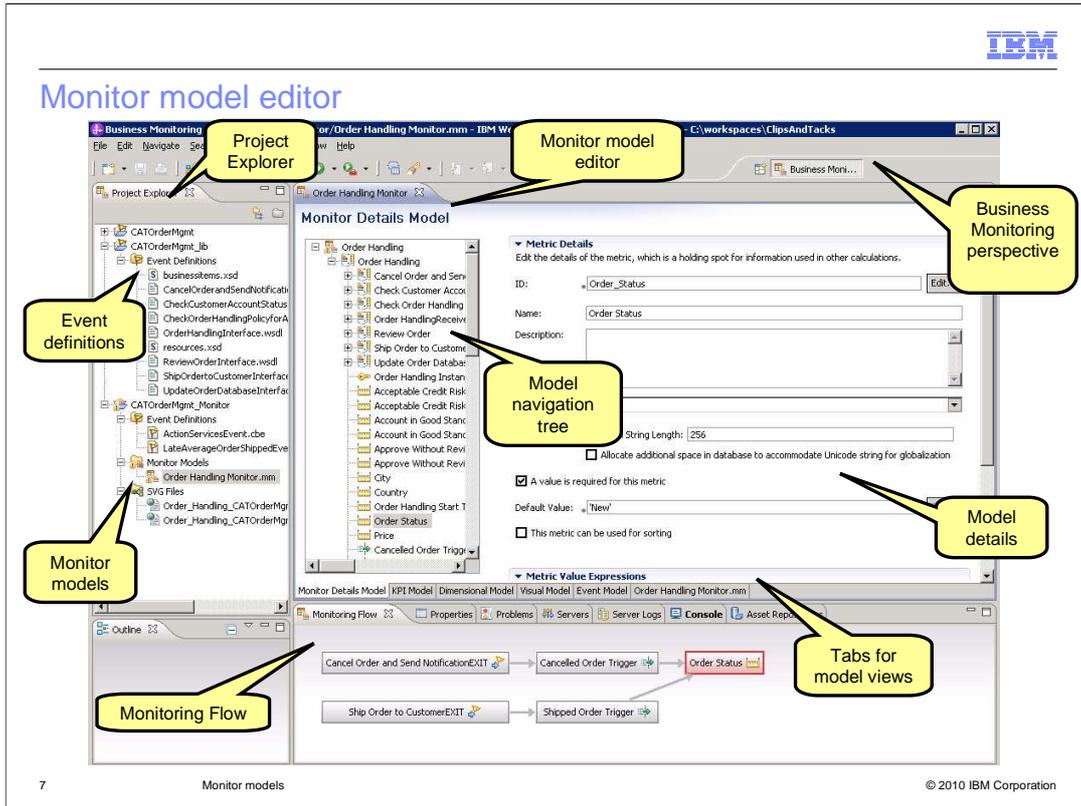
Counters provide a pre-built mechanism for counting how many times things happen within your model. When triggers are fired, you can add one to a counter, subtract one from the counter or reset the counter to zero.

Keys are used to correlate the events coming into the model with the correct monitoring context instances. The key value can be set only once during the lifetime of the MC instance and must be unique among peer MC instances.



This chart shows how the different monitor elements are used for monitoring your processes and applications.

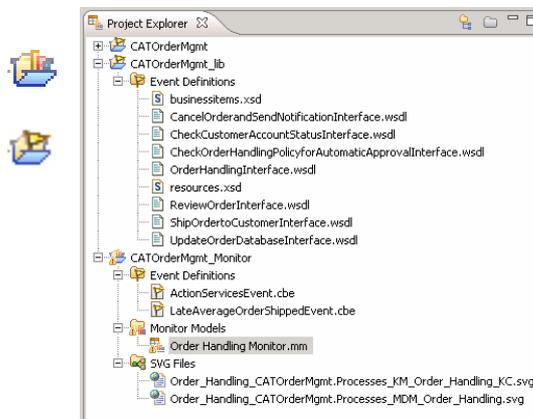
Number 1 references a monitoring context which is the parent container for your monitor data. A key is defined in the model and refers to a business element that provides a unique identifier for each monitored instance. Customer order number is an example of a unique business key. Number 2 shows that an event definition defines the business payload for an event that comes in for monitoring. Number 3 points to an inbound event at runtime which contains your business objects and are processed by your monitor model. Typically these inbound events are used to instantiate and terminate monitoring context instances. They are also used to fire triggers which will populate metrics in the model with business data. Number 4 shows a trigger which can fire based on an inbound event or timer event. Number 5 is a metric which is used to capture business payload from an inbound event. Number 6 refers to cubes which are used to aggregate numerical metrics so that you can organize them into specific groupings called dimensions. In number 7 you see the aggregated measure which performs a function on a metric. So you might have an average function applied to customer order amount. These aggregated measures are used in dimensional analysis. In number 8, Key Performance Indicators (KPI's) also use aggregated information to display performance relative to targets. Number 9 shows a trigger which can be used based on the value of a KPI exceeding a threshold which then will fire an outbound event. In number 10 the outbound event is used by monitor action services to alert a user of a business situation that requires corrective action.



The Monitor model editor contains a business monitoring perspective where you will do most of your work. The project explorer view on the left shows a list of the monitor models that you create. The monitor model editor on the right contains tabs for each of the five models: monitor details model, KPI model, dimensional model, visual model and event model. Also there is a tab for the XML editor that is used for editing the monitor XML file. At the bottom, the monitoring flow view shows relationships between elements in the model.

Project explorer

- Two project types
 - Monitor modeling project capable of containing both monitor models and event definitions
 - Monitor support project capable of containing only event definitions
- Wizards to
 - Create business monitoring project
 - Create monitor model
 - Create event definition
 - Associate projects
- Double click event definition or monitor model to open editors



There are two types of projects in the project explorer, the monitor modeling project and monitor support project. The monitor modeling project can contain monitor models and event definitions. Support projects can only contain event definitions.

Monitor support projects are most commonly seen after generating a monitor model from a BPEL process. Because the event definitions for the BPEL process are generated within the process module project, the monitor model must be able to access those event definitions.

In the project explorer view you will see a list of any event definition files in the project. These are used for defining inbound and outbound events in the model. Also you will see a list of monitor models which are the files with an mm extension. These are XML files that contain the business measures that you want to track in your monitor models.

In the project explorer, there are wizards available to help you create business monitoring projects, monitor models and event definitions. You can use the menu option 'Associate Projects' to associate event definitions in other projects with the current project.

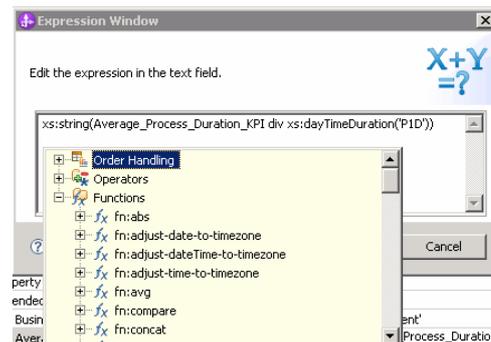
To open the editors on an object, just double click an event definition listed in the project explorer or double click a monitor model in the list.

Expressions

▼ **Outbound Event Content**
Specify the triggers that cause the event to be sent. Use the Expression column to specify the value for each event attribute when the event is sent.

Name	Type	Expression
Average Process Duration KPI Trigger 1		
Property Data		
Extended Data		
BusinessSituationName	A string	'Order Fulfillment Event'
AverageOrderProcessingTime	A string	xs:string(Average_Process_Duration_KPI div xs:dayTimeDuration('P1D'))

- Expressions used for
 - Metric values
 - Trigger conditions
 - Inbound event filters, correlation
 - Outbound event attributes, filters
- Ctrl-space for content assist
- Errors show in problems view and with markers
- Subset of the XML Path Language (XPath) 2.0



9

Monitor models

© 2010 IBM Corporation

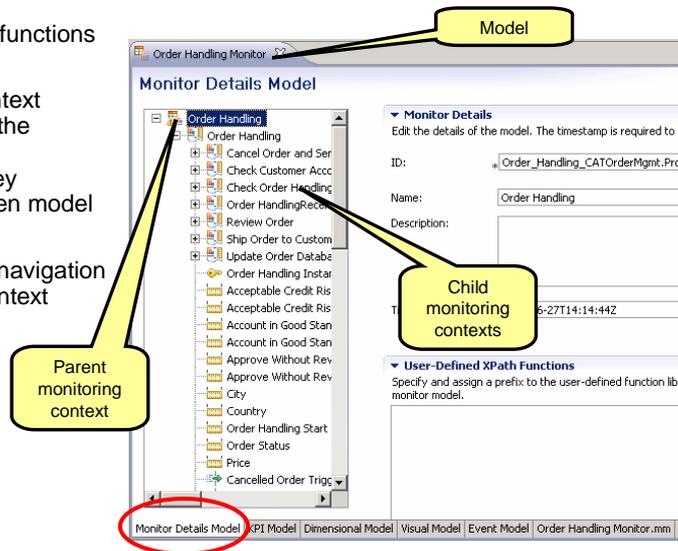
Expressions are used in various places in the monitor model editor to identify content for metrics, conditions and filters. At the top you see an example of an expression used to calculate the data on an outbound event. You can click the button to open the expression editor. On the lower right you see the expression editor. You can type Ctrl-space for content assist. The context of the expression determines the elements that are valid and selectable in content assist. You can also type the expression in the dialog directly without content assist. After entering an expression, validation is performed immediately and errors will show with error markers directly on the expression or with error messages in the problems view.

The content assist window is made up of model elements, operators, and functions, including both built-in functions and user-defined XML Path Language (XPath) functions. Selecting an item from the tree and pressing 'Enter' places the XPath representation of that item in the expression at the cursor position. The content assist window is sensitive to the expression that you are editing, and, when possible, shows only valid options.

Monitor uses a subset of the XML Path Language (XPath) 2.0 to create expressions.

Monitor details model

- Specify user defined XPath functions to be used in the model
- One or more monitoring context definitions can be added to the monitor details model
 - Monitoring context and key automatically created when model is created
- Right click the model in the navigation tree > New > Monitoring Context



10

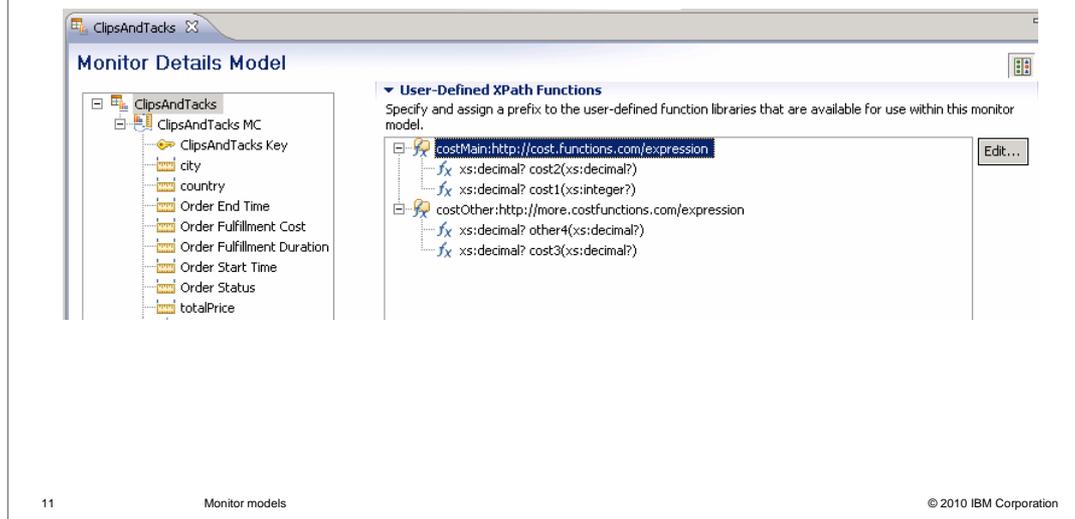
Monitor models

© 2010 IBM Corporation

Shown here is the monitor details model tab. The navigation tree is displayed on the left and you can select an item in the tree to display the details on the right. Here you can specify any user defined XPath functions to use in the model. In the navigation tree, you can create one or more monitoring contexts for a model, but when you create a new model, a monitoring context and key are automatically created for you. To create another monitoring context, right click in the tree and select 'New', then select 'Monitoring Context'.

User defined functions

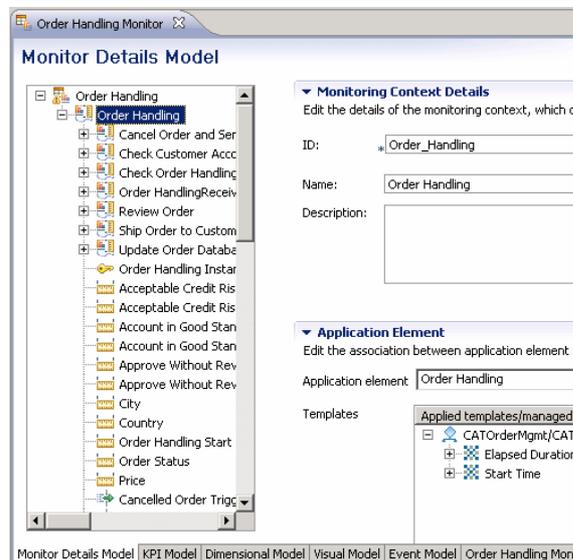
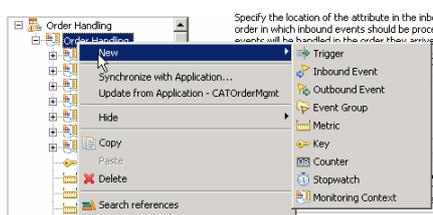
- User defined functions available in the monitor model are listed in the monitor details model and available throughout the model using content assist



You can create your own user defined functions in Java to be used anywhere in the model where there is expression support, including KPI expressions. In this screen capture you can see the user defined function section of the monitor details model, and it lists the functions which you have created and defined to the monitor model. To use these functions you can use content assist while editing expressions, then the user defined functions are listed alongside the built-in functions.

Monitoring context

- On this page you can maintain quality of service data
 - Event sequence path
 - Event creation time
 - Global instance ID
- Single click an item in the tree to see the details for that item
- Right click the monitoring context to create new monitoring elements



12

Monitor models

© 2010 IBM Corporation

If you select the monitoring context in the navigation tree of the monitor model editor, you will see the monitoring context page where you can view the details of the monitoring context. Here you can specify the location of important configuration information such as event sequence path, event creation time, and global instance ID. If you know that events are likely to arrive out of order at run time, you can specify an event sequence path. The Monitor server uses the event creation time for functions such as determining the stop and start times of a stopwatch. By default, the value used at run time is the path to the creationTime attribute on the event. The global instance ID is used to relate a problem determination message to a specific event when there is an error at run time which you can see listed in the WebSphere Application Server log.

To open the editor for an item in the monitoring context, click the item in the navigation tree.

A monitoring context can contain monitoring elements such as triggers, inbound events, outbound events, event groups, metrics, keys, counters, stopwatches and other child monitoring contexts. To create them, right click the monitoring context in the tree then select 'New'.

Event sequencing

- Optionally specify event sequence path for the monitoring context or inbound event to indicate sequencing order for inbound events
- Path specified at the monitoring context level will apply to all descendent inbound events unless they override this

The screenshot displays the 'Monitor Details Model' interface. On the left, a tree view shows a hierarchy of monitoring contexts under 'Order Handling', including 'Order Handling', 'Cancel Order and Send', 'Check Customer Accou', 'Check Order Handling F', 'Order HandlingReceive', 'Review Order', 'Ship Order to Customel', 'Update Order Databas', 'Order Handling Instanc', and 'Acceptable Credit Risk?'. The 'Order Handling' context is selected. On the right, the 'Quality of Service' section is expanded, showing the 'Event Sequence Path' configuration. The text below the section reads: 'Specify the location of the attribute in the inbound events in this monitoring context that indicates the order in which inbound events should be processed. If there is no event sequence path specified, events will be handled in the order they arrive at the Monitor server.' The 'Event Sequence Path' field contains the following XPath expression: `fn:concat(fn:substring('00000000000000000000', fn:string-length(xs:string(*BPELDData/bpc:eventProgressCounter))), *BPELDData/bpc:eventProgressCounter, ':', fn:substring('00000000000000000000', fn:string-length(xs:string(*BPELDData/bpc:eventLocalCounter))), *BPELDData/bpc:eventLocalCounter)`. The interface also shows a '13' in the bottom left, 'Monitor models' in the bottom center, and '© 2010 IBM Corporation' in the bottom right.

The event sequence identifier is used to enable event reordering, so an event stream which is out of sequence can be re-ordered before being processed by the monitor model.

This is an optional field and you can specify it at the monitoring context level or inbound event level to indicate sequencing order for inbound events.

When an event sequence path is specified at the inbound event level, it takes precedence over any event sequence paths specified in parent monitoring contexts.

Monitoring context

- A monitoring context must contain at least one key
- An item defined within a context cannot be directly referenced by another peer context
- A context is the definition of what an instance of this context contains at runtime
- An instance of a context must be started and it must terminate.
- A monitoring context can contain other monitoring contexts in a hierarchy
- Expressions can reference only data in the monitoring context from which their evaluation is triggered, or its ancestors.
- To pass data to subordinate monitoring contexts, use outbound events

Any given monitoring context (MC) must contain at least one key which uniquely identifies the monitoring context instance and can be used to correlate inbound events to MC instances. You cannot reference items in one monitoring context from a peer monitoring context. In the monitor model editor, you create an MC definition which is the definition of what all instances of this context will contain at runtime. You are responsible for defining creation and termination criteria for every MC, and this is typically done using inbound events that carry creation and termination semantics.

Monitoring context definitions can be nested to form hierarchies. For example, you might want to track individual task instances within a given process instance. If a particular task can run more than once within a given process instance, you can monitor the task using a nested monitoring context to track each instance of the task separately. Expressions can reference only fields in the monitoring context from which their evaluation is triggered, or its ancestors. To pass data to subordinate monitoring contexts, use outbound events that are emitted in a higher-level monitoring context and received in a lower-level context.

Counters

- Use sorting to create an index on the schema for dashboard performance
- Specify trigger or inbound event
- Specify resulting action
 - Add one
 - Subtract one
 - Reset to zero

Counter Details
Edit the details of the counter, which counts the number of occurrences of some situation or event.

ID:

Name:

Description:

This counter can be used for sorting

Counter Controls
Specify what causes the counter to change and what action is taken.

Trigger / Inbound Event	Resulting Action
<input checked="" type="checkbox"/> Shipped Order Trigger	Add One

On this slide is an example of a counter. You can select the check box so that the counter is used for sorting, which means an index is created on the schema for improved performance.

In 'Counter Controls', you define the trigger or inbound event that triggers the resulting action, which will add one, subtract one or reset the counter.

Stopwatches

- Use accumulating to sum multiple pairs of start/stop
- Specify trigger or inbound event
- Specify resulting action – start/stop use event creation time not Monitor Server system time
 - Start
 - Stop
 - Reset

▼ Stopwatch Details
 Edit the details of the stopwatch, which keeps track of elapsed time. If an accumulating stopwatch is given two or more start and stop times, the durations are added together to produce an accumulated time.

ID:

Name:

Description:

This stopwatch is an accumulating stopwatch

This stopwatch can be used for sorting

▼ Stopwatch Controls
 Specify what causes the stopwatch to change and what action is taken.

Trigger / Inbound Event	Resulting Action
<input type="checkbox"/> Cancel Order and Send NotificationSKIPPED	Start
<input type="checkbox"/> Cancel Order and Send NotificationENTRY	Start
<input type="checkbox"/> Cancel Order and Send NotificationEXIT	Stop
<input type="checkbox"/> Cancel Order and Send NotificationTERMINATED	Stop
<input type="checkbox"/> Cancel Order and Send NotificationSTOPPED	Stop
<input type="checkbox"/> Cancel Order and Send NotificationFAILED	Stop
<input type="checkbox"/> Cancel Order and Send NotificationEXPIRED	Stop
<input type="checkbox"/> Cancel Order and Send Notification#COMPLETED	Stop

On this slide is an example of a stopwatch. An accumulating stopwatch indicates that the specified trigger actions result in accumulation of duration times. So, if you have two pairs of start and stop actions, the duration times from both pairs are added together to produce an accumulated time.

In 'Stopwatch Controls', you specify the trigger or inbound events that cause the resulting actions, which can be start, stop or reset. The start and stop functions use the creation time on the inbound event, not the server system time.

Metrics

- Type – boolean, date, dateTime, decimal, duration, integer, string, time
- Default value
- Metric value expressions
 - Trigger
 - Expression

▼ Metric Details
Edit the details of the metric, which is a holding spot for information used in other calculations.

ID:

Name:

Description:

Type:

Maximum String Length:

Allocate additional space in database to accommodate Unicode string for globalization

A value is required for this metric

Default Value:

This metric can be used for sorting

▼ Metric Value Expressions
Specify the expressions that set the value of the metric. If a trigger is specified, the expression is evaluated when the trigger fires.

Trigger	Expression
<input type="checkbox"/> Cancelled Order Trigger	'Cancelled'
<input type="checkbox"/> Shipped Order Trigger	'Shipped'

Here is a screen capture showing a metric in the monitor model editor.

A metric can be of type boolean, date, dateTime, decimal, duration, integer, string or time. You can specify a default value that is set when the monitoring context instance is created and it can be a literal or a function. The 'Metric Value Expressions' identify the triggers that set the value of the metric based on the expression that you specify. The trigger is optional and if you do not specify it, then the expression evaluates whenever any of its inputs change. So, the expression can reference an inbound event, metric or counter, and when these change in value then the metric expression can be evaluated at that time.

Keys

- Type – boolean, date, dateTime, decimal, duration, integer, string, time
- Default value
 - Literal or function
- Key value expressions

Key Details
Edit the details of the key. Each monitoring context requires at least one key.

ID:

Name:

Description:

Type:

Maximum String Length:

Allocate additional space in database to accommodate Unicode string for globalization

A value is required for this key

Default Value:

This key can be used for sorting

Key Value Expressions
Specify the expressions that set the value of the key.

Expression	
<input type="text" value="New_Order_Event/extendedData/OrderBOData/orderNumber"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

On this slide you see a screen capture of a key in the monitor model editor. The type can be boolean, date, dateTime, decimal, duration, integer, string or time. In 'Key Value Expressions' you specify the expression that typically sets the value of the key from some payload information on an inbound event.

Triggers

- Repeatabe means trigger it every time condition evaluates to true
- Terminate monitoring context
- Trigger source
 - Recurring wait time
 - Trigger
 - Value change on key, metric or counter
 - Inbound event
- Trigger condition

▼ Trigger Details
Edit the details of the trigger, which detects an occurrence and initiates an action in response.

ID:

Name:

Description:

Trigger is repeatable

Terminate monitoring context

▼ Trigger Sources
Specify the source of this trigger.

Source Type	Source
Event	Activity Event

▼ Trigger Condition
Specify the condition that determines whether the trigger will fire.

This slide shows a screen capture of a trigger. You can set the trigger to be repeatable. Repeatable means that the trigger is fired beginning when the condition first resolves to 'true' and every time after that when the trigger source is received and the condition remains 'true'. Non-repeatable means that the trigger is fired beginning when the condition first resolves to 'true', and will not fire again unless the condition first resolves to 'false' and then to 'true' again.

You can specify that the trigger causes the monitoring context to terminate.

The trigger source can be a recurring wait time, another trigger, an inbound event, or a value change on a key, metric or counter. You can also specify a condition to further refine when the trigger source will cause the trigger to fire.

Inbound event

- Type is a pre-existing event definition
- Optional filter condition to restrict event set
- Optional correlation expression based on the specified type
- Instance matching behavior

Event Type Details

Specify the event type or the XML schemas that together describe the structure of this inbound event. You can specify an extension name, event parts, or both.

Extension name:

Event parts:

ID	Name	Type	Path

Filter Condition

Define a condition based on the event attributes to identify whether to accept an event of this type.

Correlation Expression

Define an expression to identify the monitoring context instance or instances that receive the event at runtime.

If no instances are found:

If one instance is found:

If multiple instances are found:

This shows a screen capture of an inbound event. You specify the event type which has already been defined in the project. Filter condition is optional and specifies criteria used to restrict the event set. The correlation expression identifies the MC instances that should be matched based on the key. On the page you also specify what to do if no instances are found, or if one instance is found, or if multiple instances are found.

Outbound event

- Type is a pre-existing event definition
- Event attribute details
 - Specify trigger
 - Assign values to event attributes
 - literal, event data, key, metric, counter, stopwatch, functions
 - Use BusinessSituationName on event for alerts
- Optional filter

Event Type Details
Specify the event type or the XML schemas that together describe the structure of this outbound event. You can specify an extension name, event parts, or both.

Extension name: DeclinedOrderEvent

Event parts:

ID	Name	Type	Path

Outbound Event Content
Specify the triggers that cause the event to be sent. Use the Expression column to specify the value for each event attribute when the event is sent.

Name	Type	Expression
Declined Order Trigger		
Property Data		
Extended Data		
BusinessSituationName	string	Too many orders have been declined

Filter Condition
Define a condition to determine whether to send the event.

This is a screen capture of an outbound event in the monitor model editor. Type is a pre-existing event definition that has been created in the project. In outbound event content, you specify the trigger that determines when this event is emitted, and you also specify expressions to set the values on the event attributes. For business situation events, you will need to use an extended data element called BusinessSituationName in the common base event (CBE) style event. Optionally, you can specify a filter to determine whether to send the event.

Event groups

- Event groups act as a light weight container for events
- Allows for the logical grouping of events
- Event groups have no representation or impact at runtime
- Event groups cannot be nested
- Inbound events within event groups should contain the same correlation expressions as if they resided directly in the containing monitoring context

Create New Event Group

Create an event group

Specify the event group name and the events to include in the group.

Name:

Select the inbound events to place in this event group:

- Order HandlingENTRY
- Order HandlingRESTARTED
- Order HandlingTERMINATED
- Order HandlingDELETED
- Order HandlingFAILED
- Order HandlingCOMPENSATED
- Order HandlingCOMPFAILED
- OrderHandling_FlowSKIPPED
- OrderHandling_Flow Order HandlingReceive to Check Order Handling F
- OrderHandling_Flow Check Order Handling Policy for Automatic Approv

Event groups are containers which are used to store your inbound events. This allows you to easily group your events to make your model more visually usable. Event groups are used only in the toolkit for grouping purposes, so they have no representation at runtime. You cannot nest them. Also, inbound events which are stored in an event group, should be setup with the same correlation information as if they were stored directly in the containing monitoring context.

In the pop-up menu, you can select to create a new event group. When you choose to create a new event group, the event group is created within the parent monitoring context. Then the dialogue is displayed, allowing you to specify the event group name and select from the events available for inclusion in the event group. All events in the containing monitoring context that are not already included in other event groups are listed.

Handling array elements in inbound event XML

- A metric cannot be setup as an array
- Option 1
 - Use standard XPath functions on the incoming event to strip all the elements and concatenate into a string metric
 - See topic 'XPath functions and operators' in the information center
- Option 2
 - Create a new monitoring context for each element in the array using function send-events
 - See topic 'Extended XPath 2.0 support'

State History	State History	1	Total
2	2009-12-08_16:34:53.532_GMT-05:00	2 m, 3	
	FINISHED		
	2009-12-08_16:33:07.657_GMT-05:00		
	CLAIMED by admin		
	2009-12-08_16:32:50.125_GMT-05:00		
	READY		
	2009-12-08_16:32:49.938_GMT-05:00		
	INACTIVE		

Use
 tag for formatting

Global Process Monitor (Advanced)		
Process	Process Statistics	Tasks
Instances		
Export		
Search for: [] Reset		
Time	Time (ps format)	State
December 8, 2009 4:32:49 PM	2009-12-08_16:32:49.938_GMT-05:00	INACTIVE
December 8, 2009 4:32:50 PM	2009-12-08_16:32:50.125_GMT-05:00	READY
December 8, 2009 4:33:07 PM	2009-12-08_16:33:07.657_GMT-05:00	CLAIMED by admin
December 8, 2009 4:34:53 PM	2009-12-08_16:34:53.532_GMT-05:00	FINISHED

To handle array elements in an inbound event, you have two options. You can use standard XPath functions to access the array elements, and then create a single string with all these array values and insert that string into a single string metric. In this example the state history metric has a separate line for each array value. To create individual lines, you can place an HTML BR tag after each item when you are building the concatenated string. For more information, see the information center topic 'XPath functions and operators'.

The second option handles each array element as a child monitoring context. In this case you would drill down in the dashboard to see each array element as a separate child monitoring context. You can use the send-events function to accomplish this. For more information, see the information center topic 'Extended XPath 2.0 support'.

Summary

- Covered monitor models and the monitor model editor in WebSphere Business Monitor

In summary, this presentation covered an overview of monitor models and the monitor model editor in WebSphere Business Monitor.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

[mailto:iea@us.ibm.com?subject=Feedback about WBPM Monitor MonitorModels.ppt](mailto:iea@us.ibm.com?subject=Feedback%20about%20WBPM%20Monitor%20MonitorModels.ppt)

This module is also available in PDF format at: [../WBPM_Monitor_MonitorModels.pdf](..\\WBPM_Monitor_MonitorModels.pdf)

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the Web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. in the United States, other countries, or both.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2010. All rights reserved.