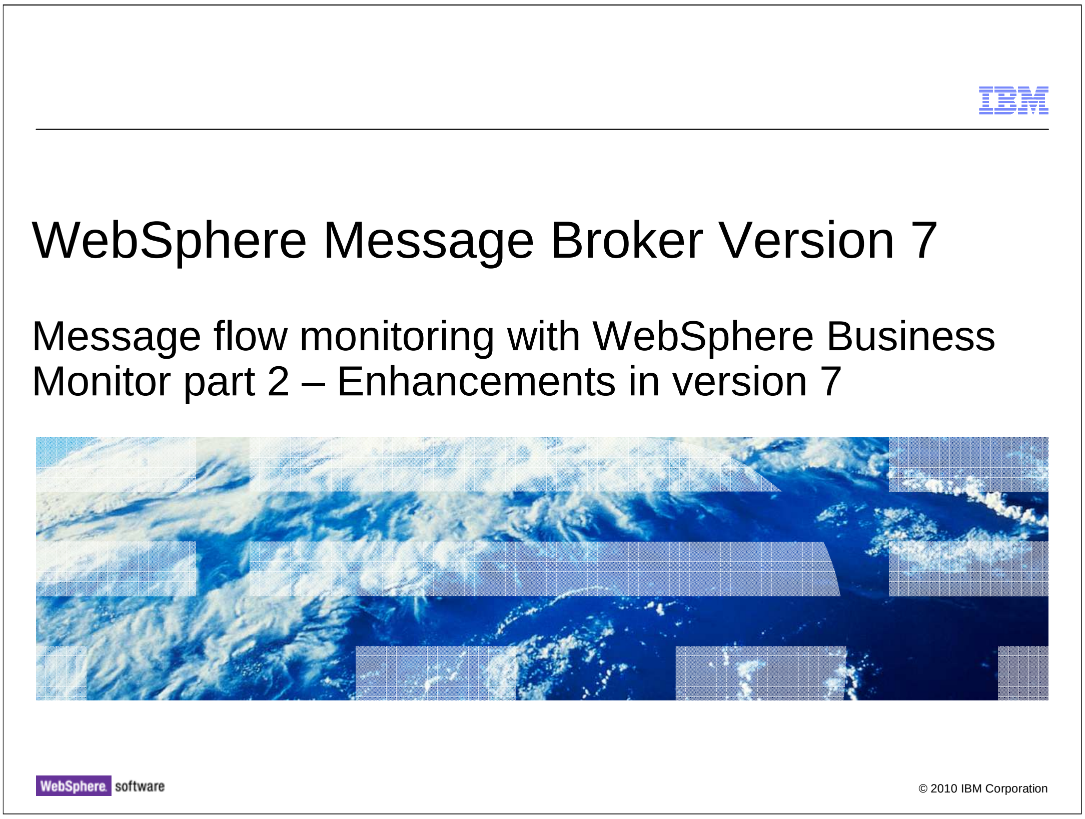
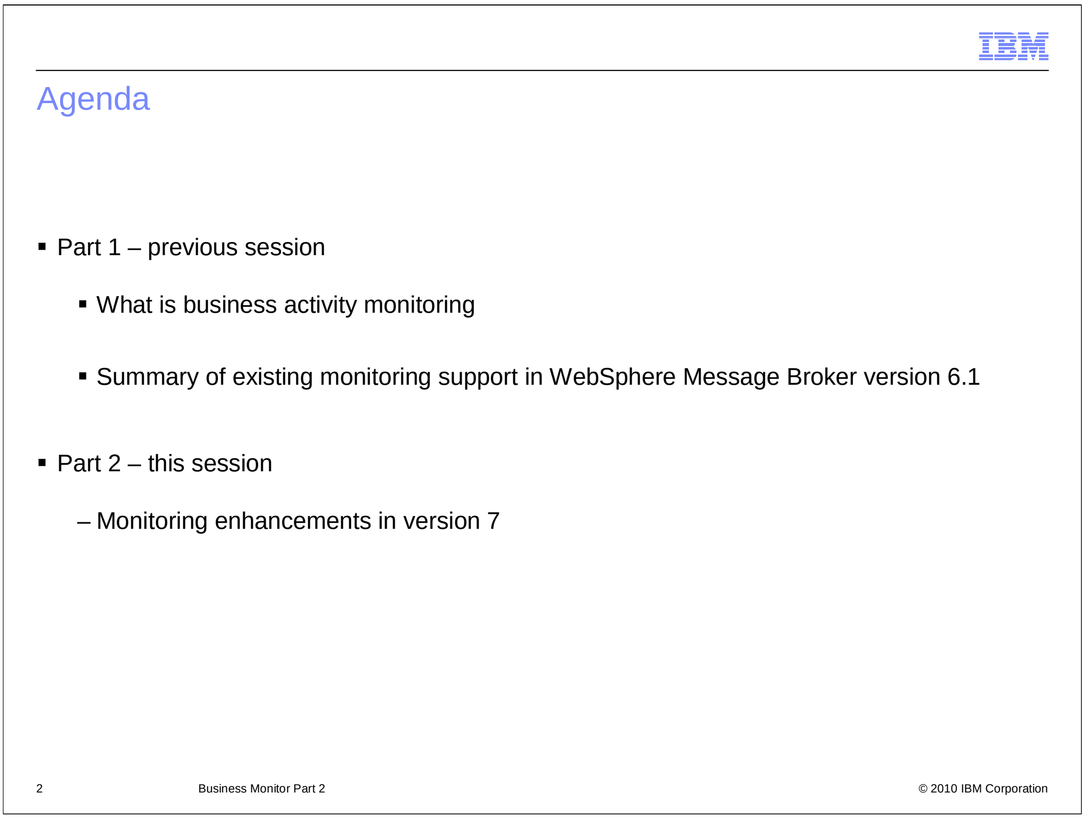


WebSphere Message Broker Version 7

Message flow monitoring with WebSphere Business Monitor part 2 – Enhancements in version 7

This session is part 2 of the topic covering the business monitoring functions in WebSphere Message Broker.



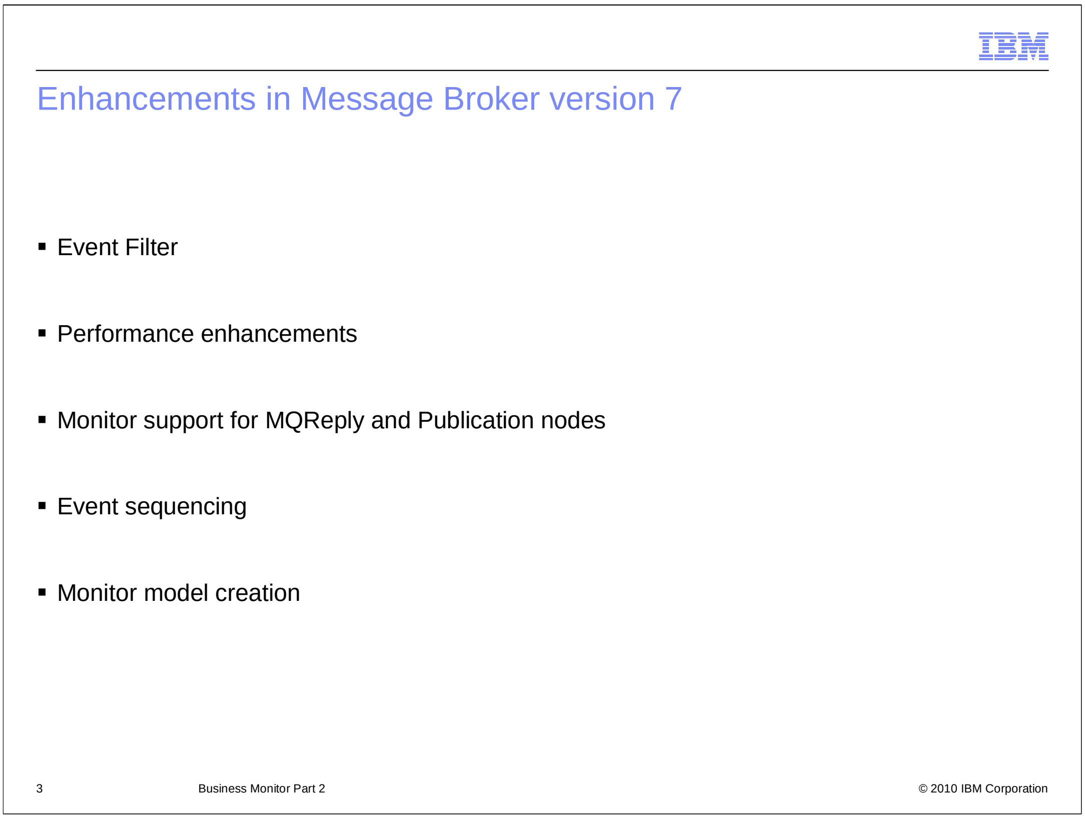
Agenda

- Part 1 – previous session
  - What is business activity monitoring
  - Summary of existing monitoring support in WebSphere Message Broker version 6.1
- Part 2 – this session
  - Monitoring enhancements in version 7

2 Business Monitor Part 2 © 2010 IBM Corporation

The first session in this topic covered the background to business monitoring, and the tools that are already available in WebSphere Message Broker version 6.1 to enable business event monitoring.

This session covers the new functions that are available for business monitoring in WebSphere Message Broker version 7, and how to take advantage of WebSphere Business Monitor to capture and display this information.



Enhancements in Message Broker version 7

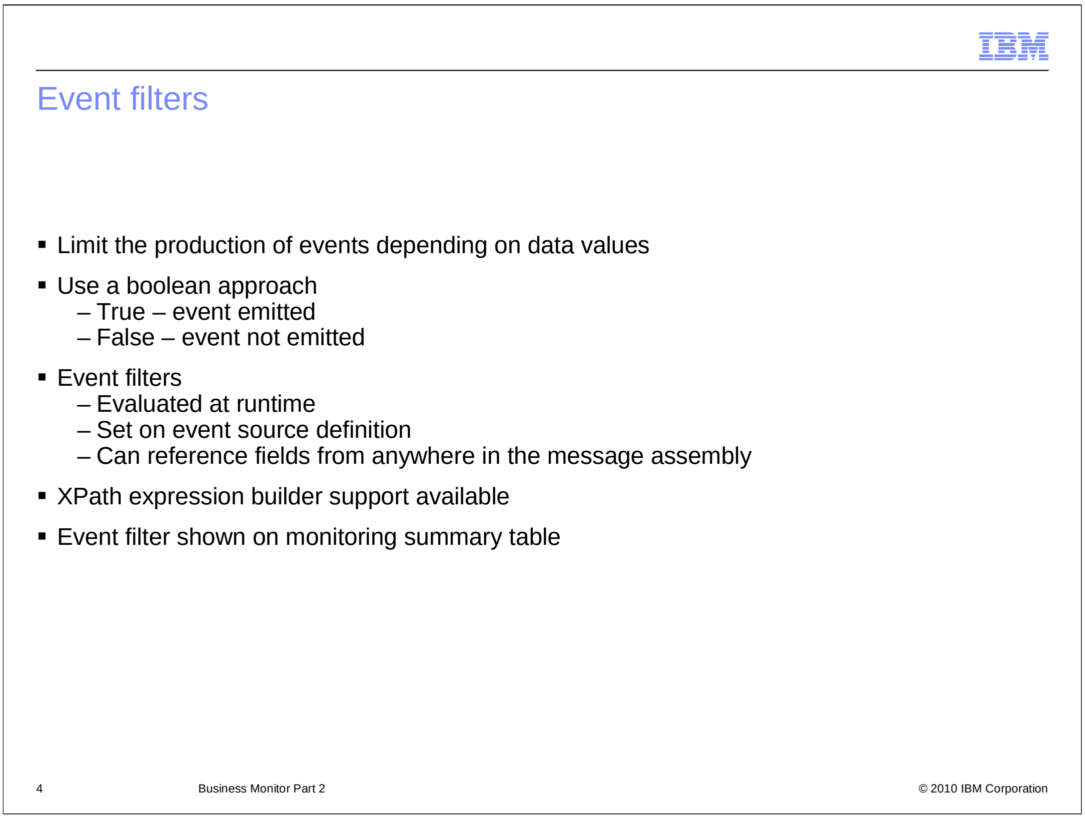
- Event Filter
- Performance enhancements
- Monitor support for MQReply and Publication nodes
- Event sequencing
- Monitor model creation

3 Business Monitor Part 2 © 2010 IBM Corporation

As described in part 1 of this topic, Message Broker version 6.1 already provides the basic infrastructure and tools to enable business events to be generated and captured.

Message Broker version 7 extends this support in several areas. The provision of event filtering provides more granularity for emitting events. Some improvements have been made to the performance of the overall system. Support for the few remaining nodes is now provided, and support for event sequencing is also available.

Finally, the integration between the Message Broker Toolkit and the corresponding WebSphere Business Monitor Toolkit plug-in is enhanced through the automatic creation of a monitor model.



Event filters

- Limit the production of events depending on data values
- Use a boolean approach
  - True – event emitted
  - False – event not emitted
- Event filters
  - Evaluated at runtime
  - Set on event source definition
  - Can reference fields from anywhere in the message assembly
- XPath expression builder support available
- Event filter shown on monitoring summary table

4 Business Monitor Part 2 © 2010 IBM Corporation

The first enhancement is event filtering. In Message Broker version 6.1, if you specify a monitoring event for a particular node or terminal, then monitoring data is emitted every time that data flows through that event point. Whilst this can be enabled or disabled, there is no further granularity of event data emission.

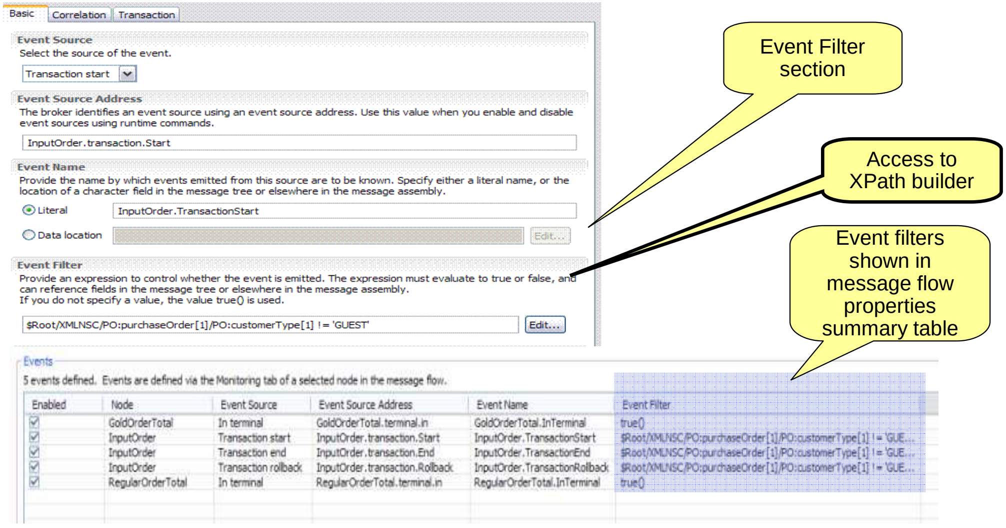
Message Broker version 7 has introduced an event filter function. This enables you to emit monitoring event data only if the event filter test is passed. This filter is tested at runtime each time the message flow monitoring function is invoked, and uses a boolean approach for this test. If the event filter test is passed, then the event data is emitted.

The event filter is specified in the Message Broker Toolkit, using the monitoring tools, and can reference any field from the message tree that is passing through the specified node. As before, this field is referenced by using the X-Path notation, and the toolkit uses the X-Path expression builder.

The event filter information is shown on the monitoring summary when you click the properties of the message flow.

This control can also be specified in the configurable service XML file. However, if this technique is used, this is not visible in the Message Broker Toolkit.

Event filter specification



Event Source

Select the source of the event.

Transaction start

Event Source Address

The broker identifies an event source using an event source address. Use this value when you enable and disable event sources using runtime commands.

InputOrder.transaction.Start

Event Name

Provide the name by which events emitted from this source are to be known. Specify either a literal name, or the location of a character field in the message tree or elsewhere in the message assembly.

● Literal InputOrder.TransactionStart

○ Data location

Event Filter

Provide an expression to control whether the event is emitted. The expression must evaluate to true or false, and can reference fields in the message tree or elsewhere in the message assembly.
If you do not specify a value, the value true() is used.

$Root/XMLNSC/PO:purchaseOrder[1]/PO:customerType[1] != 'GUEST'

Events

5 events defined. Events are defined via the Monitoring tab of a selected node in the message flow.

Enabled	Node	Event Source	Event Source Address	Event Name	Event Filter
☑	GoldOrderTotal	In terminal	GoldOrderTotal.terminal.in	GoldOrderTotal.InTerminal	true()
☑	InputOrder	Transaction start	InputOrder.transaction.Start	InputOrder.TransactionStart	$Root/XMLNSC/PO:purchaseOrder[1]/PO:customerType[1] != 'GUE...
☑	InputOrder	Transaction end	InputOrder.transaction.End	InputOrder.TransactionEnd	$Root/XMLNSC/PO:purchaseOrder[1]/PO:customerType[1] != 'GUE...
☑	InputOrder	Transaction rollback	InputOrder.transaction.Rollback	InputOrder.TransactionRollback	$Root/XMLNSC/PO:purchaseOrder[1]/PO:customerType[1] != 'GUE...
☑	RegularOrderTotal	In terminal	RegularOrderTotal.terminal.in	RegularOrderTotal.InTerminal	true()

When you open the event monitoring wizard, you will see a window similar to that shown in this example. In Message Broker version 7, the event filter section has been added, on the basic tab. Click the Edit button to define or change event filters. This will open the XPath expression builder dialogue, shown on the next slide.

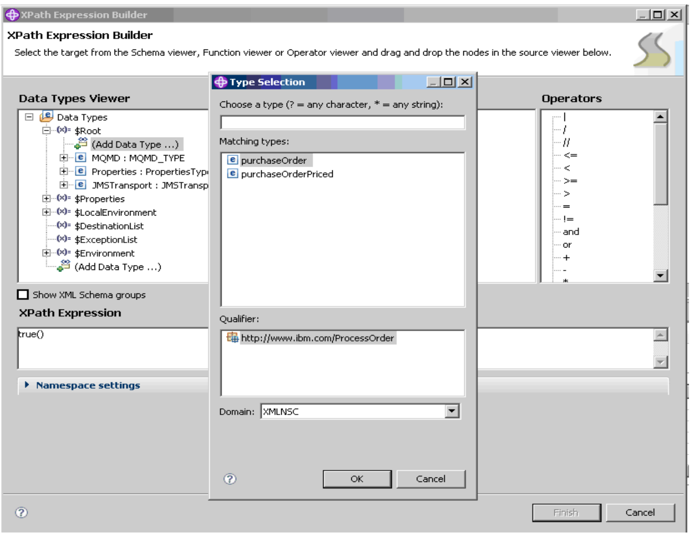
The event filter can be set to a numeric, boolean or string XPath expression which will evaluate to boolean true or false. Typically, this is the result of an equality test, or a “greater than” or “less than” test.

If the expression evaluates to true then events are emitted. If the expression evaluates to false then events are not emitted.

The default setting is true.

The second screen capture on this slide shows the monitoring events summary, which is shown when you click the message flow canvas, and select the Monitoring tab. A new column called “Event Filter” has been introduced, showing all the defined filters for this message flow.

Event filter XPath expression builder



XPath Expression Builder

Select the target from the Schema viewer, Function viewer or Operator viewer and drag and drop the nodes in the source viewer below.

Type Selection

Choose a type (? = any character, * = any string):

Matching types:

- purchaseOrder
- purchaseOrderPriced

Qualifier: http://www.ibm.com/ProcessOrder

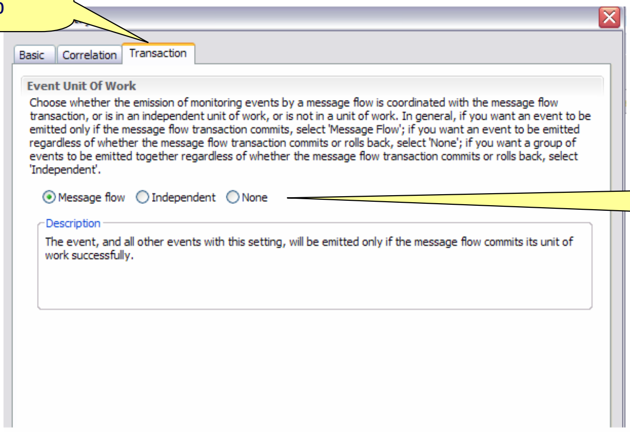
Domain: XMLNSC

XPath Expression: true()

Clicking the Edit button for the event filter opens the X-Path expression builder. This uses the same dialogue to create the required X-Path as with the event payload specification, and with other nodes which need to specify a message field location.

Performance enhancements – Transaction inclusion

New transaction tab



Event Unit Of Work

Choose whether the emission of monitoring events by a message flow is coordinated with the message flow transaction, or is in an independent unit of work, or is not in a unit of work. In general, if you want an event to be emitted only if the message flow transaction commits, select 'Message Flow'; if you want an event to be emitted regardless of whether the message flow transaction commits or rolls back, select 'None'; if you want a group of events to be emitted together regardless of whether the message flow transaction commits or rolls back, select 'Independent'.

● Message flow ○ Independent ○ None

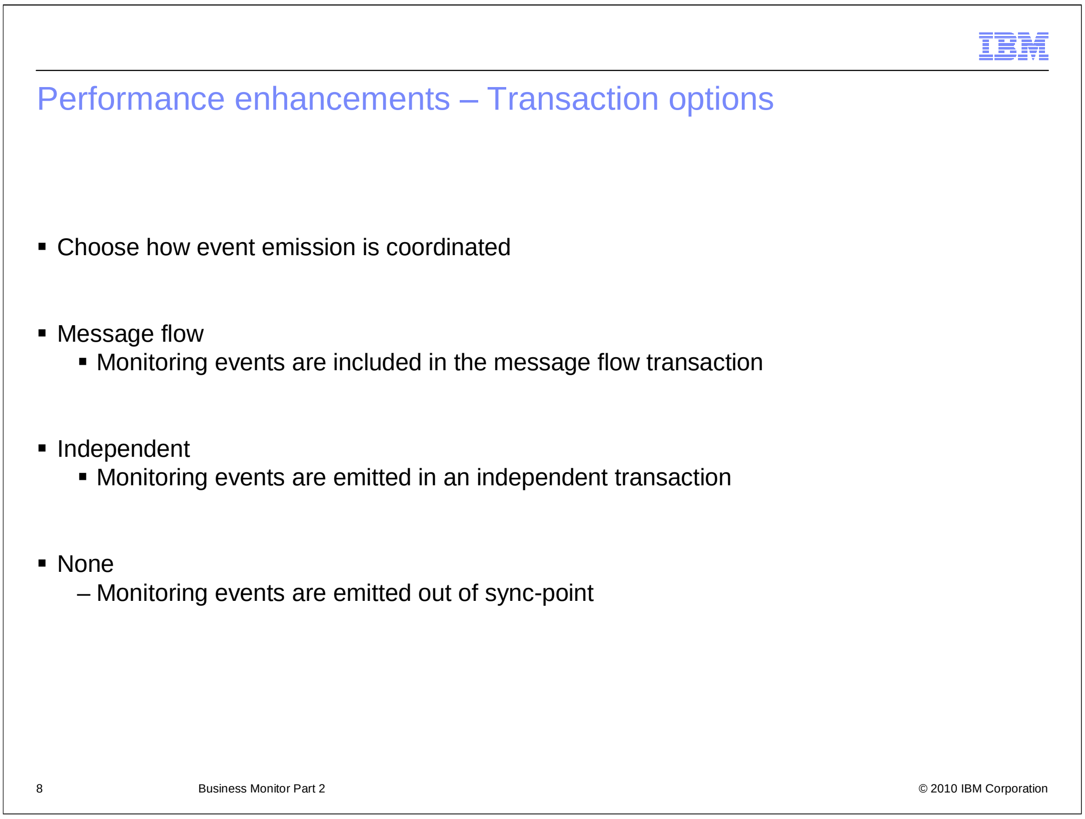
Description

The event, and all other events with this setting, will be emitted only if the message flow commits its unit of work successfully.

Select the unit of work the event is emitted in

Message Broker version 7 has improved performance in several ways. Whilst most of these are internal optimizations, it is possible to affect performance by specifying whether a monitoring event is created within the transaction boundary of the associated message flow.

When you open the monitoring event window, you use the Transaction tab to specify how monitoring events are to be emitted. A monitoring event can be emitted as part of the message flow transaction, as part of an independent transaction, or outside the control of any transaction.



Performance enhancements – Transaction options

- Choose how event emission is coordinated
- Message flow
  - Monitoring events are included in the message flow transaction
- Independent
  - Monitoring events are emitted in an independent transaction
- None
  - Monitoring events are emitted out of sync-point

8 Business Monitor Part 2 © 2010 IBM Corporation

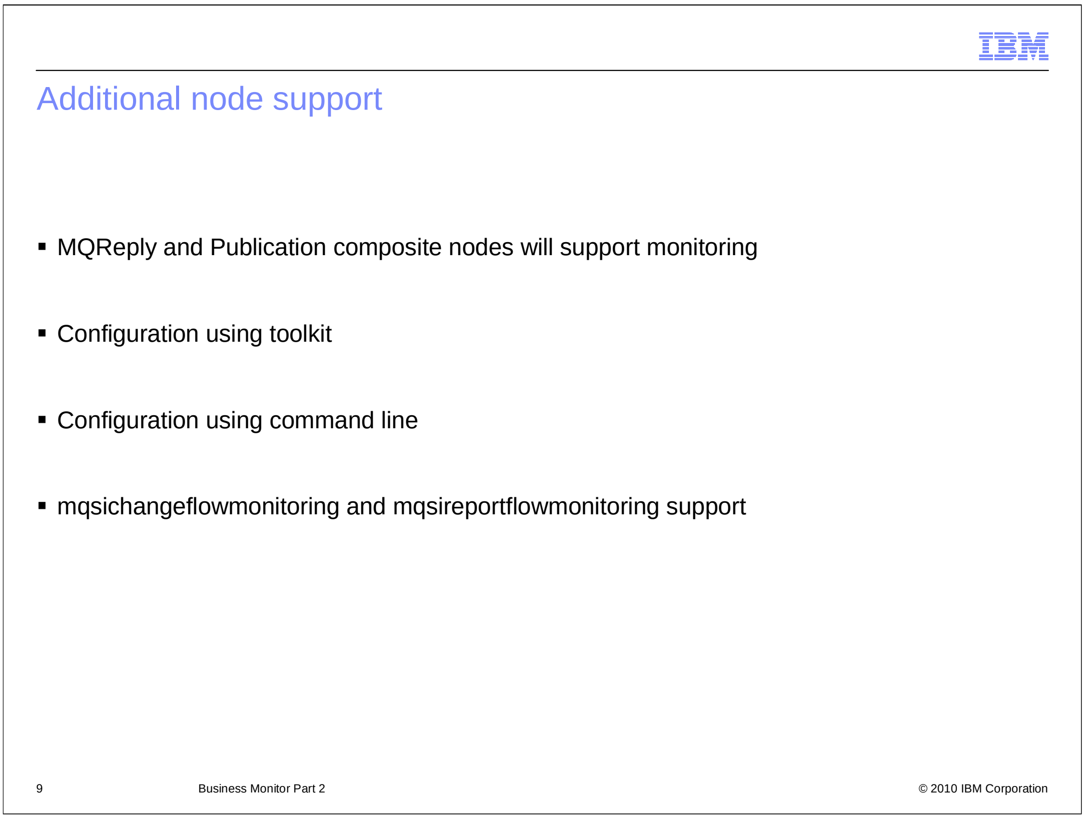
When a message is processed by a message flow, the MQ updates are included in a unit of work referred to as the “M**essage Flow**” unit of work. This transaction is committed if the message processing is successful and rolled back if it fails.

If you specify “Message Flow” for the unit of work for the monitoring event, this event is included in the transaction of the overall message flow. This will typically be used for all events created on the successful path through the message flow. When a message flow completes successfully, then all event data will also be published and committed with the message flow. If the message flow is rolled back, then monitoring data from these event points will also be rolled back.

The “**Independent**” unit of work is a separate unit of work which is created and committed regardless of whether the message is processed successfully or not. Use this for events, such as those related to error paths, that must be published even if the flow fails. This will typically be used for events that are created on error paths of the message flow, and is invoked as part of the transaction that is started if the main part of the message flow terminates.

If you do not want a monitoring event to be included in any unit of work, choose the “**None**” option.

For information, in Message Broker version 6.1, this facility is not available. In this version, events are published outside of the message flow transaction, the equivalent of the None option in version 7.



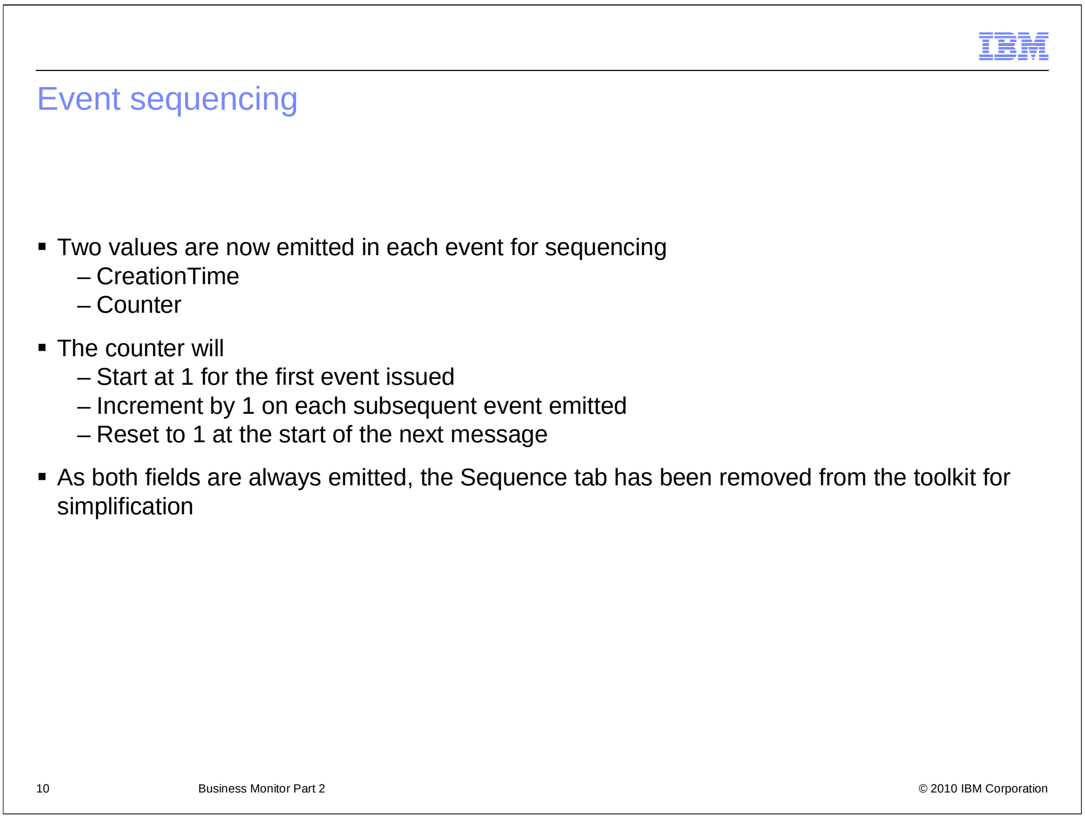
Additional node support

- MQReply and Publication composite nodes will support monitoring
- Configuration using toolkit
- Configuration using command line
- mqsichangeflowmonitoring and mqsireportflowmonitoring support

9 Business Monitor Part 2 © 2010 IBM Corporation

The third enhancement in version 7 is the provision of the monitoring event points on the MQ Reply and Publication nodes. These nodes did not previously support monitoring, because of the internal structure of the node. This has now been addressed in version 7.

As with all other nodes, the specification and configuration of these nodes is done using the toolkit and the command line interface. The flow monitoring commands are unchanged, and can now be applied to message flows which contain these nodes.



Event sequencing

- Two values are now emitted in each event for sequencing
  - CreationTime
  - Counter
- The counter will
  - Start at 1 for the first event issued
  - Increment by 1 on each subsequent event emitted
  - Reset to 1 at the start of the next message
- As both fields are always emitted, the Sequence tab has been removed from the toolkit for simplification

10 Business Monitor Part 2 © 2010 IBM Corporation

The next enhancement in version 7 is the provision of event sequencing.

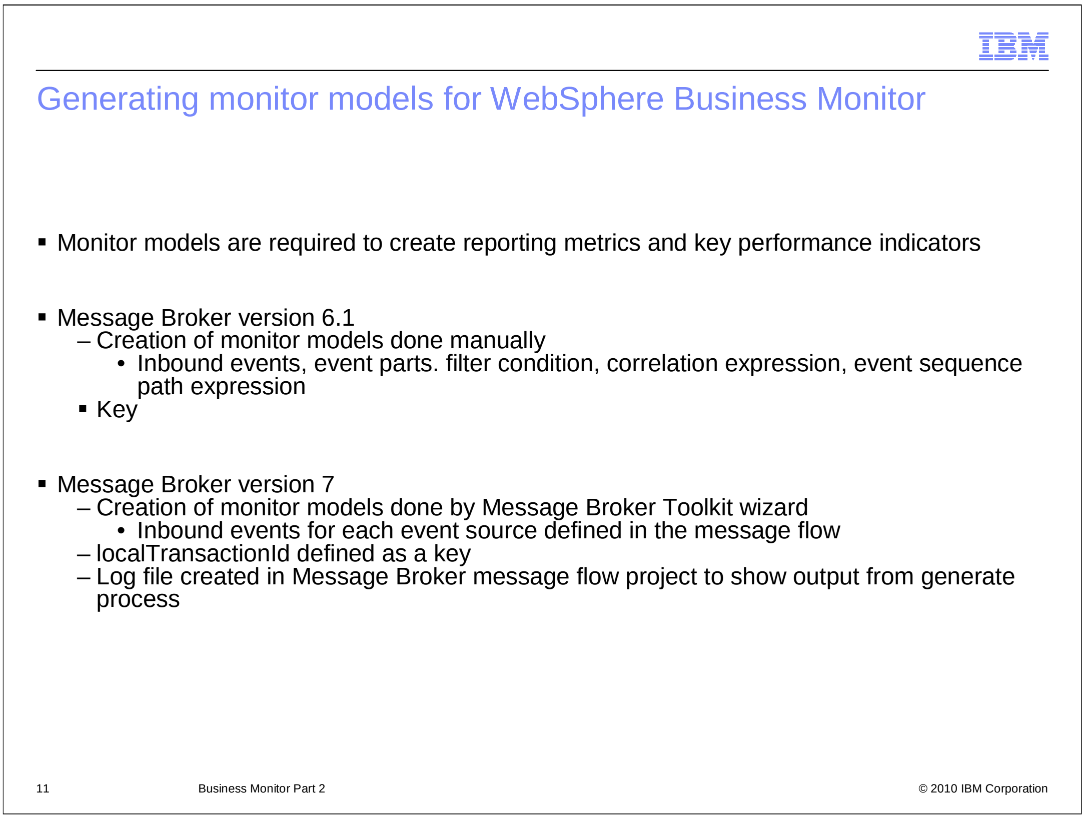
In Message Broker version 6.1, the creation time of the published event is provided as an attribute of the event message. In some situations, this creation time might not be granular enough, and it is possible for two monitoring events to have the same creation time. This creation time is not sufficient fine-grained to use to sequence the events.

Message Broker version 7 has introduced the event sequence counter. This sequence counter is set to 1 at the start of each message flow, and is incremented for each monitoring event in the message flow. At the start of the next flow, t is reset to 1.

With the addition of the counter this provides the necessary level of granularity. The sequence counter is always emitted, with the creation time in the event sequence element of the event. The sequence tab in the monitor wizard has therefore been removed in version 7.

The creation time information can be useful for calculating the duration of a message flow.

The monitor model generation, which is described later in this session, sets the event sequence path to use both fields.



Generating monitor models for WebSphere Business Monitor

- Monitor models are required to create reporting metrics and key performance indicators
- Message Broker version 6.1
  - Creation of monitor models done manually
    - Inbound events, event parts. filter condition, correlation expression, event sequence path expression
  - Key
- Message Broker version 7
  - Creation of monitor models done by Message Broker Toolkit wizard
    - Inbound events for each event source defined in the message flow
  - localTransactionId defined as a key
  - Log file created in Message Broker message flow project to show output from generate process

11 Business Monitor Part 2 © 2010 IBM Corporation

The final enhancement in Message Broker version 7 is the provision of an automated set of tools to create the monitor model in the Toolkit.

To enable WebSphere Business Monitor to capture and report business events and key performance indicators, it is necessary to create a monitor model for each message flow. This is used by the WebSphere Business Monitor toolkit plug-in to define key performance indicators and metrics for the required message flow.

In Message Broker version 6.1, this is a manual task. Knowledge of the format of the monitor model XML schema is required, and this can be error-prone. Several items must be created, such as the descriptions of the required events, filter conditions and correlation expressions.

In Message Broker version 7, the Toolkit has been enhanced to perform the generation of the monitor model automatically.

Inbound events names are taken from the event source address name.

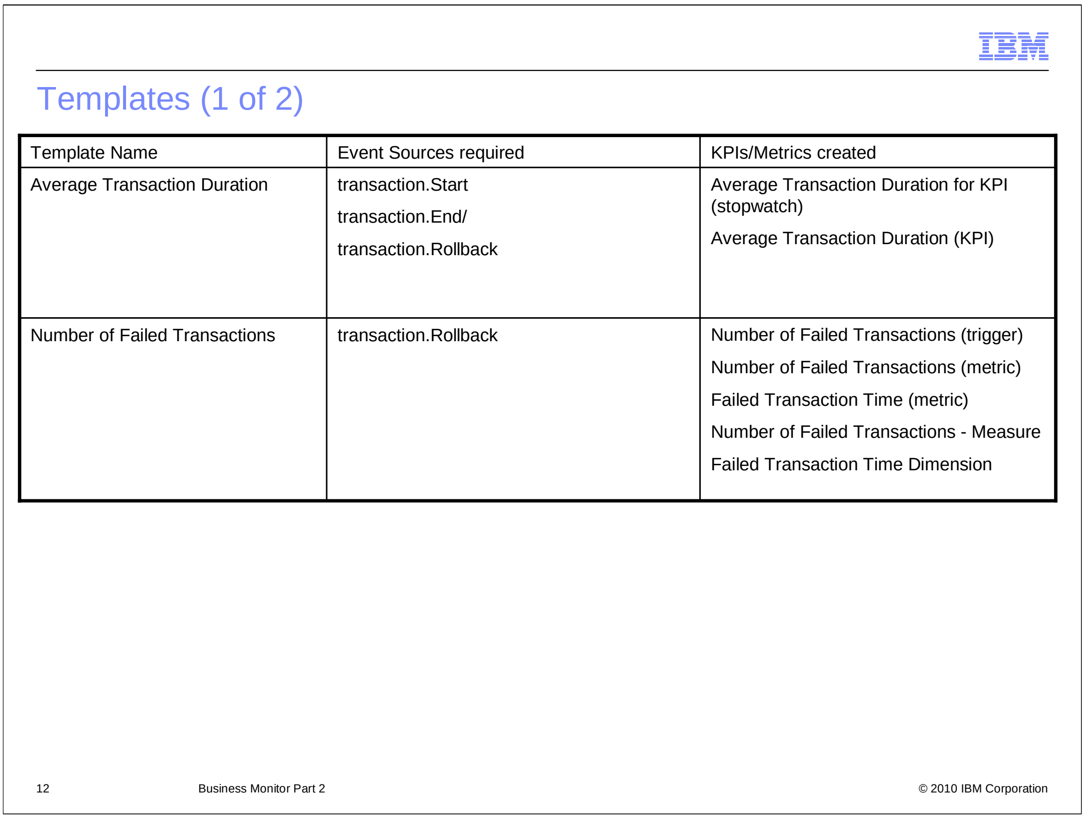
Inbound events for Message Broker transaction event sources are created at the flow level in the model.

Event groups are created for each of the nodes where Message Broker terminal event sources have been defined. The inbound events for these Message Broker terminal events are then stored in the event groups

The “Synchronies with Application” option available in the Business Monitor toolkit can be used to update the created model with changes in the message flow event source definitions.

The correlation, filter expressions and Quality of Service are read-only, and appear grayed out. This is to maintain synchronization with the application so that you can continue to update the monitor model following changes to the message flow.

Finally, the process that creates the monitor model also creates a log file. This is stored in the Message Broker workspace, in the project containing the message flow.



Templates (1 of 2)

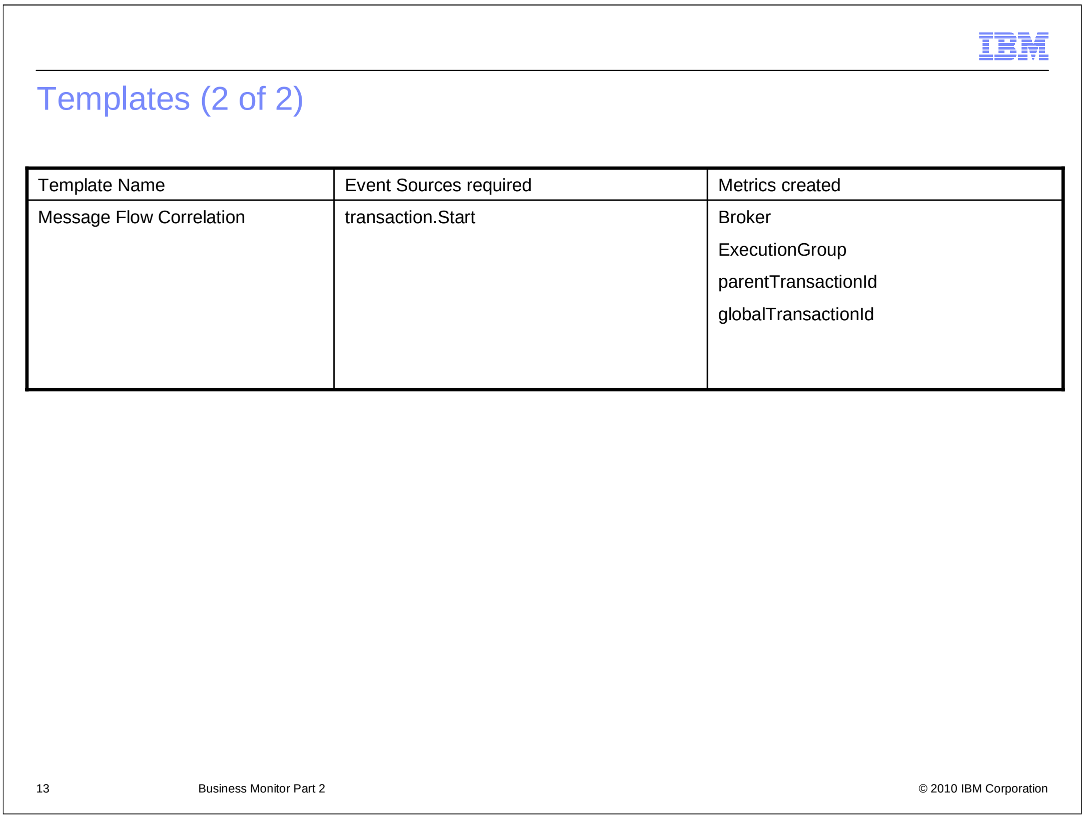
Template Name	Event Sources required	KPIs/Metrics created
Average Transaction Duration	transaction.Start transaction.End/ transaction.Rollback	Average Transaction Duration for KPI (stopwatch) Average Transaction Duration (KPI)
Number of Failed Transactions	transaction.Rollback	Number of Failed Transactions (trigger) Number of Failed Transactions (metric) Failed Transaction Time (metric) Number of Failed Transactions - Measure Failed Transaction Time Dimension

12 Business Monitor Part 2 © 2010 IBM Corporation

Templates are used to define the key performance indicators and metrics that are used for the monitoring of your message flows. Pre-supplied templates are provided which define some basic indicators, and they are shown here and on the next slide. It is probable that you will also need to define your own additional templates to report the specific events in your own applications.

The second column shows the event sources that are required for the particular template that is selected. For example, if you select the template to report the average transaction duration, then you must include the transaction start, end and rollback events.

The third column shows the metrics that are provided if this template is selected. For example, if you select the “Failed Transaction” template, this will report a number of indicators providing information such as the number of failed transactions, and the time of the failed transaction.



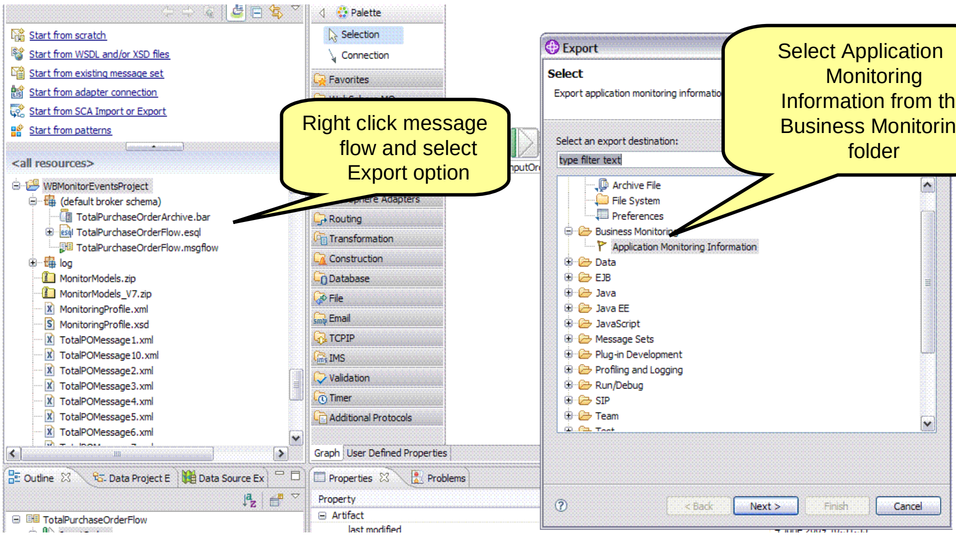
Templates (2 of 2)

Template Name	Event Sources required	Metrics created
Message Flow Correlation	transaction.Start	Broker ExecutionGroup parentTransactionId globalTransactionId

13 Business Monitor Part 2

This template is used to correlate events across multiple message flows. The metrics that are reported relate more to the broker environment, and include information about the broker, execution group, and transactions IDs.

Exporting message flow monitoring information



Right click message flow and select Export option

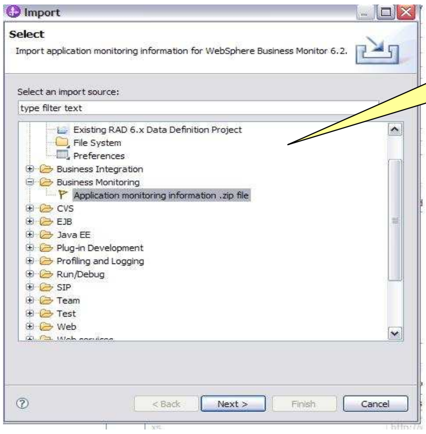
Select Application Monitoring Information from the Business Monitoring folder

14 Business Monitor Part 2 © 2010 IBM Corporation

The next slides show several screen captures, and describe the process of generating a monitor model for a message flow in the Message Broker Toolkit. The output of this process is a ".zip" file, which contains the message flow schema which describes the monitoring requirements for the particular message flow. This is imported into WebSphere Integration Developer to complete the specification of the key performance indicators.

When the message flow has been completed, and the monitoring event points have been defined, save the message flow in the Message Broker Toolkit. Right-click the message flow, and select the "export" option, then expand the "Business Monitoring" folder, as shown by this screen capture. Select "application monitoring information", and click next. Specify the location of the generated ".zip" file, and then click finish.

Import monitoring information into WebSphere Business Monitor toolkit



Import

Select

Import application monitoring information for WebSphere Business Monitor 6.2.

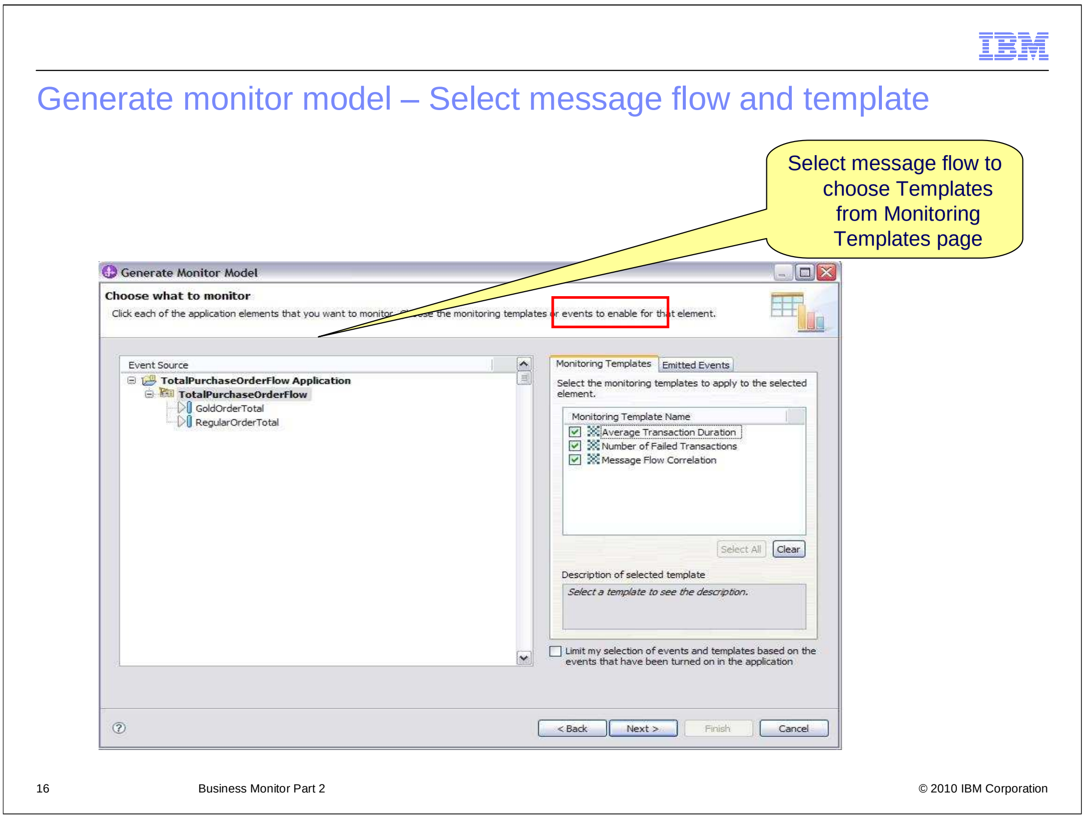
Select an import source:

type filter text

Select Import then Application monitoring information .zip file from Business Monitoring folder

Now switch to the WebSphere Integration Developer Toolkit. This should have the plug-in for WebSphere Business Monitor installed.

Import the “.zip” file into this toolkit in the normal way. When the import window opens, select the business monitoring section, and the application monitoring “.zip” file, as shown here.



Generate monitor model – Select message flow and template

Select message flow to choose Templates from Monitoring Templates page

Generate Monitor Model

Choose what to monitor

Click each of the application elements that you want to monitor. Choose the monitoring templates or events to enable for that element.

Event Source
- TotalPurchaseOrderFlow Application
  - TotalPurchaseOrderFlow
    - GoldOrderTotal
    - RegularOrderTotal

Monitoring Templates | Emitted Events

Select the monitoring templates to apply to the selected element.

Monitoring Template Name
☑ Average Transaction Duration
☑ Number of Failed Transactions
☑ Message Flow Correlation

Description of selected template

Select a template to see the description.

□ Limit my selection of events and templates based on the events that have been turned on in the application

16 Business Monitor Part 2 © 2010 IBM Corporation

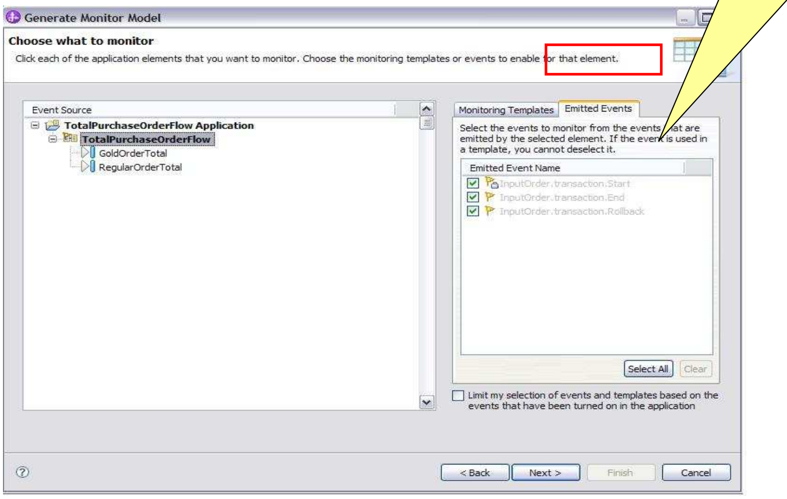
Completion of the this import process will start the “Generate Monitor Model” wizard. On the first page, in the left pane, you will see the name of the message flow that you have imported, and the nodes for which monitoring events have been defined.

In the left pane, click the name of the message flow.

In the right pane, you will see two tabs. The first tab is the list of templates that are available for selection for this monitor model. Select, or de-select these templates as required.

Generate monitor model – Select message flow and emitted events

Select Transaction Events



Generate Monitor Model

Choose what to monitor

Click each of the application elements that you want to monitor. Choose the monitoring templates or events to enable for that element.

Event Source

- ⊟ **TotalPurchaseOrderFlow Application**
  - ⊟ **TotalPurchaseOrderFlow**
    - GoldOrderTotal
    - RegularOrderTotal

Monitoring Templates | Emitted Events

Select the events to monitor from the events that are emitted by the selected element. If the event is used in a template, you cannot deselect it.

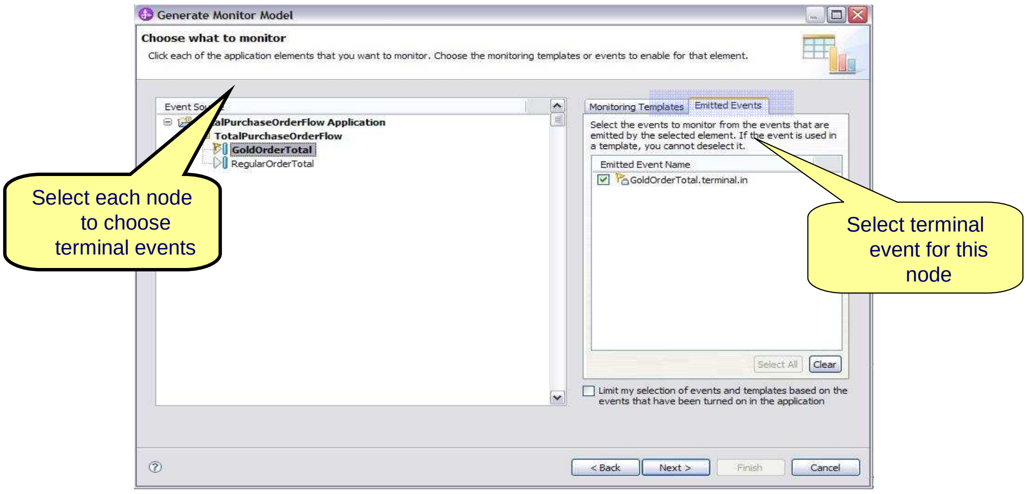
Emitted Event Name

- ☑ InputOrder.transaction.Start
- ☑ InputOrder.transaction.End
- ☑ InputOrder.transaction.Rollback

□ Limit my selection of events and templates based on the events that have been turned on in the application

On the same window, click the “Emitted Events” tab. Here you can select, or de-select, the specific events that are required for this message flow.

Generate monitor model – Select node and specify terminal events



Generate Monitor Model

Choose what to monitor

Click each of the application elements that you want to monitor. Choose the monitoring templates or events to enable for that element.

Event So…e
- …alPurchaseOrderFlow Application
  - TotalPurchaseOrderFlow
    - **GoldOrderTotal**
    - RegularOrderTotal

Monitoring Templates | Emitted Events

Select the events to monitor from the events that are emitted by the selected element. If the event is used in a template, you cannot deselect it.

Emitted Event Name
☑ GoldOrderTotal.terminal.in

☐ Limit my selection of events and templates based on the events that have been turned on in the application

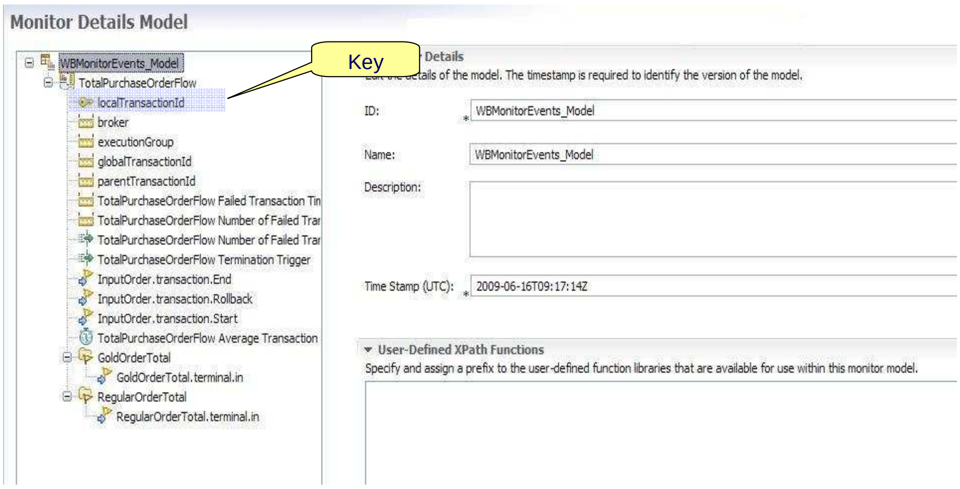
Select each node to choose terminal events

Select terminal event for this node

On the same window, in the left pane, select the specific node which has a monitoring event defined. In the right pane, select the “Emitted Events” tab, and select, or de-select, the specific terminal event for this node.

When all configurations and selections have been made, click next, and finish, to complete generation of the monitor model.

Monitor model - Key



Monitor Details Model

- WBMonitorEvents_Model
  - TotalPurchaseOrderFlow
    - localTransactionId
    - broker
    - executionGroup
    - globalTransactionId
    - parentTransactionId
    - TotalPurchaseOrderFlow Failed Transaction Tin
    - TotalPurchaseOrderFlow Number of Failed Trar
    - TotalPurchaseOrderFlow Number of Failed Trar
    - TotalPurchaseOrderFlow Termination Trigger
    - InputOrder.transaction.End
    - InputOrder.transaction.Rollback
    - InputOrder.transaction.Start
    - TotalPurchaseOrderFlow Average Transaction
    - GoldOrderTotal
      - GoldOrderTotal.terminal.in
    - RegularOrderTotal
      - RegularOrderTotal.terminal.in

Details

tails of the model. The timestamp is required to identify the version of the model.

ID: * WBMonitorEvents_Model

Name: WBMonitorEvents_Model

Description:

Time Stamp (UTC): * 2009-06-16T09:17:14Z

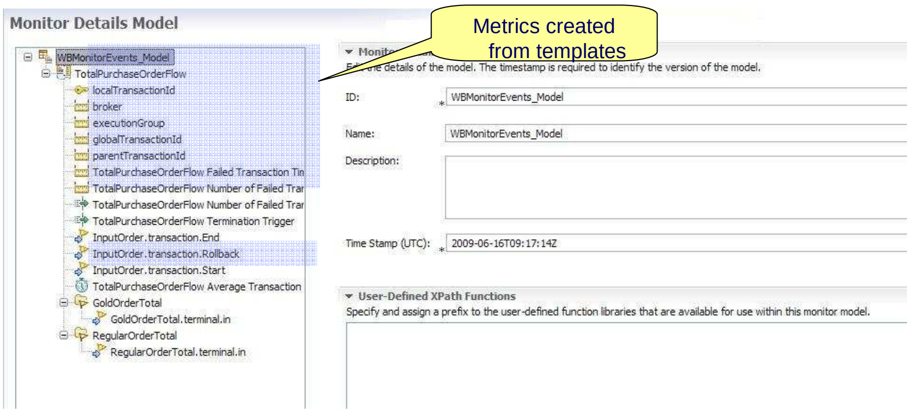
User-Defined XPath Functions

Specify and assign a prefix to the user-defined function libraries that are available for use within this monitor model.

At the completion of this process, the monitor model will have been generated, and you will see the output similar to this screen capture.

Several values in this model have been generated by the monitor model wizard. First, the key of the monitor model has been set to the value of the “local transaction ID” in the message flow.

Monitor model - Metrics



Metrics created from templates

Monitor Details Model

- WBMonitorEvents_Model
  - TotalPurchaseOrderFlow
    - localTransactionId
    - broker
    - executionGroup
    - globalTransactionId
    - parentTransactionId
    - TotalPurchaseOrderFlow Failed Transaction Tin
    - TotalPurchaseOrderFlow Number of Failed Trar
    - TotalPurchaseOrderFlow Number of Failed Trar
    - TotalPurchaseOrderFlow Termination Trigger
    - InputOrder.transaction.End
    - InputOrder.transaction.Rollback
    - InputOrder.transaction.Start
    - TotalPurchaseOrderFlow Average Transaction
    - GoldOrderTotal
      - GoldOrderTotal.terminal.in
    - RegularOrderTotal
      - RegularOrderTotal.terminal.in

▾ Monit

Edit the details of the model. The timestamp is required to identify the version of the model.

ID: * WBMonitorEvents_Model

Name: WBMonitorEvents_Model

Description:

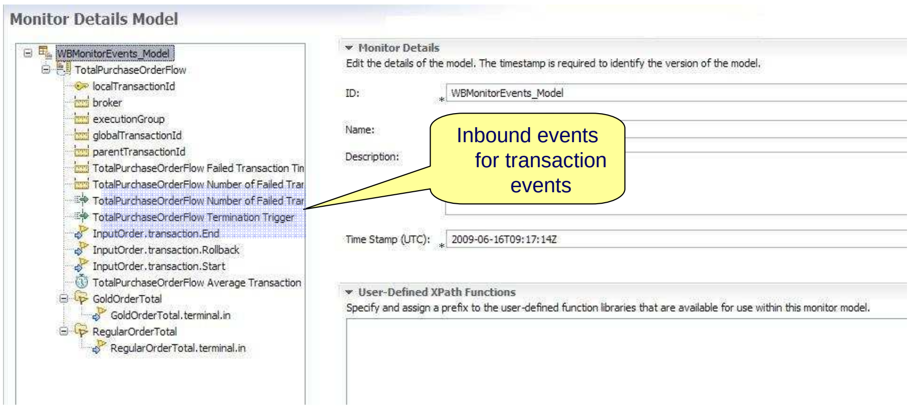
Time Stamp (UTC): * 2009-06-16T09:17:14Z

▾ User-Defined XPath Functions

Specify and assign a prefix to the user-defined function libraries that are available for use within this monitor model.

The model contains several metrics, shown here in blue. These were generated by applying the templates that were shown earlier.

Monitor model – Transaction events



Monitor Details Model

- WBMonitorEvents_Model
  - TotalPurchaseOrderFlow
    - localTransactionId
    - broker
    - executionGroup
    - globalTransactionId
    - parentTransactionId
    - TotalPurchaseOrderFlow Failed Transaction Tin
    - TotalPurchaseOrderFlow Number of Failed Trar
    - TotalPurchaseOrderFlow Number of Failed Trar
    - TotalPurchaseOrderFlow Termination Trigger
    - InputOrder.transaction.End
    - InputOrder.transaction.Rollback
    - InputOrder.transaction.Start
    - TotalPurchaseOrderFlow Average Transaction
    - GoldOrderTotal
      - GoldOrderTotal.terminal.in
    - RegularOrderTotal
      - RegularOrderTotal.terminal.in

Monitor Details

Edit the details of the model. The timestamp is required to identify the version of the model.

ID: * WBMonitorEvents_Model

Name:

Description:

Time Stamp (UTC): * 2009-06-16T09:17:14Z

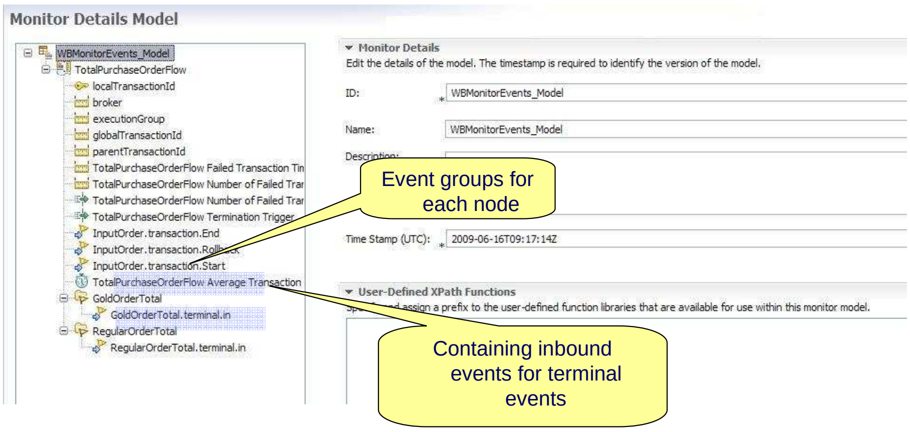
User-Defined XPath Functions

Specify and assign a prefix to the user-defined function libraries that are available for use within this monitor model.

Inbound events for transaction events

The inbound events are shown here. These represent the transaction events in the message flow, which are the transaction start, transaction end and transaction rollback events.

Monitor model – Node groups and terminal events



Monitor Details Model

- WBMonitorEvents_Model
  - TotalPurchaseOrderFlow
    - localTransactionId
    - broker
    - executionGroup
    - globalTransactionId
    - parentTransactionId
    - TotalPurchaseOrderFlow Failed Transaction Tin
    - TotalPurchaseOrderFlow Number of Failed Trar
    - TotalPurchaseOrderFlow Number of Failed Trar
    - TotalPurchaseOrderFlow Termination Trigger
    - InputOrder.transaction.End
    - InputOrder.transaction.Rollback
    - InputOrder.transaction.Start
    - TotalPurchaseOrderFlow Average Transaction
    - GoldOrderTotal
      - GoldOrderTotal.terminal.in
    - RegularOrderTotal
      - RegularOrderTotal.terminal.in

▾ Monitor Details

Edit the details of the model. The timestamp is required to identify the version of the model.

ID: * WBMonitorEvents_Model

Name: WBMonitorEvents_Model

Description:

Time Stamp (UTC): * 2009-06-16T09:17:14Z

▾ User-Defined XPath Functions

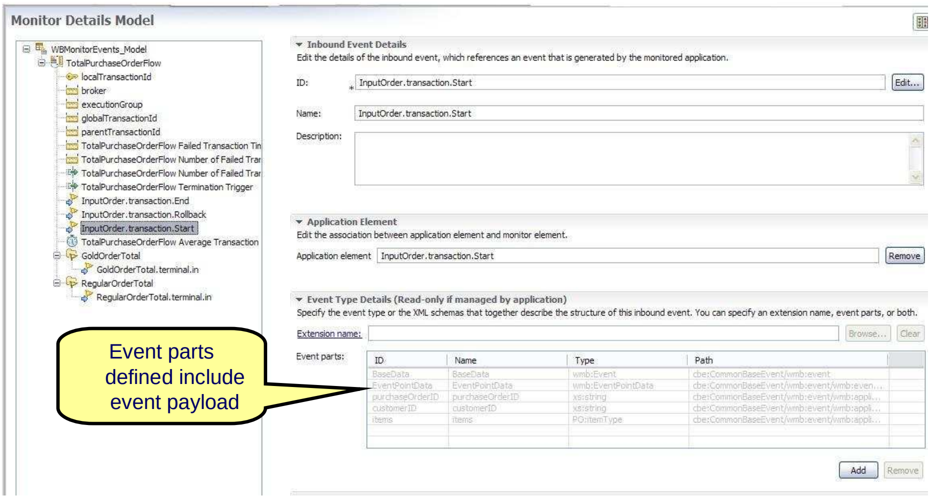
Sp... and assign a prefix to the user-defined function libraries that are available for use within this monitor model.

Event groups for each node

Containing inbound events for terminal events

Next are the entries for the event groups for this message flow model. Each group corresponds to a message node. Expanding each group shows the particular event that has been defined for that node.

Monitor model – Event payload



Monitor Details Model

- WBMonitorEvents_Model
  - TotalPurchaseOrderFlow
    - localTransactionId
    - broker
    - executionGroup
    - globalTransactionId
    - parentTransactionId
    - TotalPurchaseOrderFlow Failed Transaction Tin
    - TotalPurchaseOrderFlow Number of Failed Trar
    - TotalPurchaseOrderFlow Number of Failed Trar
    - TotalPurchaseOrderFlow Termination Trigger
    - InputOrder.transaction.End
    - InputOrder.transaction.Rollback
    - InputOrder.transaction.Start
    - TotalPurchaseOrderFlow Average Transaction
    - GoldOrderTotal
      - GoldOrderTotal.terminal.in
    - RegularOrderTotal
      - RegularOrderTotal.terminal.in

Inbound Event Details

Edit the details of the inbound event, which references an event that is generated by the monitored application.

ID: * InputOrder.transaction.Start [Edit...]

Name: InputOrder.transaction.Start

Description:

Application Element

Edit the association between application element and monitor element.

Application element InputOrder.transaction.Start [Remove]

Event Type Details (Read-only if managed by application)

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: [Browse...] [Clear]

Event parts:

ID	Name	Type	Path
BaseData	BaseData	wmb:Event	cbe:CommonBaseEvent/wmb:event
EventPointData	EventPointData	wmb:EventPointData	cbe:CommonBaseEvent/wmb:event/wmb:even...
purchaseOrderID	purchaseOrderID	xs:string	cbe:CommonBaseEvent/wmb:event/wmb:appli...
customerID	customerID	xs:string	cbe:CommonBaseEvent/wmb:event/wmb:appli...
items	items	PO:itemType	cbe:CommonBaseEvent/wmb:event/wmb:appli...

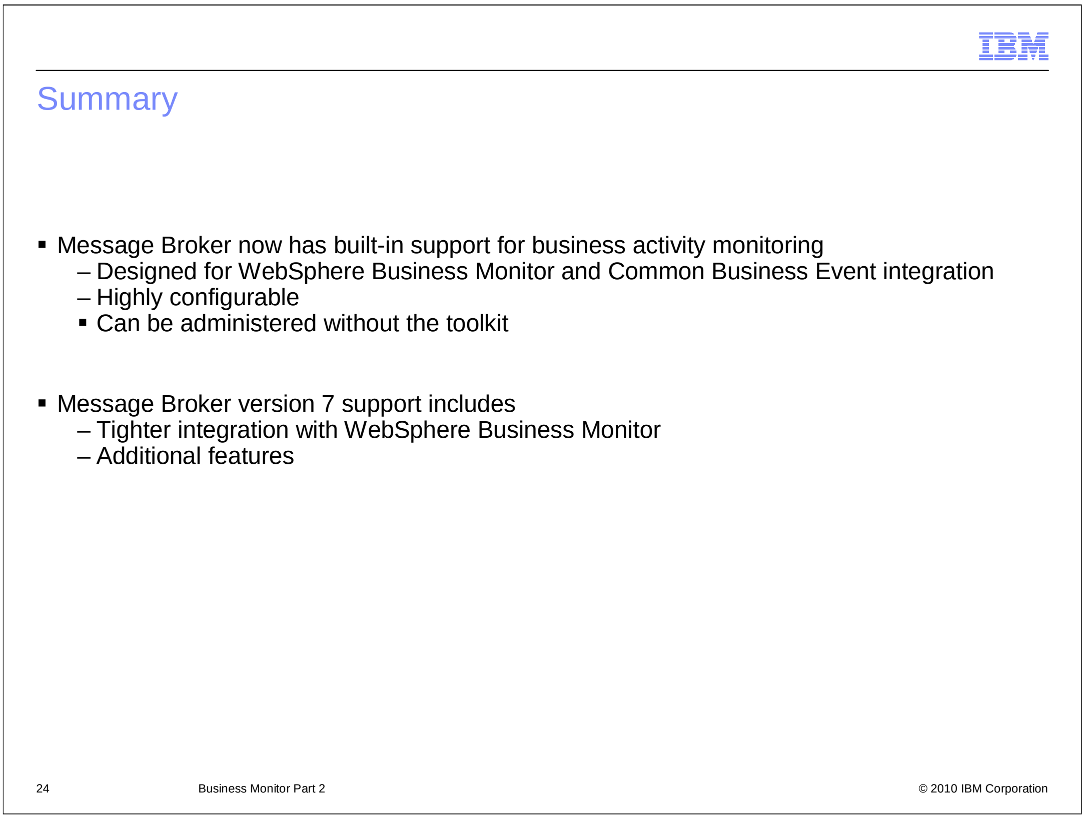
[Add] [Remove]

Event parts defined include event payload

Finally, the right pane shows the specific information about the monitoring event payload, and the specific fields that are produced.

Click a specific item in the navigator, and the event payload information is shown. In this example, the “transaction start” event has been selected, and the event payload information is shown in the right pane.

This completes the definition of the monitor model. You might require to define additional key performance indicators, and perform additional tailoring with the WebSphere Business Space application. This is described in the IBM Education Assistant modules, in the section describing WebSphere Business Monitor.



Summary

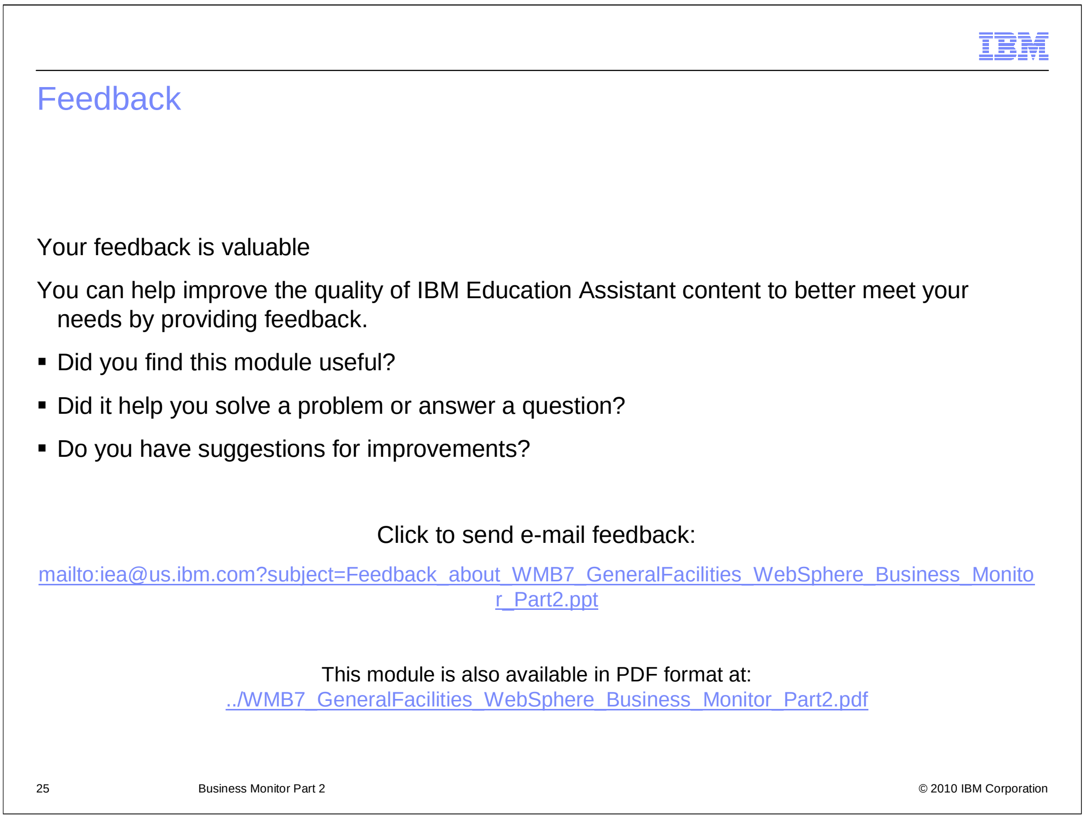
- Message Broker now has built-in support for business activity monitoring
  - Designed for WebSphere Business Monitor and Common Business Event integration
  - Highly configurable
  - Can be administered without the toolkit

- Message Broker version 7 support includes
  - Tighter integration with WebSphere Business Monitor
  - Additional features

24 Business Monitor Part 2 © 2010 IBM Corporation

In summary, WebSphere Message Broker now has built-in support for business monitoring. This support published monitoring events that can be consumed by any business monitoring reporting tool. This can be configured using the Message Broker Toolkit, or just with the command interface.

There are specific extensions in Message Broker version 7 for WebSphere Business Monitor. In particular, the automatic generation of the monitor models enables the monitoring tools to be configured easily for Message Broker.



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?

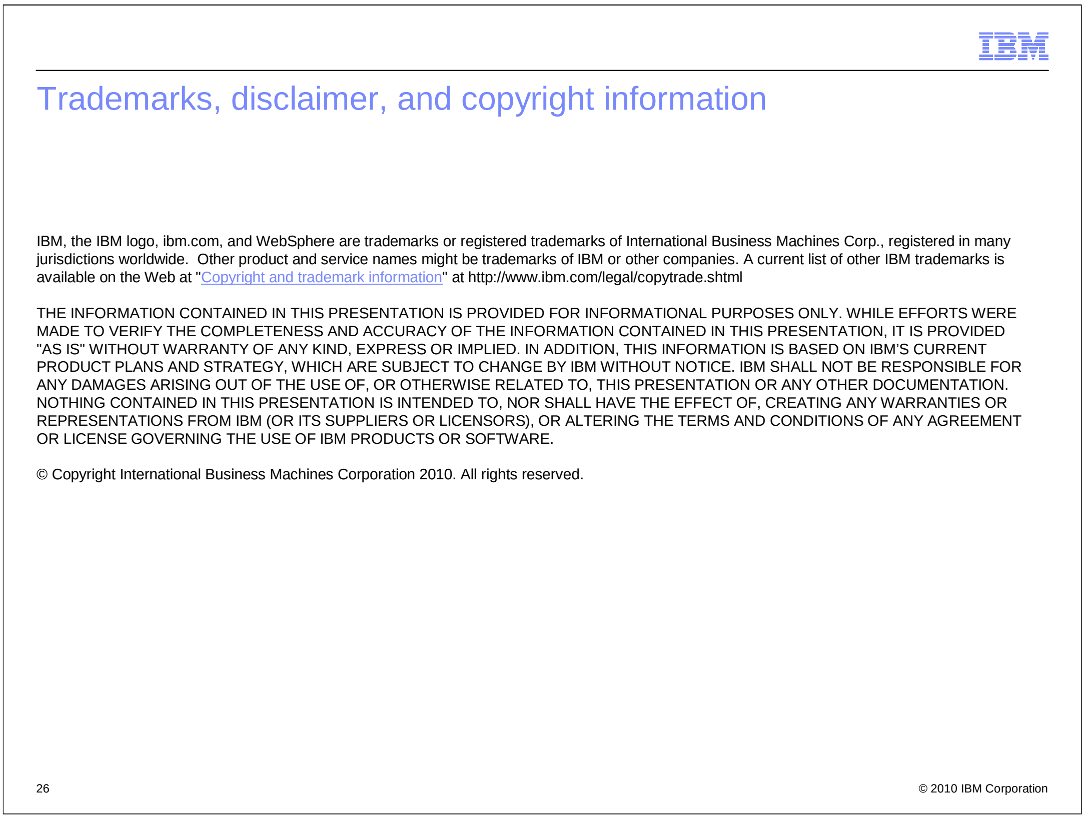
Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WMB7_GeneralFacilities_WebSphere_Business_Monitor_Part2.ppt

This module is also available in PDF format at:
../WMB7_GeneralFacilities_WebSphere_Business_Monitor_Part2.pdf

25 Business Monitor Part 2

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



Trademarks, disclaimer, and copyright information

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.