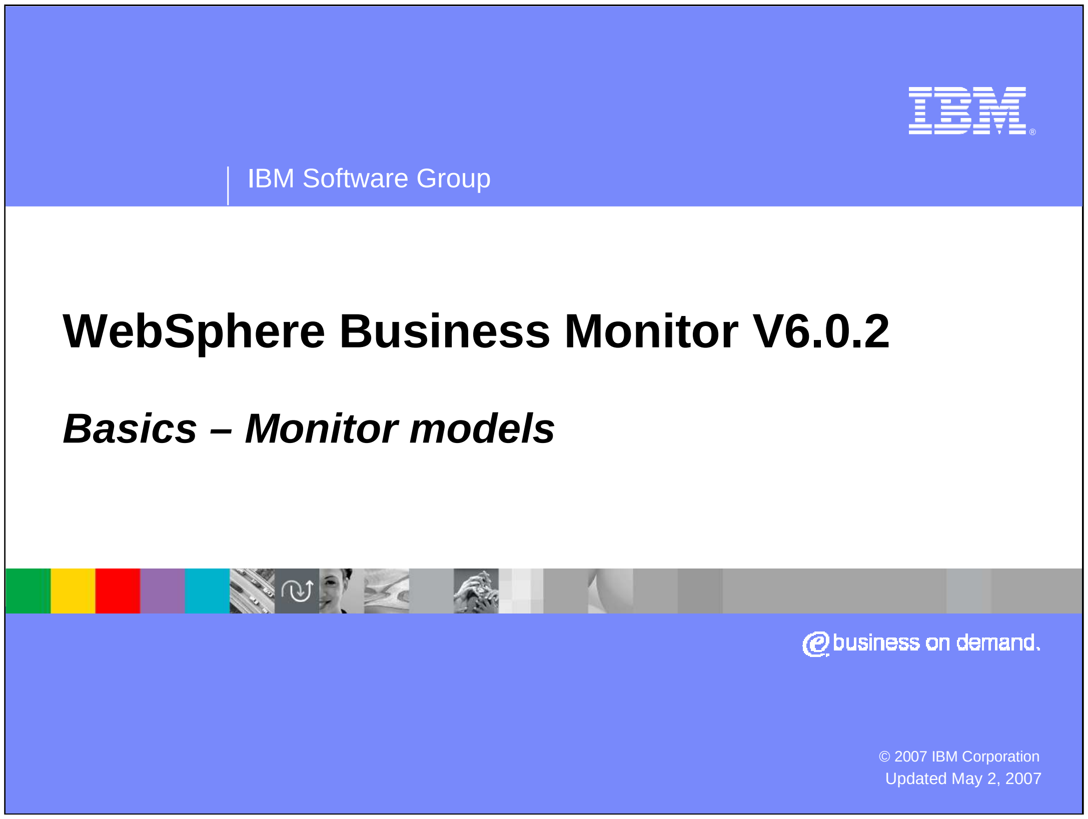


IBM Software Group

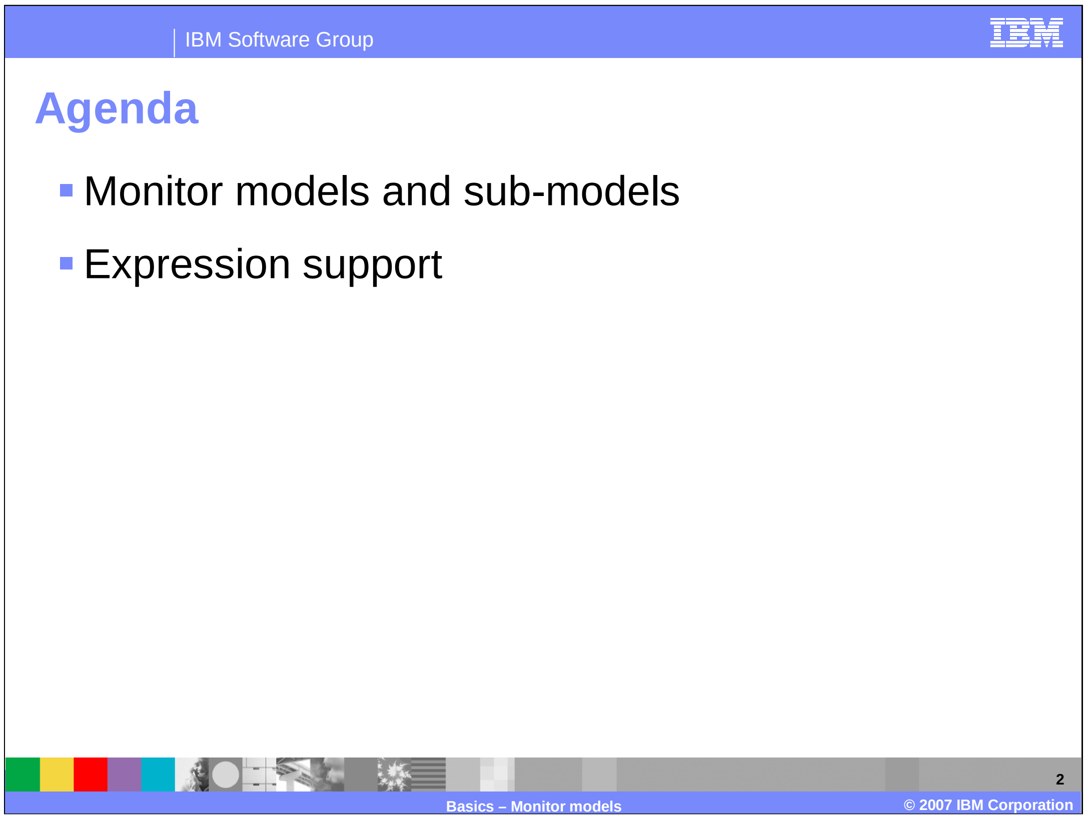
WebSphere Business Monitor V6.0.2

Basics – Monitor models

business on demand.

Updated May 2, 2007

This presentation should give you a good understanding of the Monitor model concept.



IBM Software Group

Agenda

- Monitor models and sub-models
- Expression support

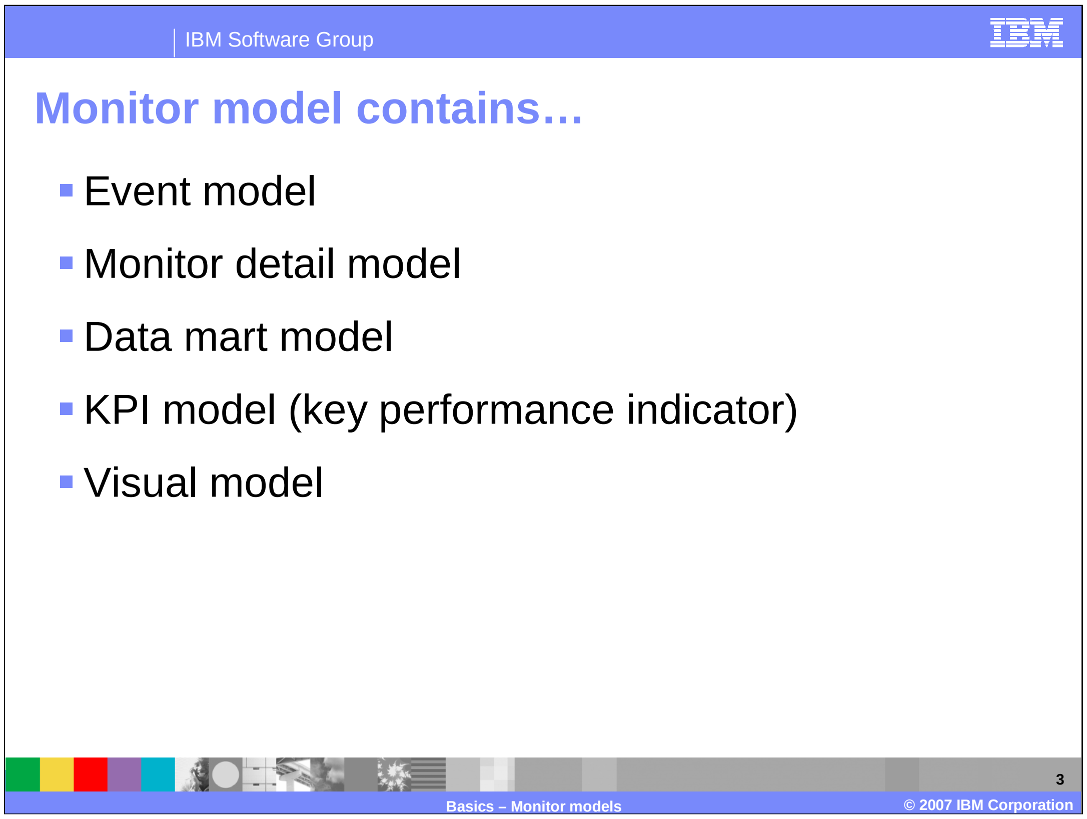
2

Basics – Monitor models

© 2007 IBM Corporation

This is the agenda for the presentation.

Monitor models will be described in detail, including the various sub-models with the monitor model. Also you will review the use of expressions in building conditions and filter criteria in your models.


IBM Software Group

Monitor model contains…

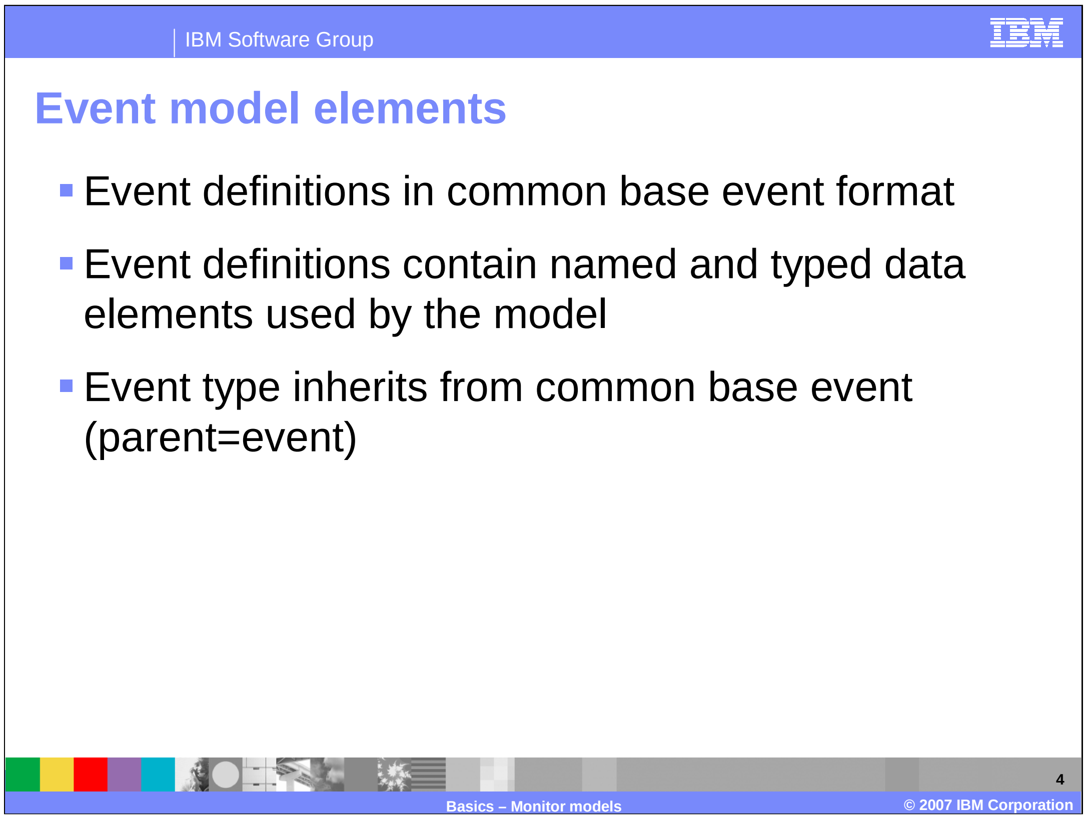
- Event model
- Monitor detail model
- Data mart model
- KPI model (key performance indicator)
- Visual model

3

Basics – Monitor models

© 2007 IBM Corporation

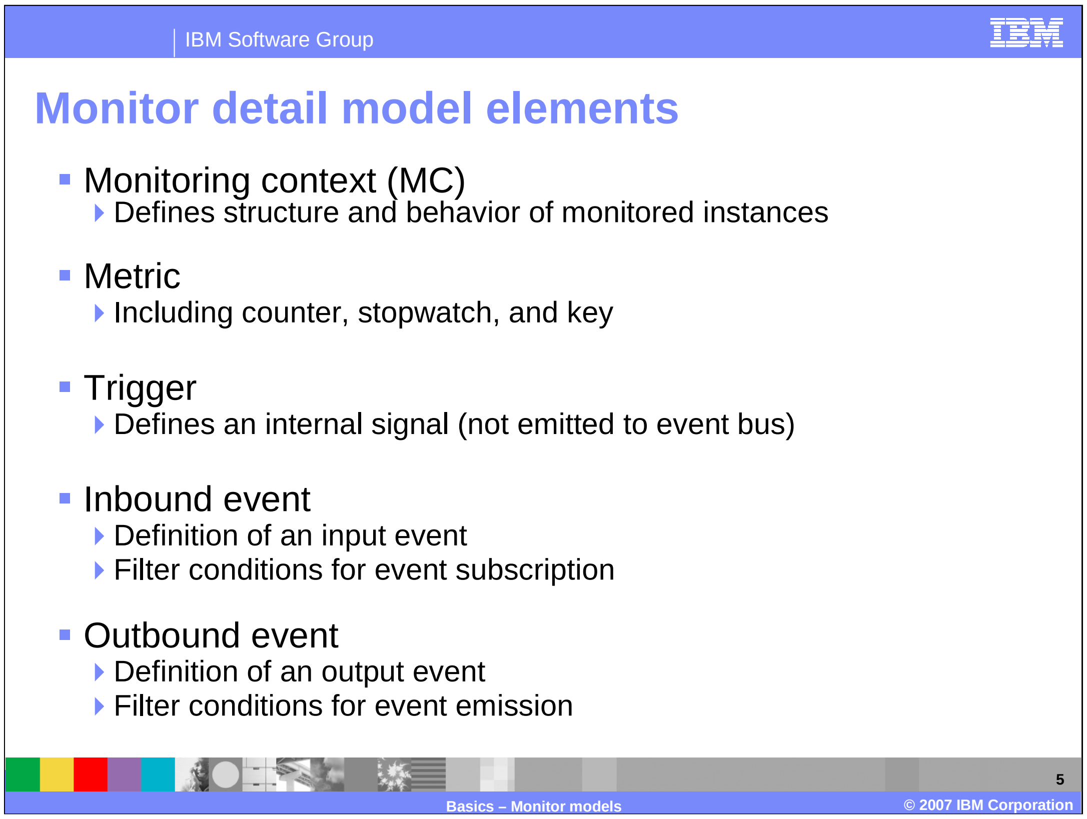
A monitor model is composed of several sub-models, the event model, monitor detail model, data mart model, KPI model, and visual model. The event model defines the common base events that are subscribed to or emitted from the model. The monitor detail model contains the definition of the metrics in the model. The data mart model contains the cubes, facts and aggregated measures for multi-dimensional analysis. The KPI model contains the key performance indicators and associated elements. And the visual model contains the annotation of the SVG diagrams for the monitoring contexts and KPI contexts.



Event model elements

- Event definitions in common base event format
- Event definitions contain named and typed data elements used by the model
- Event type inherits from common base event (parent=event)

The event model contains definitions of common base events that are used by the model. The Common Base Events are in a standard format and contain the business payload that you will be using to calculated metrics in the model. The event types that you define using the toolkit event editor will automatically inherit from the parent Common Base Event type, so you automatically inherit certain standard properties such as extensionName, creationTime and more.



Monitor detail model elements

- Monitoring context (MC)
  - Defines structure and behavior of monitored instances
- Metric
  - Including counter, stopwatch, and key
- Trigger
  - Defines an internal signal (not emitted to event bus)
- Inbound event
  - Definition of an input event
  - Filter conditions for event subscription
- Outbound event
  - Definition of an output event
  - Filter conditions for event emission

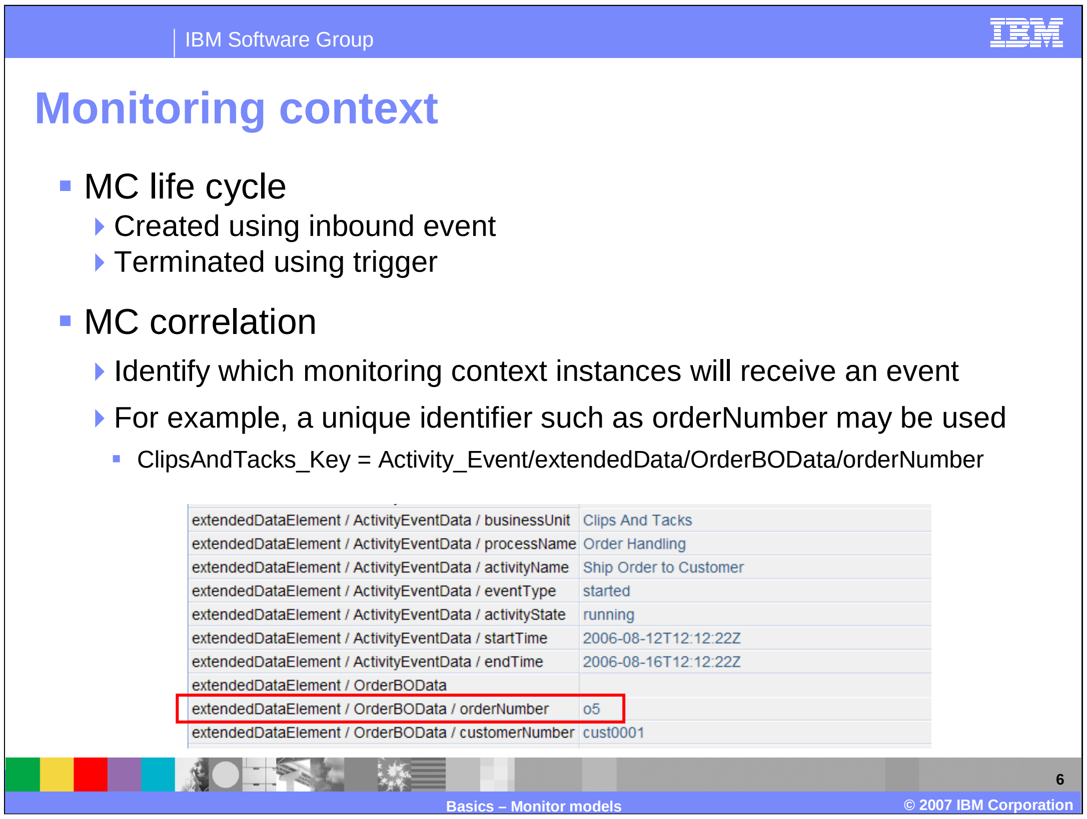
You can create one or more monitoring contexts (MCs) per model. The MC is a container for your metrics.

A metric is a business measure such as a counter, stopwatch or key value.

A trigger is used to fire calculation of the values of your metrics and to fire the emissions of events from the model.

An inbound event is based on a Common Base Event type that you have defined in the model, and it can contain conditions and filter criteria for determining if the event will be processed.

An outbound event is also based on a Common Base Event type, and it can contain conditions and filter criteria to restrict the set of events that are emitted.



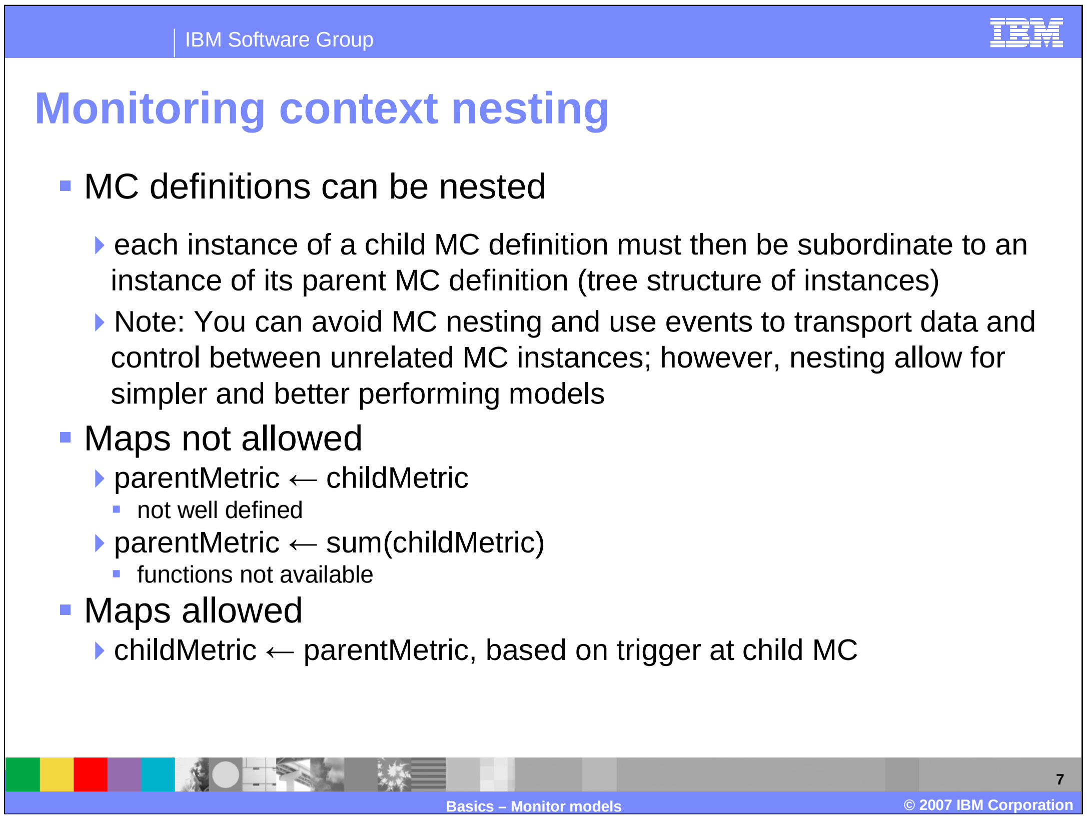
Monitoring context

- MC life cycle
  - Created using inbound event
  - Terminated using trigger
- MC correlation
  - Identify which monitoring context instances will receive an event
  - For example, a unique identifier such as orderNumber may be used
    - ClipsAndTacks_Key = Activity_Event/extendedData/OrderBOData/orderNumber

extendedDataElement / ActivityEventData / businessUnit	Clips And Tacks
extendedDataElement / ActivityEventData / processName	Order Handling
extendedDataElement / ActivityEventData / activityName	Ship Order to Customer
extendedDataElement / ActivityEventData / eventType	started
extendedDataElement / ActivityEventData / activityState	running
extendedDataElement / ActivityEventData / startTime	2006-08-12T12:12:22Z
extendedDataElement / ActivityEventData / endTime	2006-08-16T12:12:22Z
extendedDataElement / OrderBOData	
extendedDataElement / OrderBOData / orderNumber	o5
extendedDataElement / OrderBOData / customerNumber	cust0001

A monitoring context has a life cycle that you manage in the monitor model. For each MC, you need to specify which events will create an MC instance, and you must also specify which triggers are used to terminate the MC instance.

In addition, for each monitoring context you need to specify correlation information so that an inbound event will be sent to the correct MC instance. For example, you might use an order number as the unique identifier which is assigned to the MC key, then every MC instance would have a unique order number associated with it. For the sample event shown, it would be sent to the MC instance that is associated with order number ‘o5’.

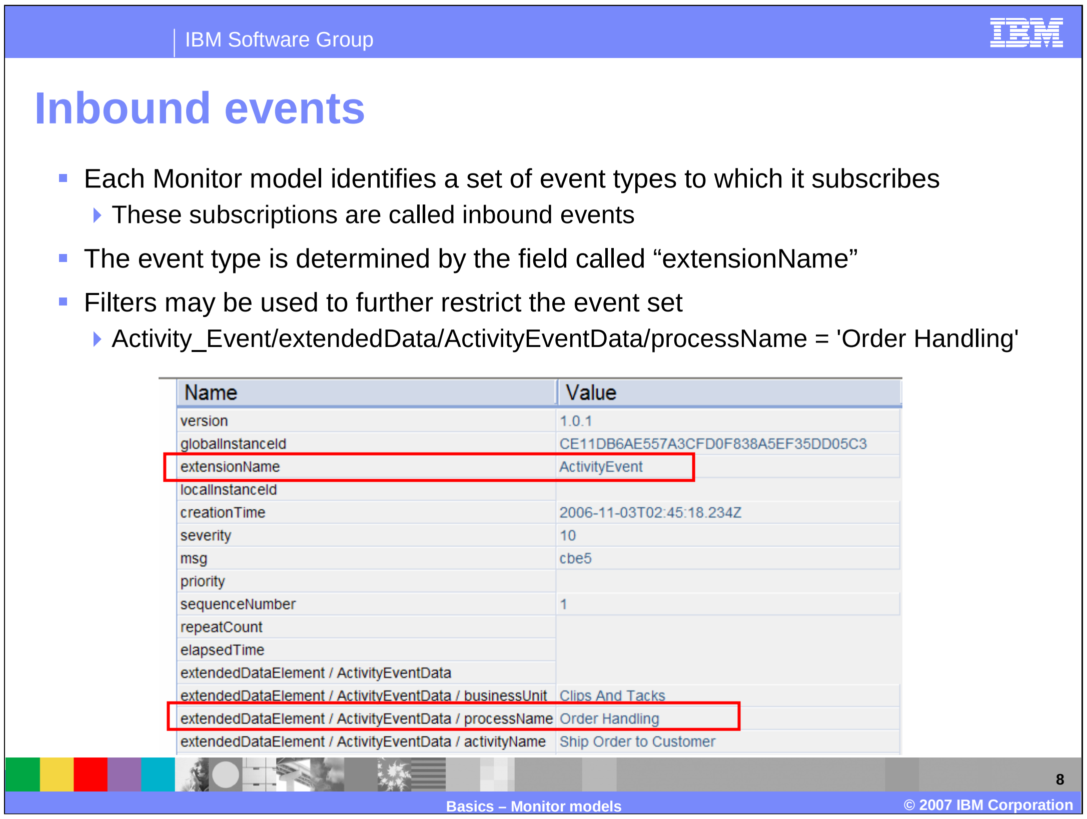


Monitoring context nesting

- MC definitions can be nested
  - each instance of a child MC definition must then be subordinate to an instance of its parent MC definition (tree structure of instances)
  - Note: You can avoid MC nesting and use events to transport data and control between unrelated MC instances; however, nesting allow for simpler and better performing models
- Maps not allowed
  - parentMetric ← childMetric
    - not well defined
  - parentMetric ← sum(childMetric)
    - functions not available
- Maps allowed
  - childMetric ← parentMetric, based on trigger at child MC

Monitor contexts can be nested such that a parent MC could have one or more child MCs, and each child MC could have one or more child MCs, all in a hierarchical arrangement. Each child MC will have its own set of metrics, triggers, and inbound events. It is possible to avoid MC nesting by using events to transmit data between peer MC instances, but this requires additional event processing and may be a performance issue.

When mapping values to metrics, you cannot assign a child metric to a parent metric, because there may be many child MC instances so this is not a well defined construct. In this release, you cannot perfrom functions such as summing of child metrics to assign values to parent metrics. Both of these cases have problems with fan-out, due to having multiple child MC instances for a parent MC. Therefore, you can assign the value of a parent metric to a child metric since this is well defined for any given child MC instance.


IBM Software Group

Inbound events

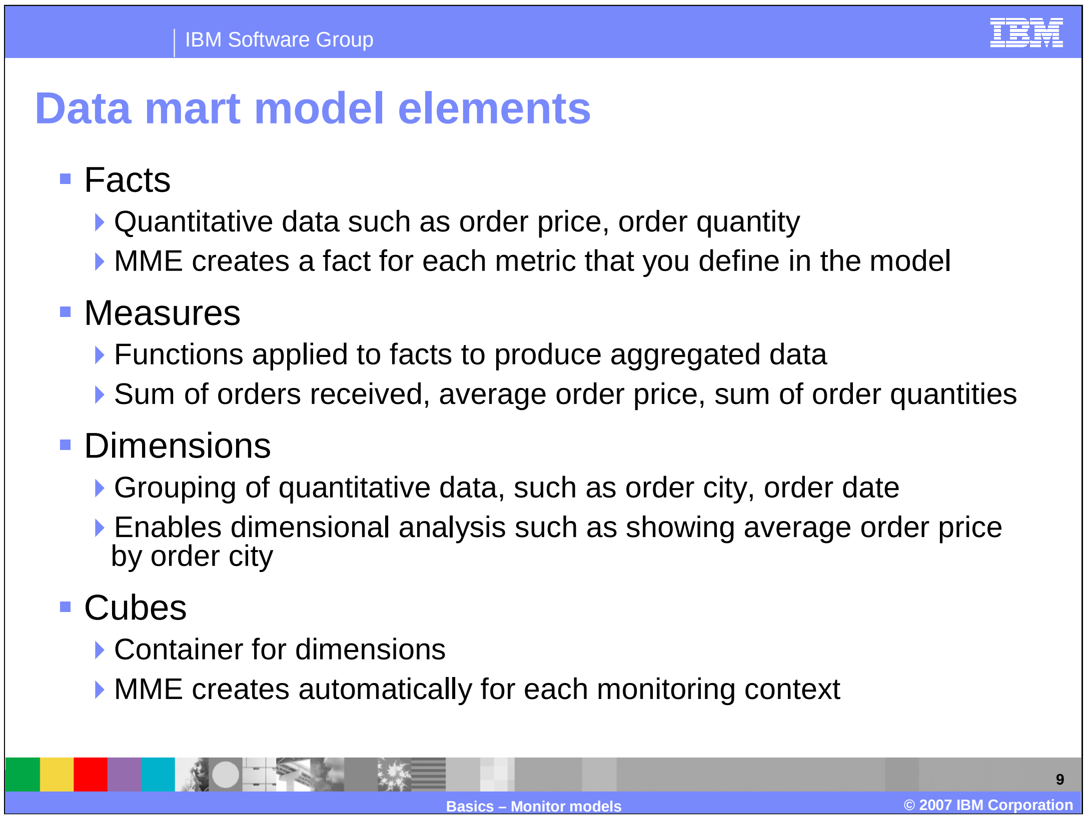
- Each Monitor model identifies a set of event types to which it subscribes
  - These subscriptions are called inbound events
- The event type is determined by the field called "extensionName"
- Filters may be used to further restrict the event set
  - Activity_Event/extendedData/ActivityEventData/processName = 'Order Handling'

Name	Value
version	1.0.1
globalInstanceId	CE11DB6AE557A3CFD0F838A5EF35DD05C3
extensionName	ActivityEvent
localInstanceId	
creationTime	2006-11-03T02:45:18.234Z
severity	10
msg	cbe5
priority	
sequenceNumber	1
repeatCount	
elapsedTime	
extendedDataElement / ActivityEventData	
extendedDataElement / ActivityEventData / businessUnit	Clips And Tacks
extendedDataElement / ActivityEventData / processName	Order Handling
extendedDataElement / ActivityEventData / activityName	Ship Order to Customer

8

Basics – Monitor models © 2007 IBM Corporation

An inbound event is a definition of an event that Monitor will subscribe to, which means that it will look for these events on the CEI server and read them for processing by the monitor model. The inbound event has a type which is defined by the field called extensionName. Filters may also be used to further restrict the set of events to process. In this example, you may want to process event types with an extensionName of ActivityEvent but only where the extendedDataElement called processName is set to 'Order Handling'.

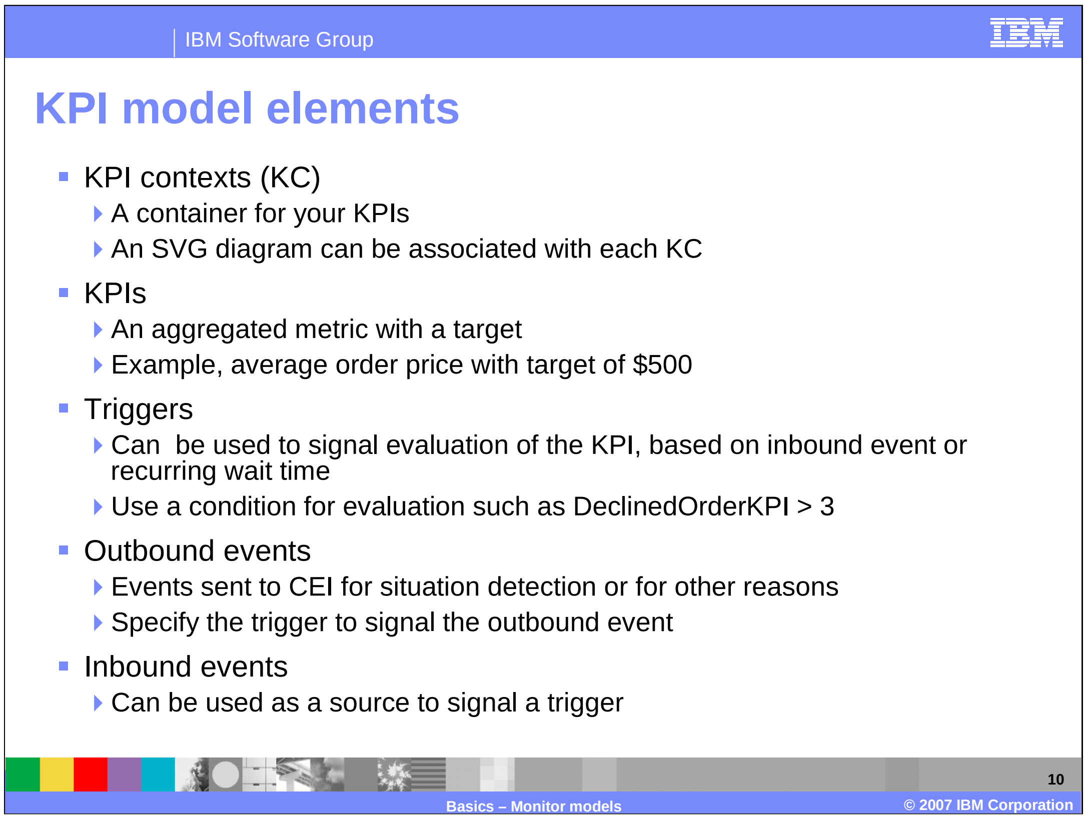


Data mart model elements

- Facts
  - Quantitative data such as order price, order quantity
  - MME creates a fact for each metric that you define in the model
- Measures
  - Functions applied to facts to produce aggregated data
  - Sum of orders received, average order price, sum of order quantities
- Dimensions
  - Grouping of quantitative data, such as order city, order date
  - Enables dimensional analysis such as showing average order price by order city
- Cubes
  - Container for dimensions
  - MME creates automatically for each monitoring context

For multi-dimensional analysis, you need to define a cube which contains dimensions, facts and aggregated measures. In the Monitor Model Editor, a new cube is automatically created for you for every monitoring context that you create. You will create dimensions in the data mart model to represent how you want to slice and dice the data in the cube. For a customer order you might want to define a time dimension and a city dimension. Then you could drill into the aggregated order data based on the month that orders were placed or based on the city that they were placed in.

Facts are the metrics that you define in the model, and measures are the functions applied to those facts to produced aggregated information, for example, the sum of customer order values.



IBM Software Group

KPI model elements

- KPI contexts (KC)
  - A container for your KPIs
  - An SVG diagram can be associated with each KC
- KPIs
  - An aggregated metric with a target
  - Example, average order price with target of $500
- Triggers
  - Can be used to signal evaluation of the KPI, based on inbound event or recurring wait time
  - Use a condition for evaluation such as DeclinedOrderKPI > 3
- Outbound events
  - Events sent to CEI for situation detection or for other reasons
  - Specify the trigger to signal the outbound event
- Inbound events
  - Can be used as a source to signal a trigger

10

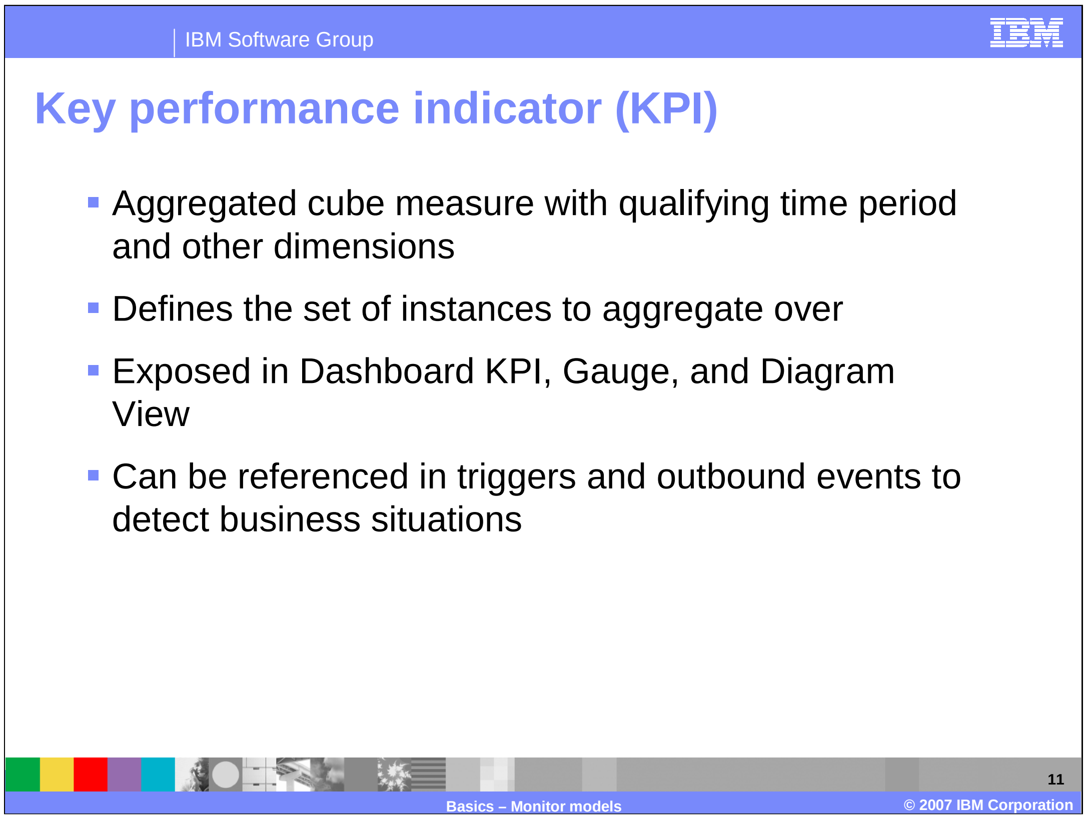
Basics – Monitor models

© 2007 IBM Corporation

A KPI context (KC) is container for your KPIs along with associated inbound events, triggers, and outbound events. You can create one or more KC’s per model.   Note that you can associate a separate SVG diagram in the visual model with each KC and MC in the model.

You can define triggers that can be used to evaluate KPIs, and the triggers can be evaluated based on inbound events or periodic evaluations. You can send outbound events in response to detected KPI situations. For example you can detect situations where KPIs go outside target or range thresholds.

You cannot create metrics, counters, or stopwatches in the KC.



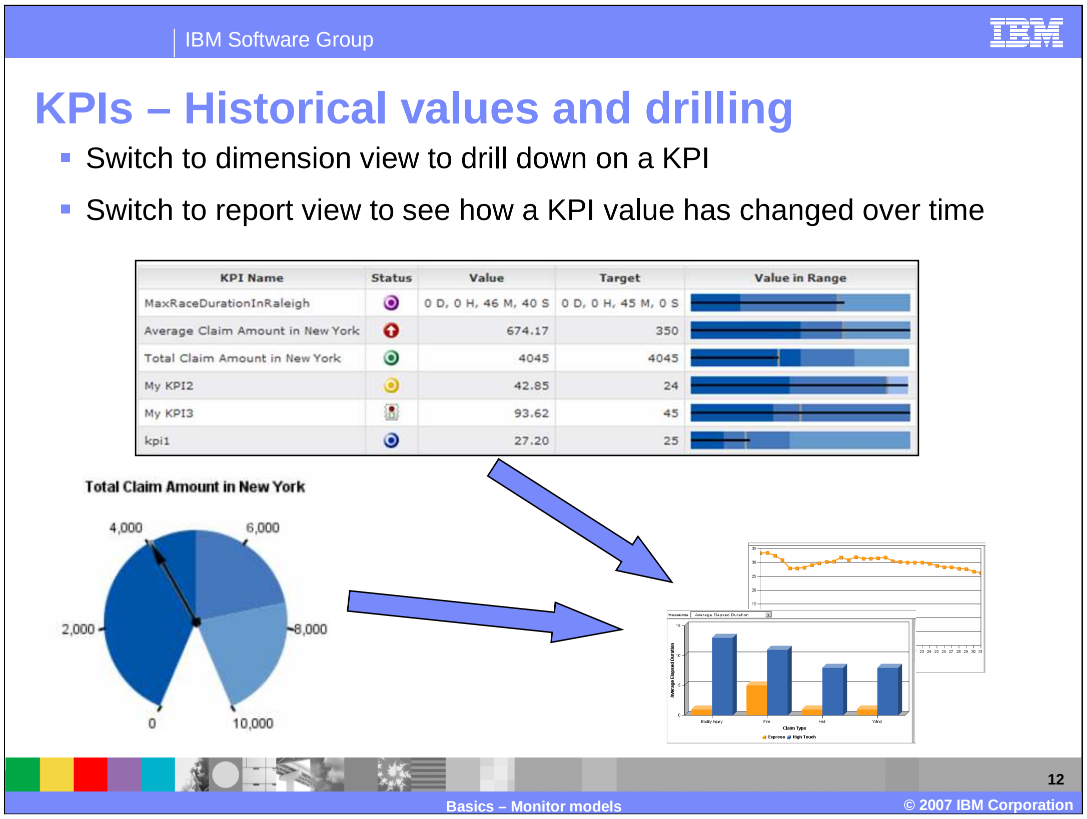
Key performance indicator (KPI)

- Aggregated cube measure with qualifying time period and other dimensions
- Defines the set of instances to aggregate over
- Exposed in Dashboard KPI, Gauge, and Diagram View
- Can be referenced in triggers and outbound events to detect business situations

The key performance indicator (KPI) is an indicator of business performance that is defined in the monitor model and visualized on the dashboards. KPIs are based on cubes so you can qualify them with time periods or any other dimension that you define in the model, then the aggregations are performed based on these dimensions.

KPIs are visualized on the dashboard server in the KPI, Gauge and Diagram portlets.

Using triggers and outbound events, you can also easily generate alerts based on thresholds that you define for the KPI values.



IBM Software Group

KPIs – Historical values and drilling

- Switch to dimension view to drill down on a KPI
- Switch to report view to see how a KPI value has changed over time

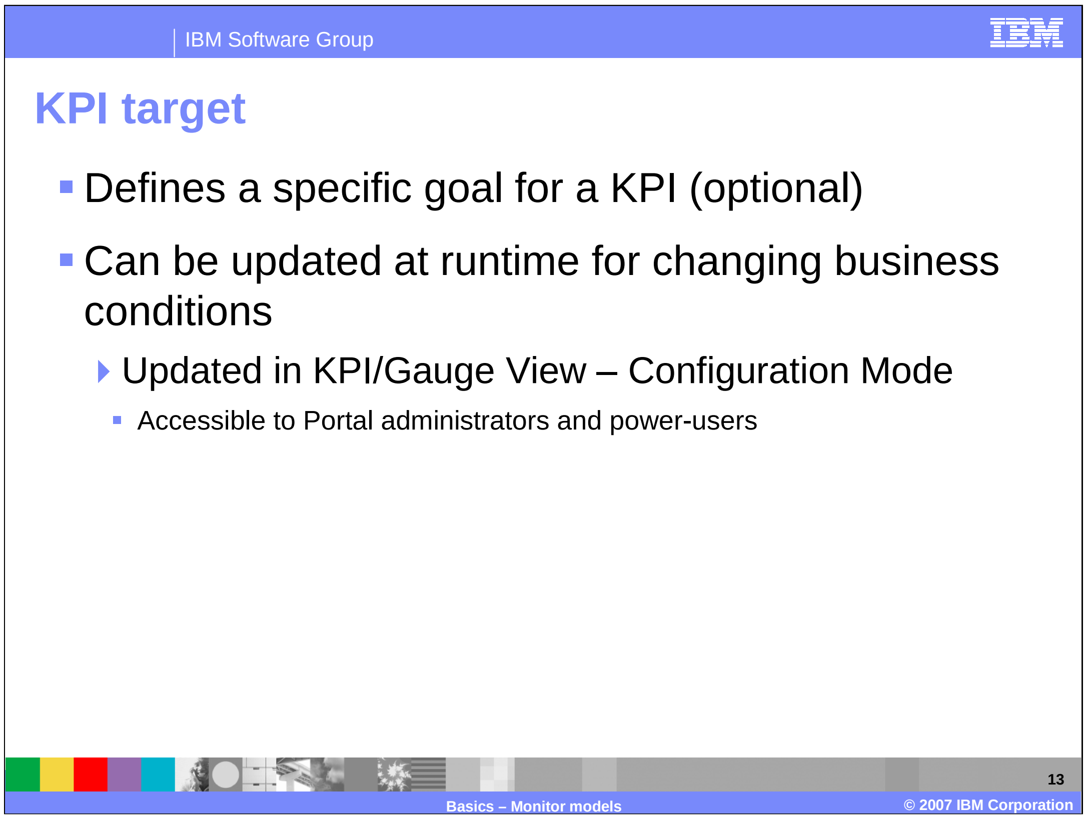
KPI Name	Status	Value	Target	Value in Range
MaxRaceDurationInRaleigh	●	0 D, 0 H, 46 M, 40 S	0 D, 0 H, 45 M, 0 S	
Average Claim Amount in New York	●	674.17	350	
Total Claim Amount in New York	●	4045	4045	
My KPI2	●	42.85	24	
My KPI3	●	93.62	45	
kpi1	●	27.20	25	

12

Basics – Monitor models

© 2007 IBM Corporation

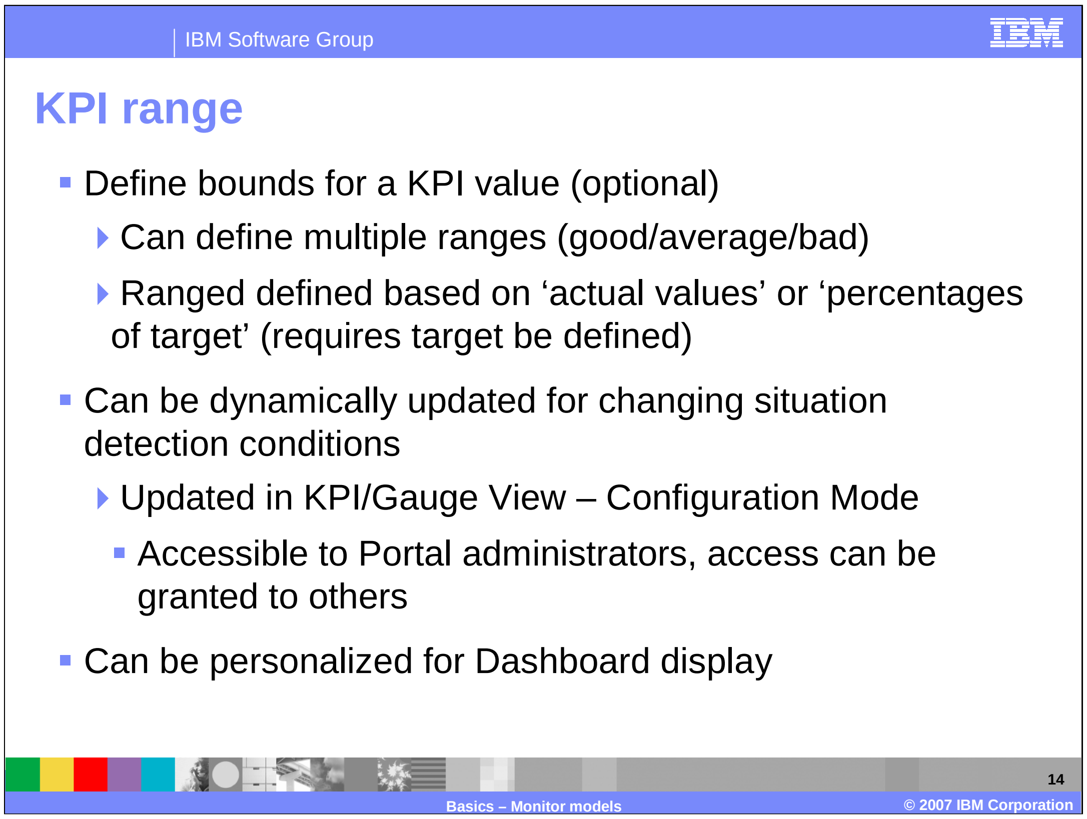
KPIs are visualized in the KPI and Gauge portlets, but since KPIs are based on cubes, the KPIs will show up in the dimension and reports views so that you can perform drill downs based on dimensions. In the report view you can drill down on the KPI based on time periods, and in the dimension view you can drill down on the KPI based on any dimension defined in the model.



KPI target

- Defines a specific goal for a KPI (optional)
- Can be updated at runtime for changing business conditions
  - Updated in KPI/Gauge View – Configuration Mode
    - Accessible to Portal administrators and power-users

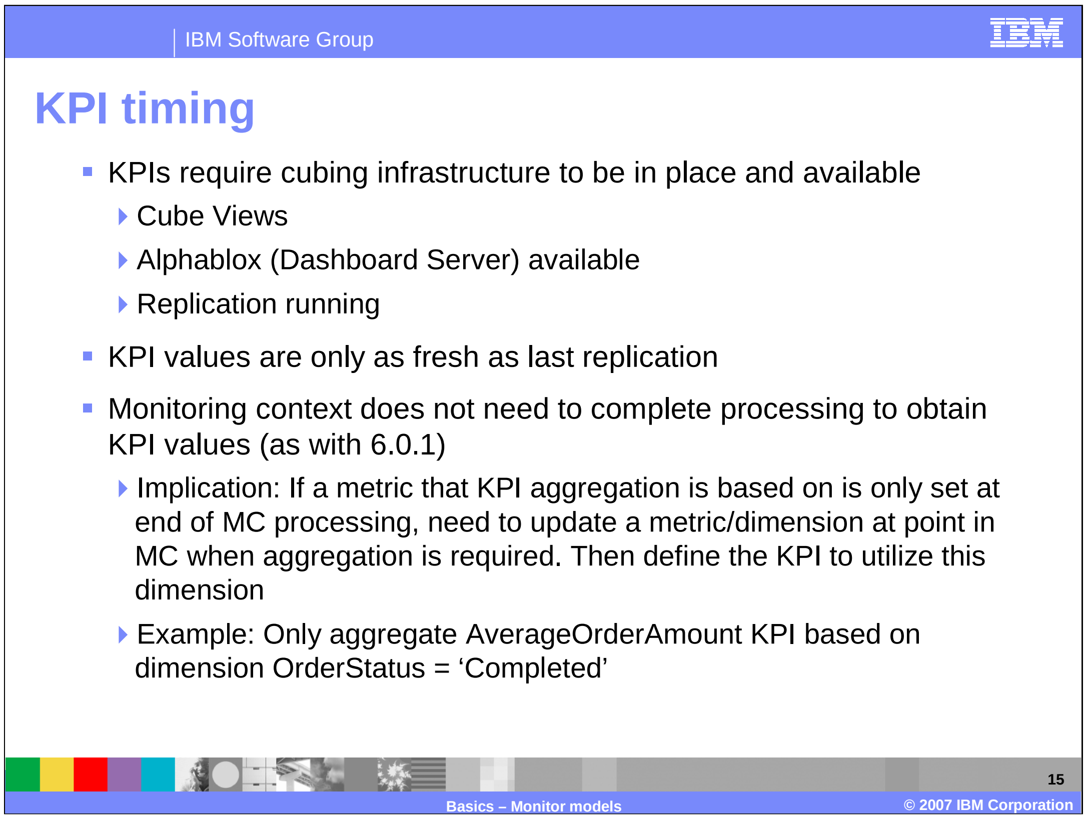
A target can optionally defined for the KPI to show the optimal goal for the KPI. It is visualized on the KPI views so that you can quickly see if the KPI is above or below target. This target can also be updated at runtime if your business needs change. This can be updated by your administrators in both the KPI view and Gauge view in configuration mode.



KPI range

- Define bounds for a KPI value (optional)
  - Can define multiple ranges (good/average/bad)
  - Ranged defined based on ‘actual values’ or ‘percentages of target’ (requires target be defined)
- Can be dynamically updated for changing situation detection conditions
  - Updated in KPI/Gauge View – Configuration Mode
    - Accessible to Portal administrators, access can be granted to others
- Can be personalized for Dashboard display

One or more ranges can optionally be created for a KPI, and they can be defined based on the actual value of the KPI or based on a percentage of the target. Just like the target, your administrators can update the ranges at runtime using the configuration mode of the KPI view and Gauge view. End users can also personalize the ranges for their own use, so then each user can have ranges based on their job roles.

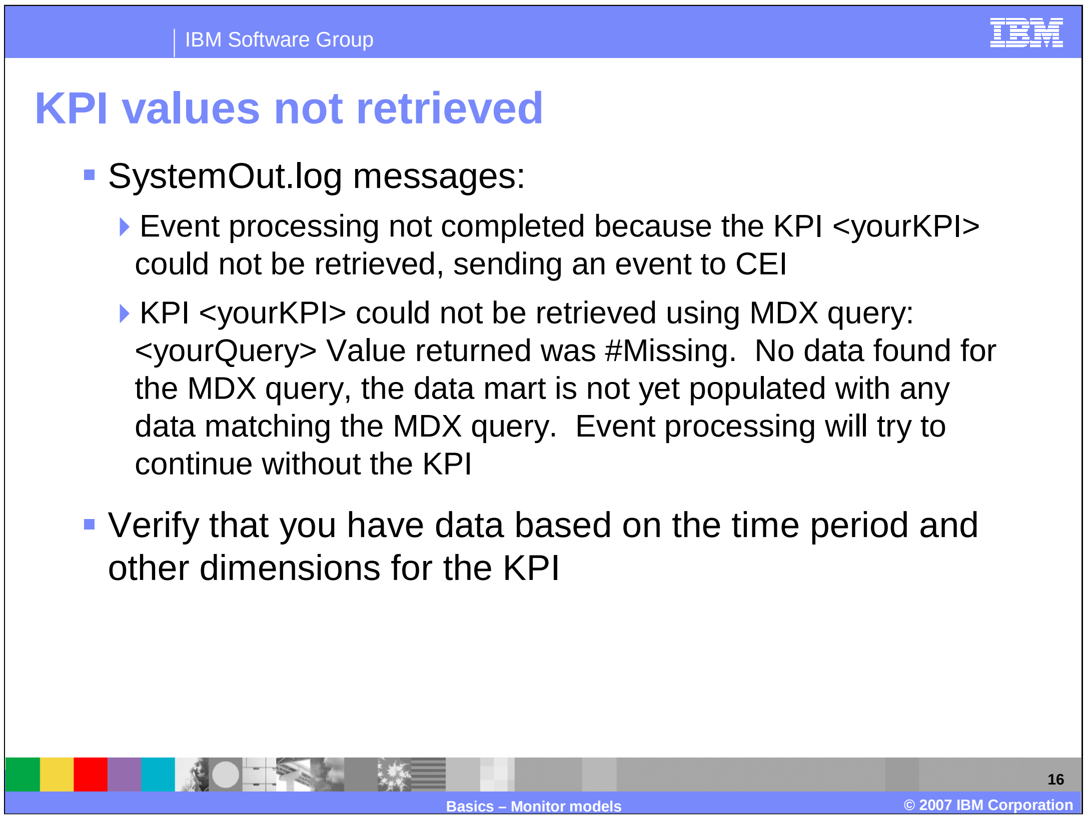


KPI timing

- KPIs require cubing infrastructure to be in place and available
  - Cube Views
  - Alphablox (Dashboard Server) available
  - Replication running
- KPI values are only as fresh as last replication
- Monitoring context does not need to complete processing to obtain KPI values (as with 6.0.1)
  - Implication: If a metric that KPI aggregation is based on is only set at end of MC processing, need to update a metric/dimension at point in MC when aggregation is required. Then define the KPI to utilize this dimension
  - Example: Only aggregate AverageOrderAmount KPI based on dimension OrderStatus = 'Completed'

Since KPIs are based on cubes, then the cubing infrastructure must be in place in Cube Views and in Alphablox. Also, replication must be running to get the information to the Datamart database, so the KPI values are only as fresh as the time of the last replication.

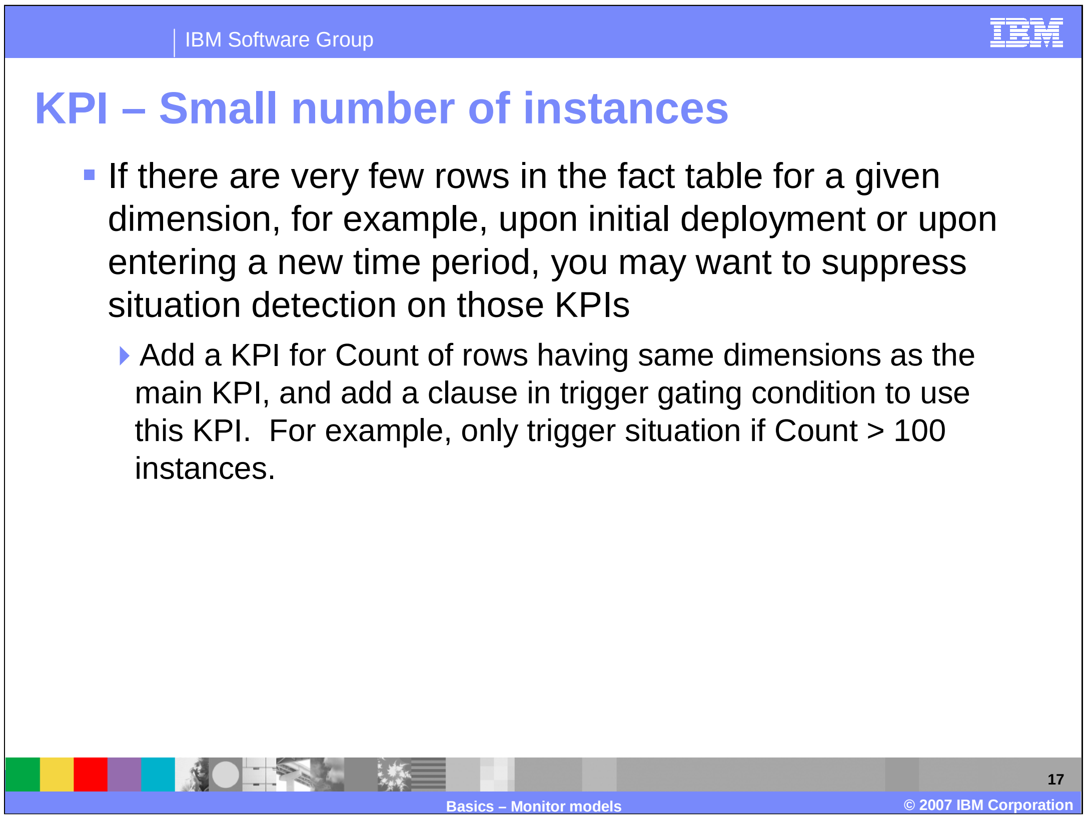
Unlike the previous version, a monitoring context can be in progress and still show current values in the KPIs. But you are in control of this in the model, so the KPI value will be updated whenever you trigger its update in the model. So, for example, you may want to update the KPI for average order amount only for orders that are in completed status. Then new orders and cancelled orders will not be included in the average calculation.



KPI values not retrieved

- SystemOut.log messages:
  - Event processing not completed because the KPI <yourKPI> could not be retrieved, sending an event to CEI
  - KPI <yourKPI> could not be retrieved using MDX query: <yourQuery> Value returned was #Missing. No data found for the MDX query, the data mart is not yet populated with any data matching the MDX query. Event processing will try to continue without the KPI
- Verify that you have data based on the time period and other dimensions for the KPI

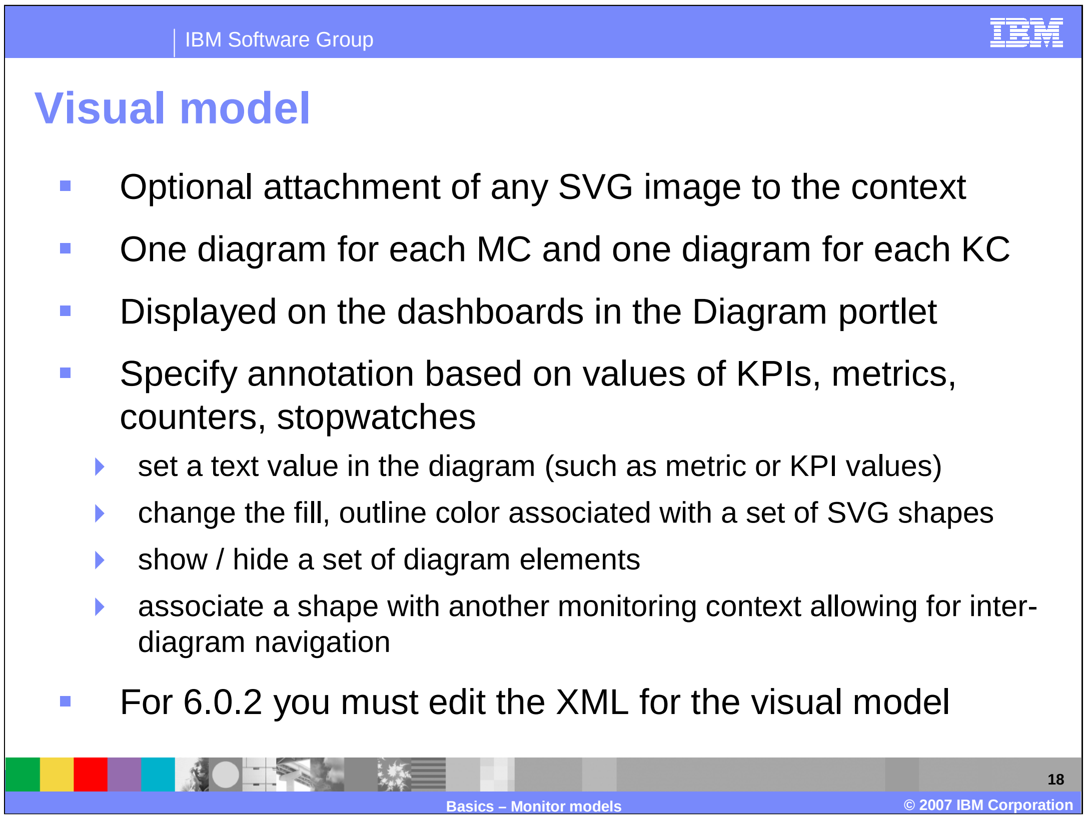
If you are receiving error messages in the log showing that the KPI values are missing, then you may not have any data in the fact table matching the dimensions that are defined for the KPI. So you should verify the time period and other dimension filters that are defined in the KPI.



KPI – Small number of instances

- If there are very few rows in the fact table for a given dimension, for example, upon initial deployment or upon entering a new time period, you may want to suppress situation detection on those KPIs
  - Add a KPI for Count of rows having same dimensions as the main KPI, and add a clause in trigger gating condition to use this KPI. For example, only trigger situation if Count > 100 instances.

If you have a small number of rows in the fact table representing a particular KPI, then the data may not be very meaningful, especially with regards to situation detection. This can happen when you first deploy a model or if a new time period starts for a KPI. To avoid this issue, you can create another KPI using the same dimensions as the original KPI, and this new KPI would count the rows. Then you can add a clause on the outbound event trigger to use the KPI and make sure it is greater than some predefined number.


IBM Software Group

Visual model

- Optional attachment of any SVG image to the context
- One diagram for each MC and one diagram for each KC
- Displayed on the dashboards in the Diagram portlet
- Specify annotation based on values of KPIs, metrics, counters, stopwatches
  - set a text value in the diagram (such as metric or KPI values)
  - change the fill, outline color associated with a set of SVG shapes
  - show / hide a set of diagram elements
  - associate a shape with another monitoring context allowing for inter-diagram navigation
- For 6.0.2 you must edit the XML for the visual model

18

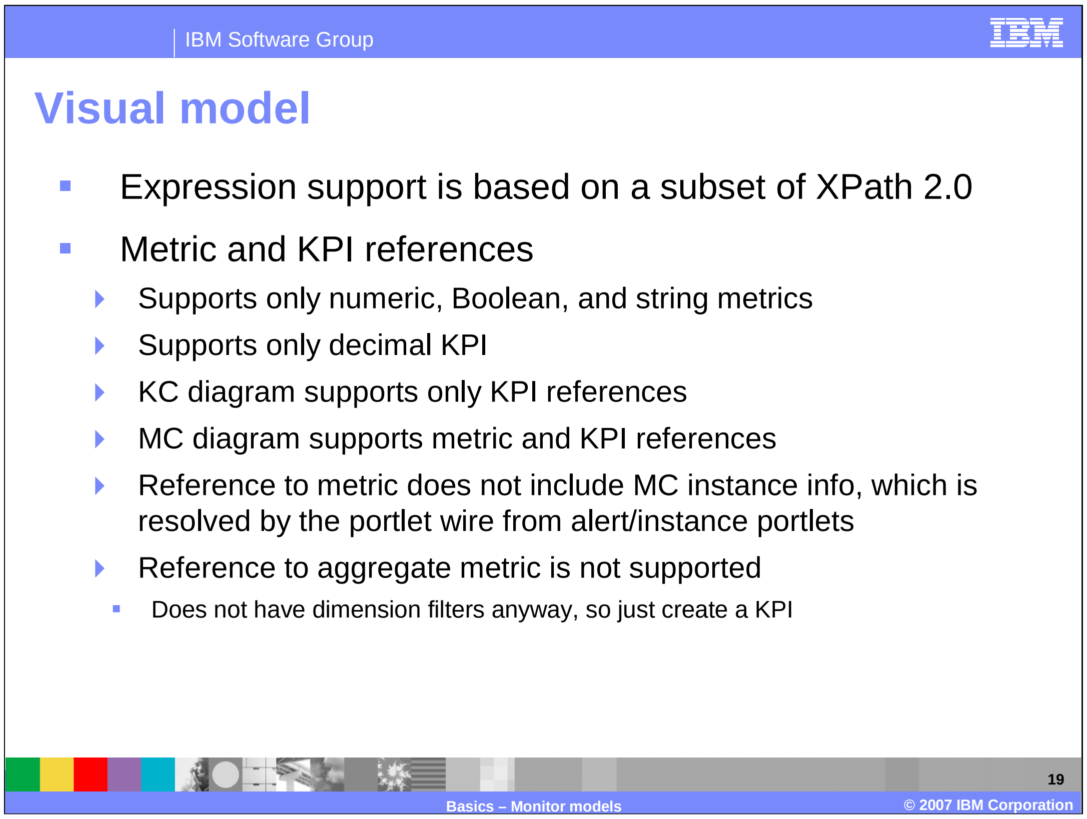
Basics – Monitor models

© 2007 IBM Corporation

The purpose of the visual model definition in the monitor model is to allow you to optionally associate a diagram representation to each context in the monitor model. This applies to both monitoring contexts or a KPI contexts. These diagrams are then visualized on the dashboard in the Diagram portlet.

A nice feature of the visual model is the ability to annotate the diagram so that you can display the values of KPIs, metrics, counters and stopwatches on the diagrams. You can change SVG shape fill colors and outline colors. You can hide or show a set of diagram shapes. And you can allow drill down to other diagrams by creating associations to the other diagrams.

Note that for this release, there aren't any graphical panels to help you define these annotations, so you must code the XML into the monitor model XML file.



Visual model

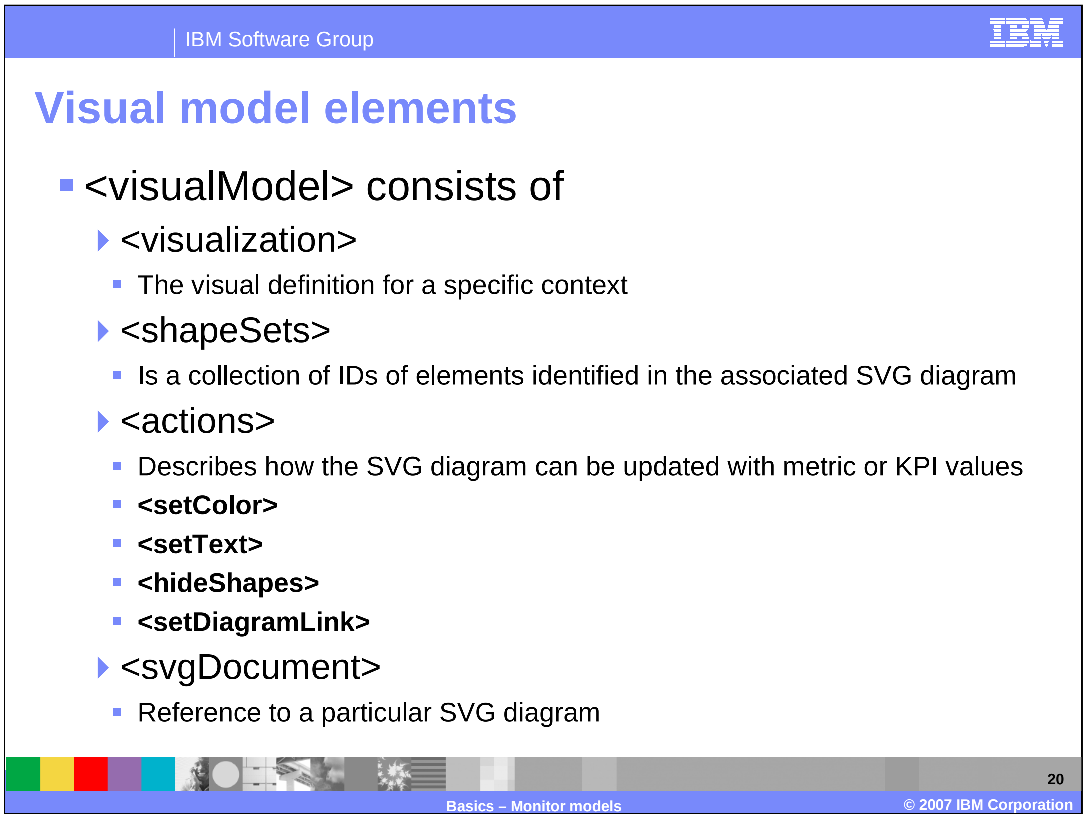
- Expression support is based on a subset of XPath 2.0
- Metric and KPI references
  - Supports only numeric, Boolean, and string metrics
  - Supports only decimal KPI
  - KC diagram supports only KPI references
  - MC diagram supports metric and KPI references
  - Reference to metric does not include MC instance info, which is resolved by the portlet wire from alert/instance portlets
  - Reference to aggregate metric is not supported
    - Does not have dimension filters anyway, so just create a KPI

Expression support in the visual model is based on XPath 2.0.

For references in the visual model to metrics, you can only use numeric, Boolean and string types. For references to KPIs, you can use only decimal types. Note that the KPI context diagram can only have KPI references, but the monitoring context diagram can have references to both metrics and KPIs.

An MC diagram is displayed when using click to action from the alert portlet or instance portlet. So, when referencing a metric, you don't have to specify which MC instance because the instance information is resolved by the wire between the alert or instance portlets and the diagram portlet.

Notice that a reference to instance metrics are supported but references to aggregate metrics are not supported. But this is not a restriction, because you can just create a KPI which is based on the aggregate metric. You wouldn't really want to have a reference to an aggregated metric anyway because they don't have dimension filters defined on them and therefore you would be displaying information for all history.



Visual model elements

- <visualModel> consists of
  - <visualization>
    - The visual definition for a specific context
  - <shapeSets>
    - Is a collection of IDs of elements identified in the associated SVG diagram
  - <actions>
    - Describes how the SVG diagram can be updated with metric or KPI values
    - **<setColor>**
    - **<setText>**
    - **<hideShapes>**
    - **<setDiagramLink>**
  - <svgDocument>
    - Reference to a particular SVG diagram

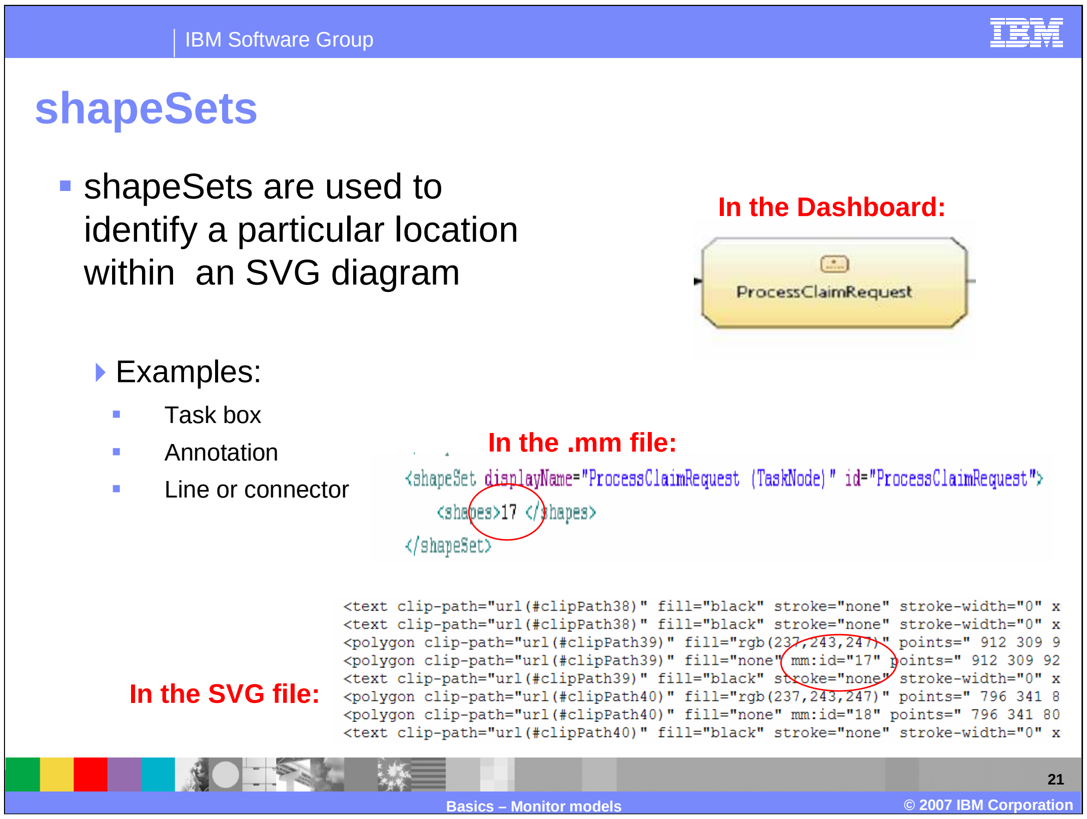
This slide shows the visual model XML tags. The visualModel tag is at the highest level.

The visualization tag is associated with one specific monitoring context or KPI context.

A shapeSet is a reference to an ID a diagram element within the SVG diagram. Think of shapeSets as a way to pinpoint where on the diagram that you want to take actions on.

There are four different actions. setColor changes the color of a shape, setText adds text to the diagram, hideShapes hides a shape, and setDiagramLink provides a link to another diagram.

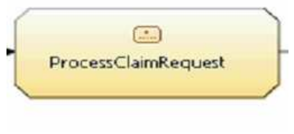
The svgDocument tag associates a particular SVG diagram with this monitoring context or KPI context.



shapeSets

- shapeSets are used to identify a particular location within an SVG diagram
  - Examples:
    - Task box
    - Annotation
    - Line or connector

In the Dashboard:



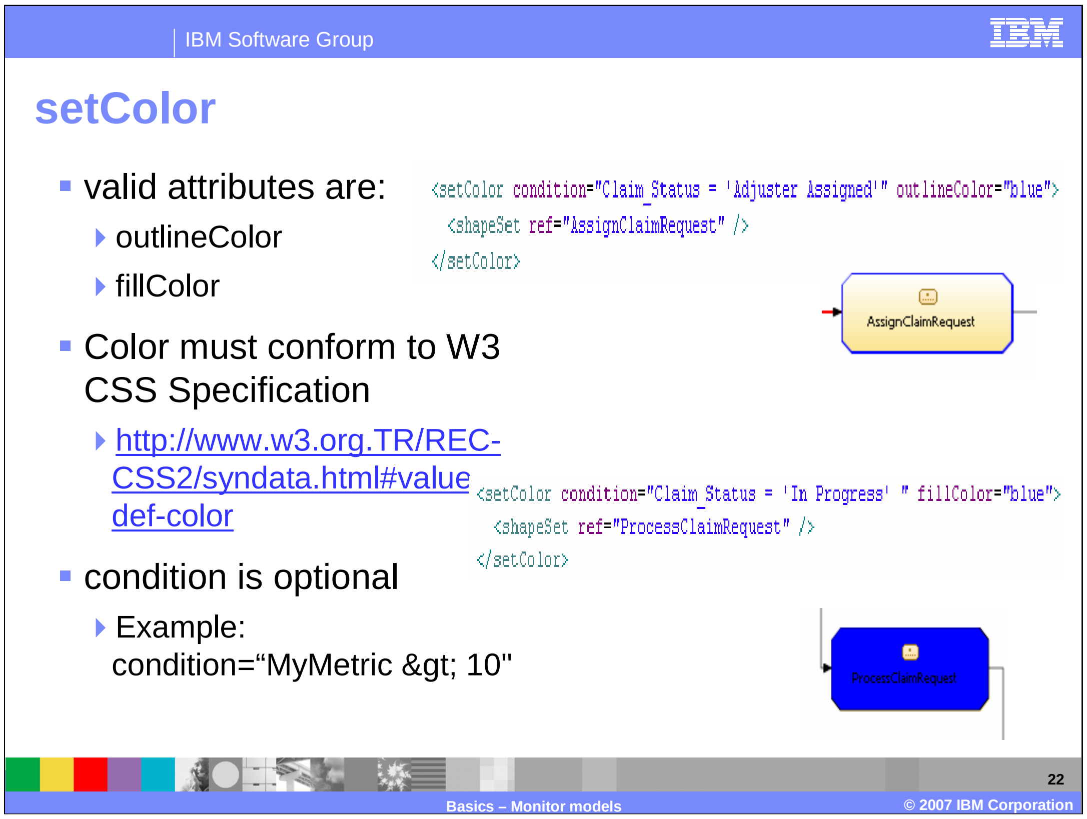
In the .mm file:

```
<shapeSet displayName="ProcessClaimRequest (TaskNode)" id="ProcessClaimRequest">
    <shapes>17 </shapes>
</shapeSet>
```

In the SVG file:

```
<text clip-path="url(#clipPath38)" fill="black" stroke="none" stroke-width="0" x
<text clip-path="url(#clipPath38)" fill="black" stroke="none" stroke-width="0" x
<polygon clip-path="url(#clipPath39)" fill="rgb(237,243,247)" points=" 912 309 9
<polygon clip-path="url(#clipPath39)" fill="none" mm:id="17" points=" 912 309 92
<text clip-path="url(#clipPath39)" fill="black" stroke="none" stroke-width="0" x
<polygon clip-path="url(#clipPath40)" fill="rgb(237,243,247)" points=" 796 341 8
<polygon clip-path="url(#clipPath40)" fill="none" mm:id="18" points=" 796 341 80
<text clip-path="url(#clipPath40)" fill="black" stroke="none" stroke-width="0" x
```

A shapeSet identifies a particular shape in the SVG diagram, which might be a box, or a line connector between boxes. In this example there is a box displayed in the diagram view but in the SVG file it is listed as a polygon shape. To provide a reference to this, you add the mm:id attribute to the SVG file. Then you create a corresponding shapeSet tag in the visual model which has the same numeric identifier, in this case the number 17.



setColor

- valid attributes are:
  - outlineColor
  - fillColor
- Color must conform to W3 CSS Specification
  - http://www.w3.org.TR/REC-CSS2/syndata.html#value def-color
- condition is optional
  - Example: condition="MyMetric &gt; 10"

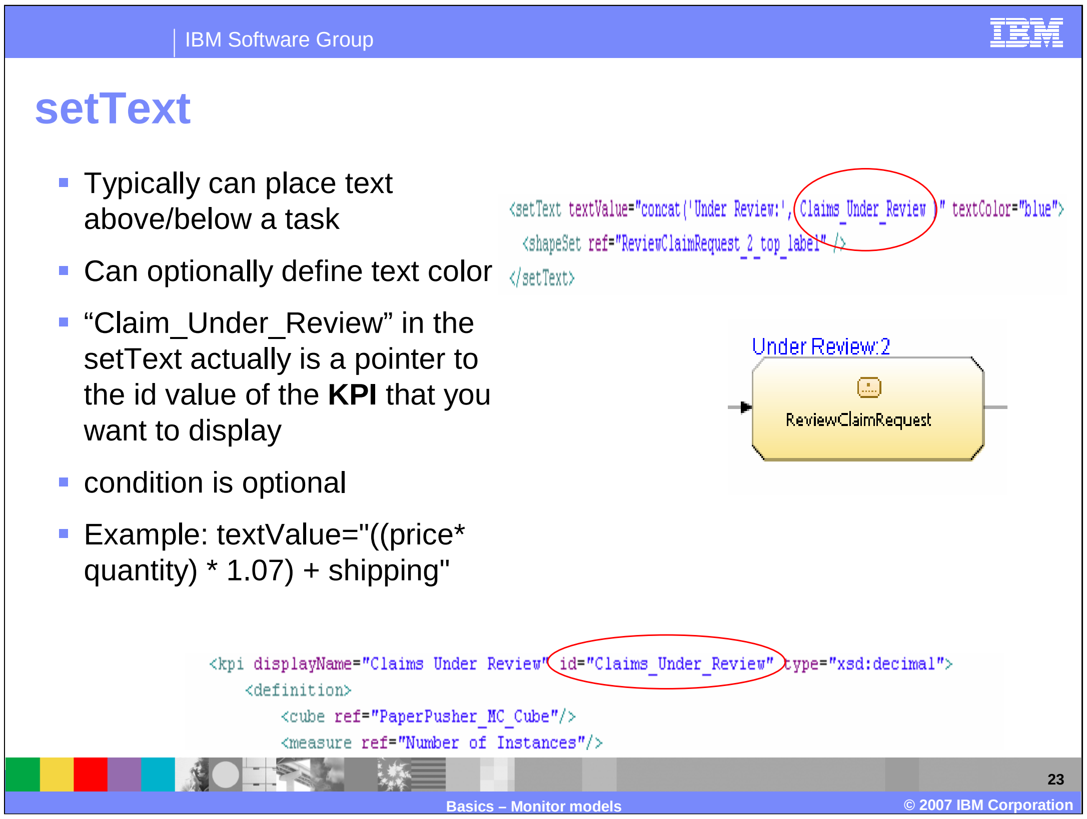
```
<setColor condition="Claim_Status = 'Adjuster Assigned'" outlineColor="blue">
  <shapeSet ref="AssignClaimRequest" />
</setColor>
```

AssignClaimRequest

```
<setColor condition="Claim_Status = 'In Progress' " fillColor="blue">
  <shapeSet ref="ProcessClaimRequest" />
</setColor>
```

ProcessClaimRequest

setColor is an action which sets the color of the shapes outline or fill. You can use attributes outlineColor or fillColor, but the color that you specify for them must adhere to the CSS specification that is listed on this slide. You can also set a condition so that the action is taken only based on the condition listed, so for example, only set the color if a particular metric is a certain value.

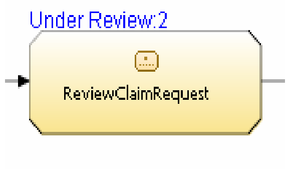


IBM Software Group

setText

- Typically can place text above/below a task
- Can optionally define text color
- “Claim_Under_Review” in the setText actually is a pointer to the id value of the **KPI** that you want to display
- condition is optional
- Example: textValue="((price* quantity) * 1.07) + shipping"

```
<setText textValue="concat('Under Review:',Claims_Under_Review )" textColor="blue">
  <shapeSet ref="ReviewClaimRequest_2_top_label" />
</setText>
```



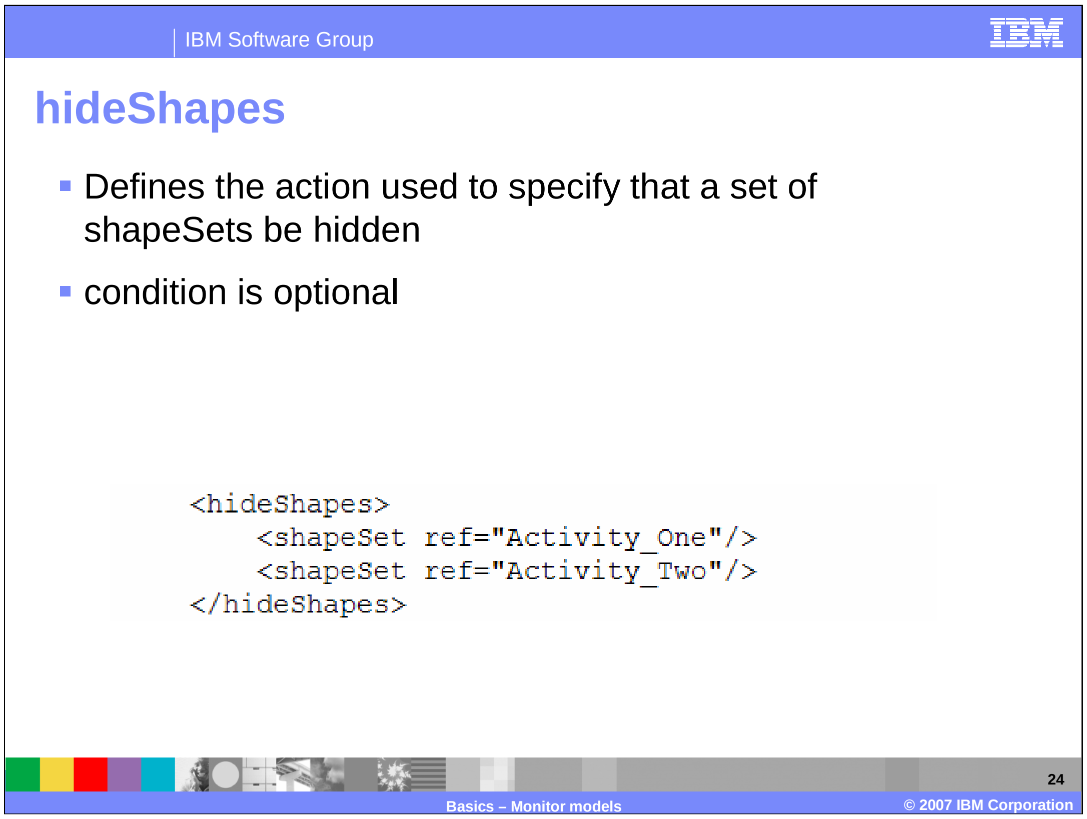
```
<kpi displayName="Claims Under Review" id="Claims_Under_Review" type="xsd:decimal">
    <definition>
        <cube ref="PaperPusher_MC_Cube"/>
        <measure ref="Number of Instances"/>
```

23

Basics – Monitor models

© 2007 IBM Corporation

The setText action tag is used to place text on the diagram. You can use the attribute textColor to define the color of the text. You can also display literal text or text which is derived from the value of a metric or KPI. You can optionally use conditions on the setText tag.

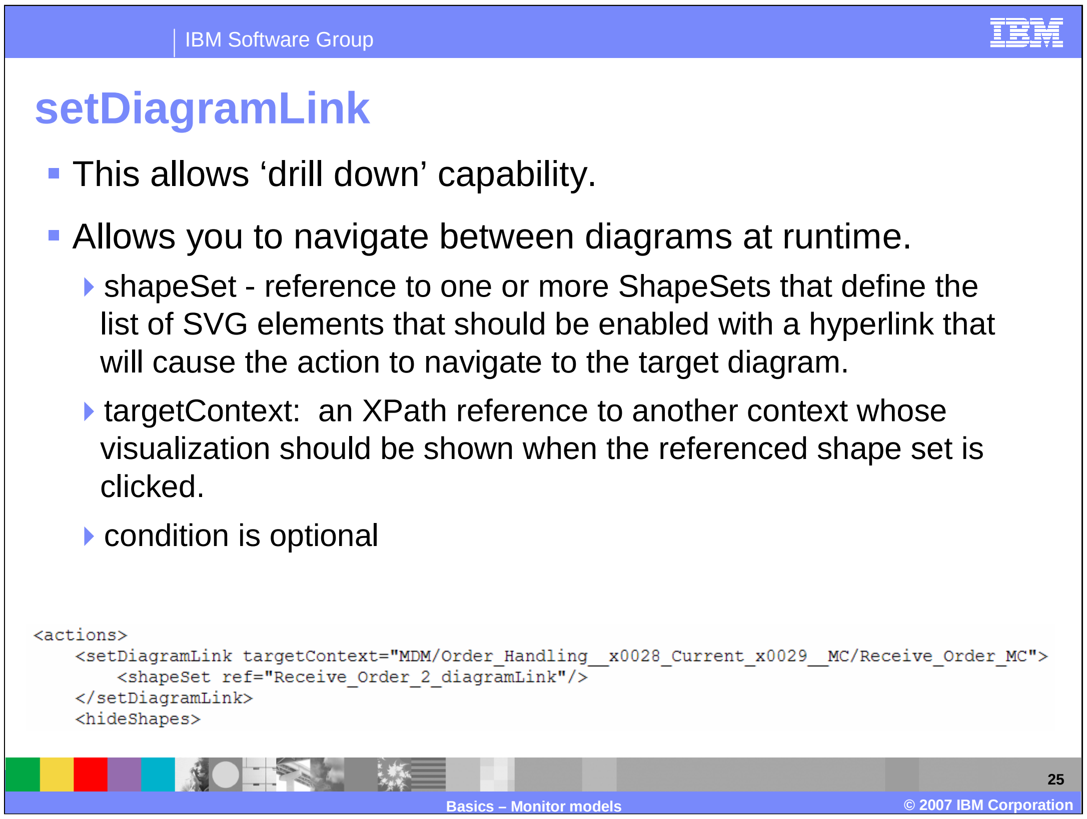


hideShapes

- Defines the action used to specify that a set of shapeSets be hidden
- condition is optional

```
<hideShapes>
    <shapeSet ref="Activity_One"/>
    <shapeSet ref="Activity_Two"/>
</hideShapes>
```

The hideShapes tag is used to define shapes that you which to hide. You can optionally use conditions on the hideShapes tag.



setDiagramLink

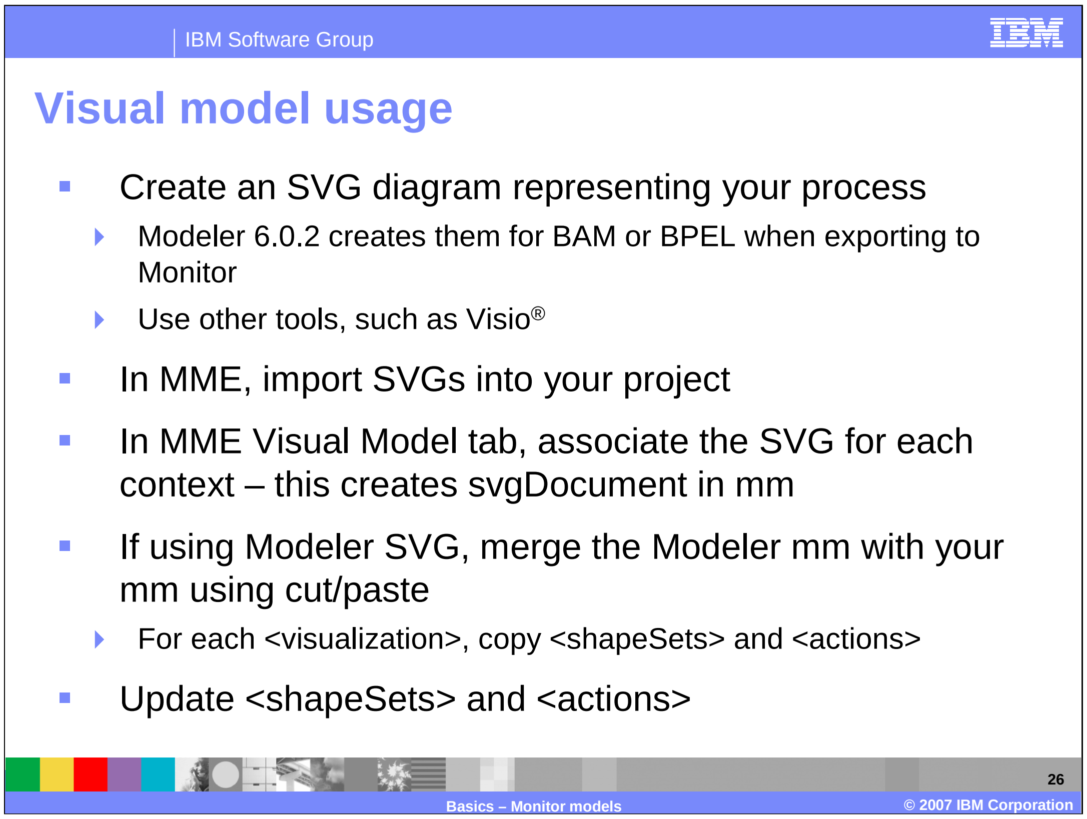
- This allows ‘drill down’ capability.
- Allows you to navigate between diagrams at runtime.
  - shapeSet - reference to one or more ShapeSets that define the list of SVG elements that should be enabled with a hyperlink that will cause the action to navigate to the target diagram.
  - targetContext: an XPath reference to another context whose visualization should be shown when the referenced shape set is clicked.
  - condition is optional

```
<actions>
    <setDiagramLink targetContext="MDM/Order_Handling__x0028_Current_x0029__MC/Receive_Order_MC">
        <shapeSet ref="Receive_Order_2_diagramLink"/>
    </setDiagramLink>
    <hideShapes>
```

The setDiagramLink tag can be used to allow drilling down to other diagrams. Using this link, you click on the referenced shape and then it is treated like a hyperlink that causes navigation to the target diagram.

The shapeSet attribute is the shape that causes the link when it is clicked. The targetContext attribute is the name of the target context in the model whose visual model will be displayed when you click on the shape.

You can optionally use conditions on the setDiagramLink tag.


IBM Software Group

Visual model usage

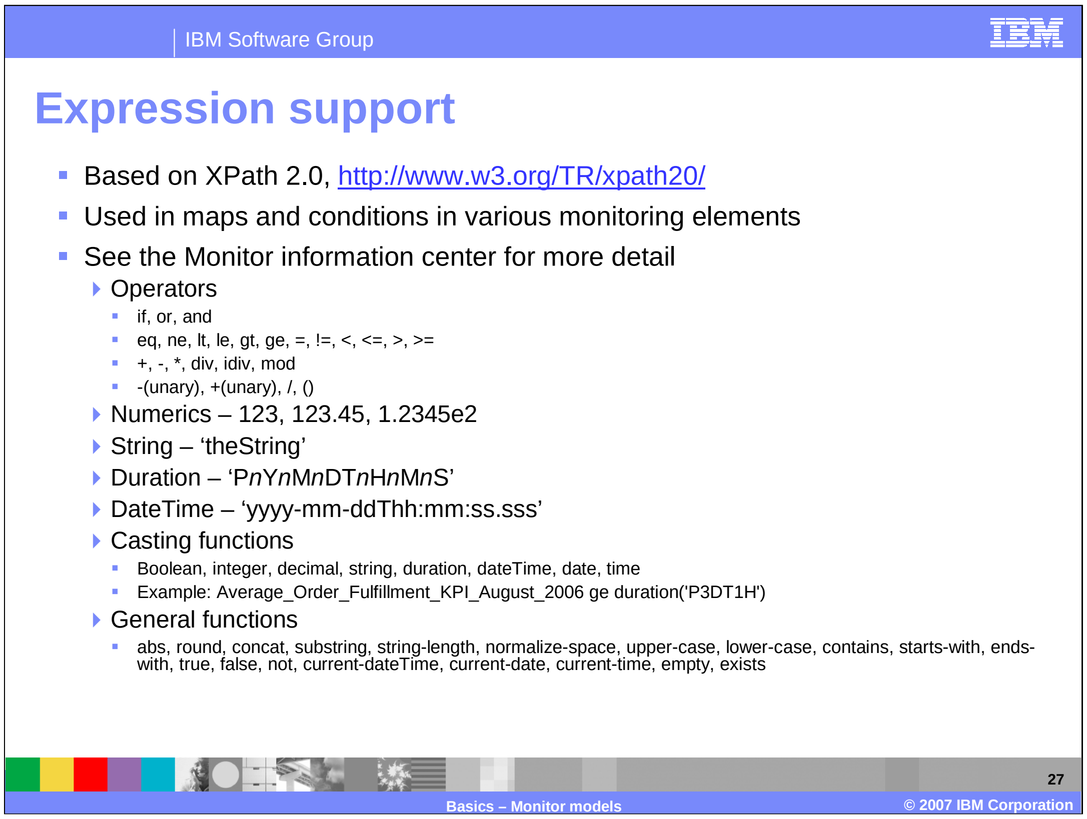
- Create an SVG diagram representing your process
  - Modeler 6.0.2 creates them for BAM or BPEL when exporting to Monitor
  - Use other tools, such as Visio®
- In MME, import SVGs into your project
- In MME Visual Model tab, associate the SVG for each context – this creates svgDocument in mm
- If using Modeler SVG, merge the Modeler mm with your mm using cut/paste
  - For each <visualization>, copy <shapeSets> and <actions>
- Update <shapeSets> and <actions>

26

Basics – Monitor models

© 2007 IBM Corporation

This is the usage scenario for the visual model. You create an SVG diagram using Modeler or other tools such as Visio. In the Monitor Model Editor, you import the SVG files into your project, then you use the visual model tab in the MME to associate an imported SVG diagram with each monitoring context and each KPI context. If you are using Modeler to create your high level monitor model with diagrams, then you will need to merge this with the monitor model that you may have already started in the MME. This is done by using cut and paste of the visualization sections of the Modeler model. Finally, you would add the shapeSets and actions to the visual model that provide the diagram actions that you want to see.

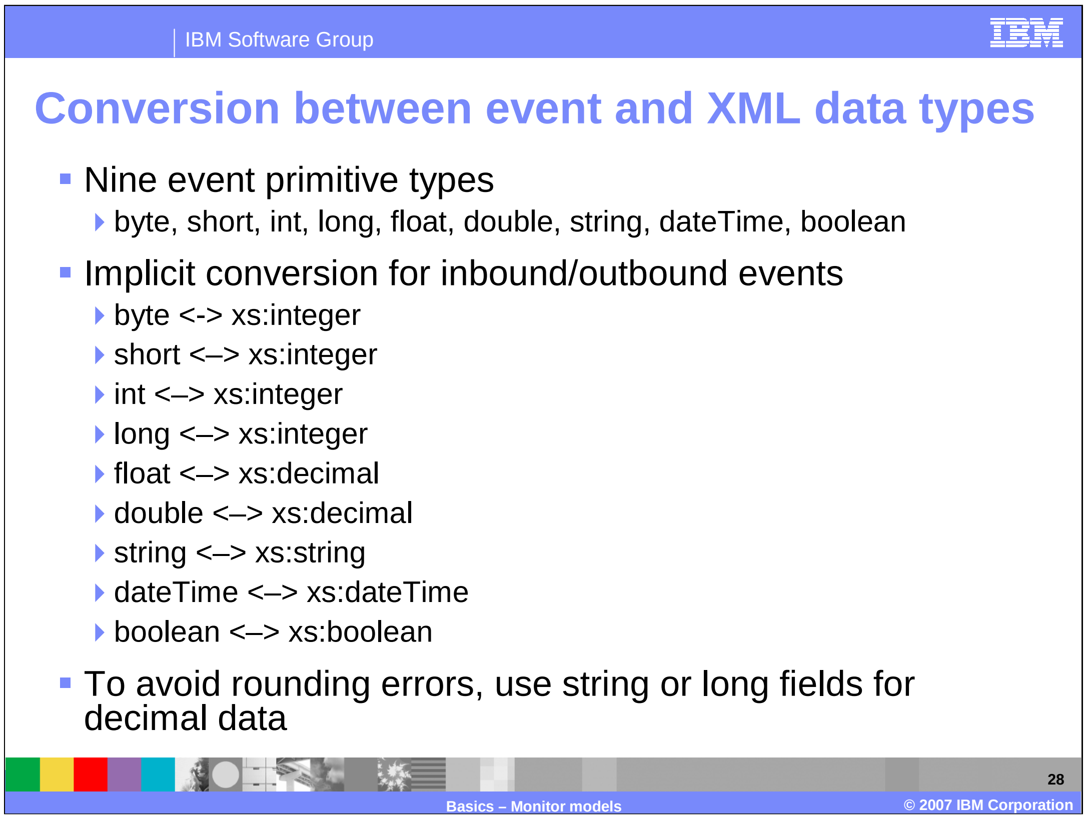


Expression support

- Based on XPath 2.0, http://www.w3.org/TR/xpath20/
- Used in maps and conditions in various monitoring elements
- See the Monitor information center for more detail
  - Operators
    - if, or, and
    - eq, ne, lt, le, gt, ge, =, !=, <, <=, >, >=
    - +, -, *, div, idiv, mod
    - -(unary), +(unary), /, ()
  - Numerics – 123, 123.45, 1.2345e2
  - String – 'theString'
  - Duration – 'P*n*Y*n*M*n*DT*n*H*n*M*n*S'
  - DateTime – 'yyyy-mm-ddThh:mm:ss.sss'
  - Casting functions
    - Boolean, integer, decimal, string, duration, dateTime, date, time
    - Example: Average_Order_Fulfillment_KPI_August_2006 ge duration('P3DT1H')
  - General functions
    - abs, round, concat, substring, string-length, normalize-space, upper-case, lower-case, contains, starts-with, ends-with, true, false, not, current-dateTime, current-date, current-time, empty, exists

Expressions that are used in the monitor model are based on XPath 2.0, and they are used in the visual model, metric maps, and conditions in various monitoring elements.

There are many operators, casting functions and general functions that you can use with XPath. For more information, see the Monitor information center.

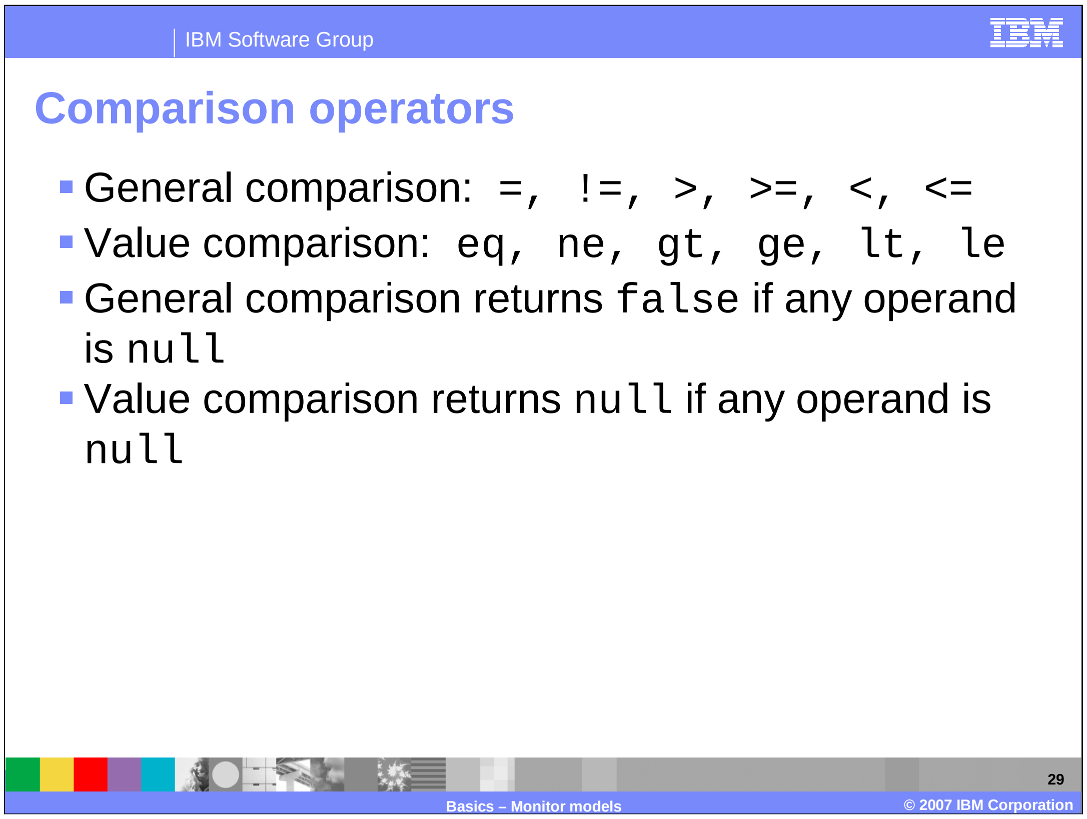


Conversion between event and XML data types

- Nine event primitive types
  - byte, short, int, long, float, double, string, dateTime, boolean
- Implicit conversion for inbound/outbound events
  - byte <-> xs:integer
  - short <–> xs:integer
  - int <–> xs:integer
  - long <–> xs:integer
  - float <–> xs:decimal
  - double <–> xs:decimal
  - string <–> xs:string
  - dateTime <–> xs:dateTime
  - boolean <–> xs:boolean
- To avoid rounding errors, use string or long fields for decimal data

Since there are different data types in common base events and XML, then conversion will have to take place for inbound and outbound events. Those are listed in this slide.

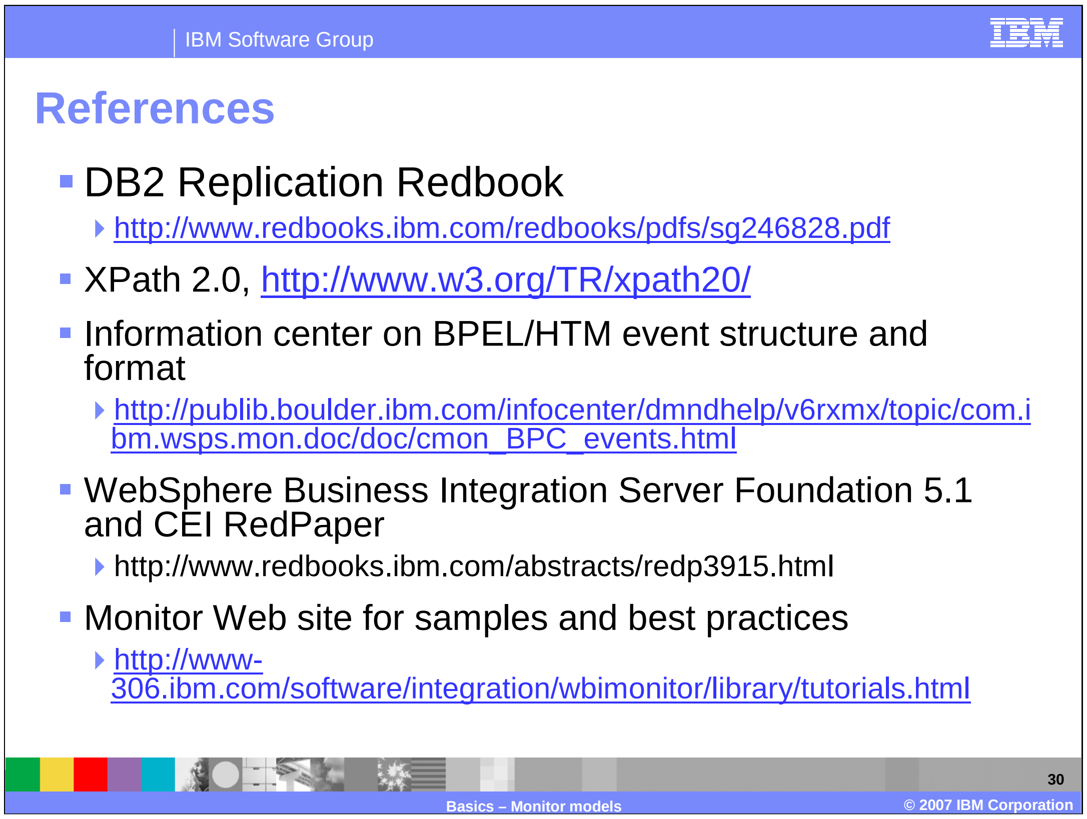
Many decimal numbers cannot be represented exactly as a binary floating point number (data types float or double). Therefore, rounding errors may occur when decimal data is transferred as float or double in events. If rounding errors must be avoided, decimal numbers should be converted to string or integer by multiplying by some fixed power of 10, and then transmitted as string or long fields in events.



Comparison operators

- General comparison: `=, !=, >, >=, <, <=`
- Value comparison: `eq, ne, gt, ge, lt, le`
- General comparison returns `false` if any operand is `null`
- Value comparison returns `null` if any operand is `null`

There are two sets of comparison operations as you see listed on the slide, either general comparison operators or value comparison operators. They work the same except with regard to nulls. Note that general comparison returns `false` if any operand is `null`, `but` value comparison returns `null` if any operand is `null`.


IBM Software Group

References

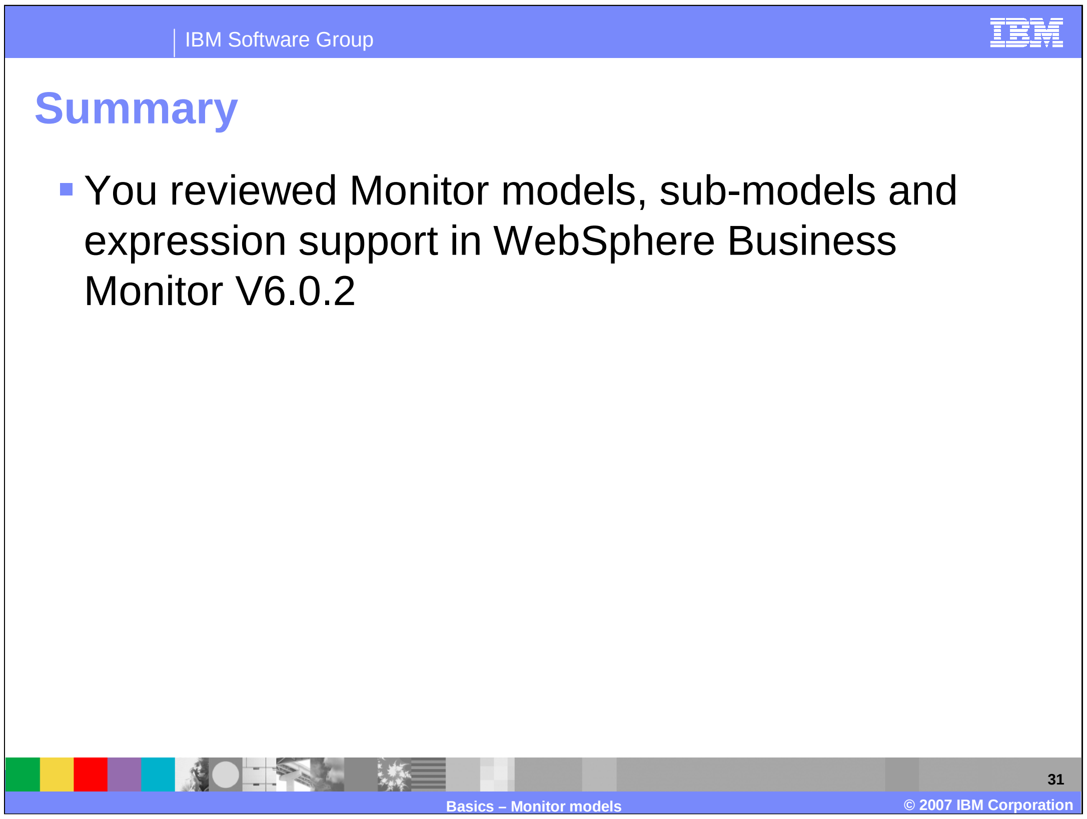
- DB2 Replication Redbook
  - http://www.redbooks.ibm.com/redbooks/pdfs/sg246828.pdf
- XPath 2.0, http://www.w3.org/TR/xpath20/
- Information center on BPEL/HTM event structure and format
  - http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wsps.mon.doc/doc/cmon_BPC_events.html
- WebSphere Business Integration Server Foundation 5.1 and CEI RedPaper
  - http://www.redbooks.ibm.com/abstracts/redp3915.html
- Monitor Web site for samples and best practices
  - http://www-306.ibm.com/software/integration/wbimonitor/library/tutorials.html

30

Basics – Monitor models

© 2007 IBM Corporation

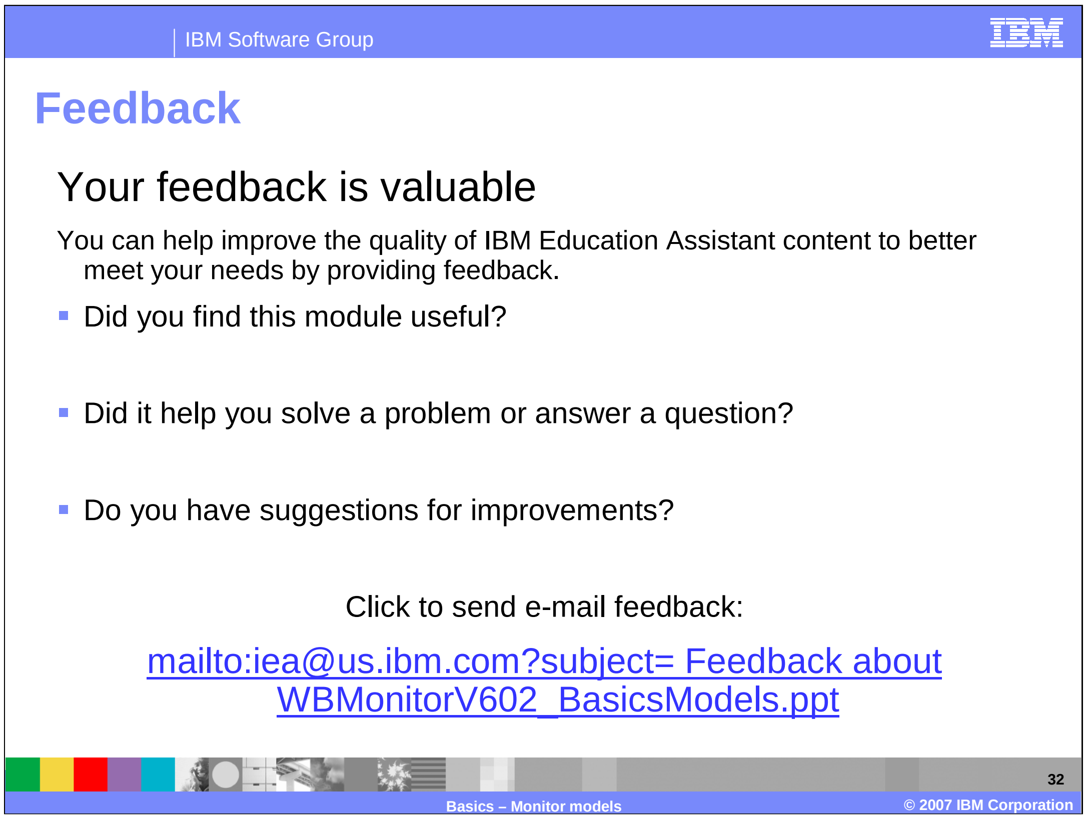
This slide shows some references including for DB2 replication, XPath, BPEL Common Base Event formats and CEI. The CEI red paper is based on Server Foundation 5.1 but is useful to understand the CEI APIs. Also, a Web site is available that contains Monitor samples and best practices.



Summary

- You reviewed Monitor models, sub-models and expression support in WebSphere Business Monitor V6.0.2

In this presentation you have reviewed Monitor models, sub-models and expression support for WebSphere Business Monitor V6.0.2.



IBM Software Group

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 e-mail feedback:

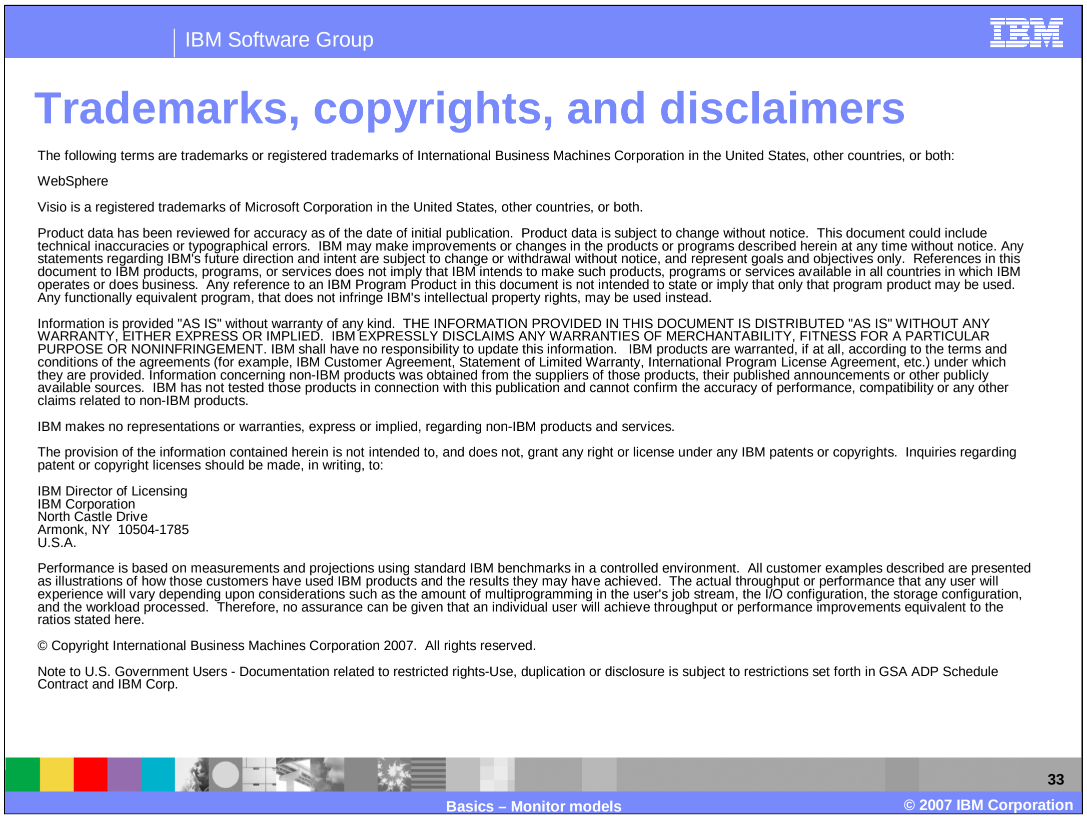
mailto:iea@us.ibm.com?subject= Feedback about WBMonitorV602_BasicsModels.ppt

32

Basics – Monitor models

© 2007 IBM Corporation

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



Trademarks, copyrights, and disclaimers