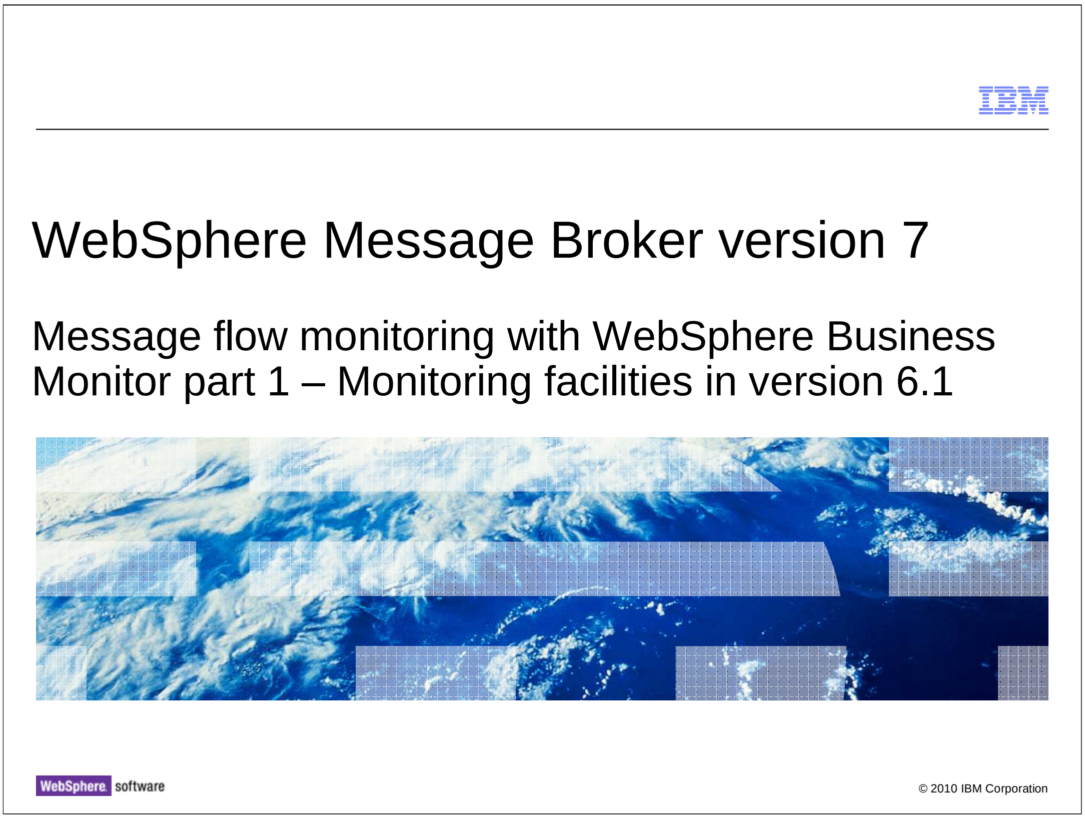
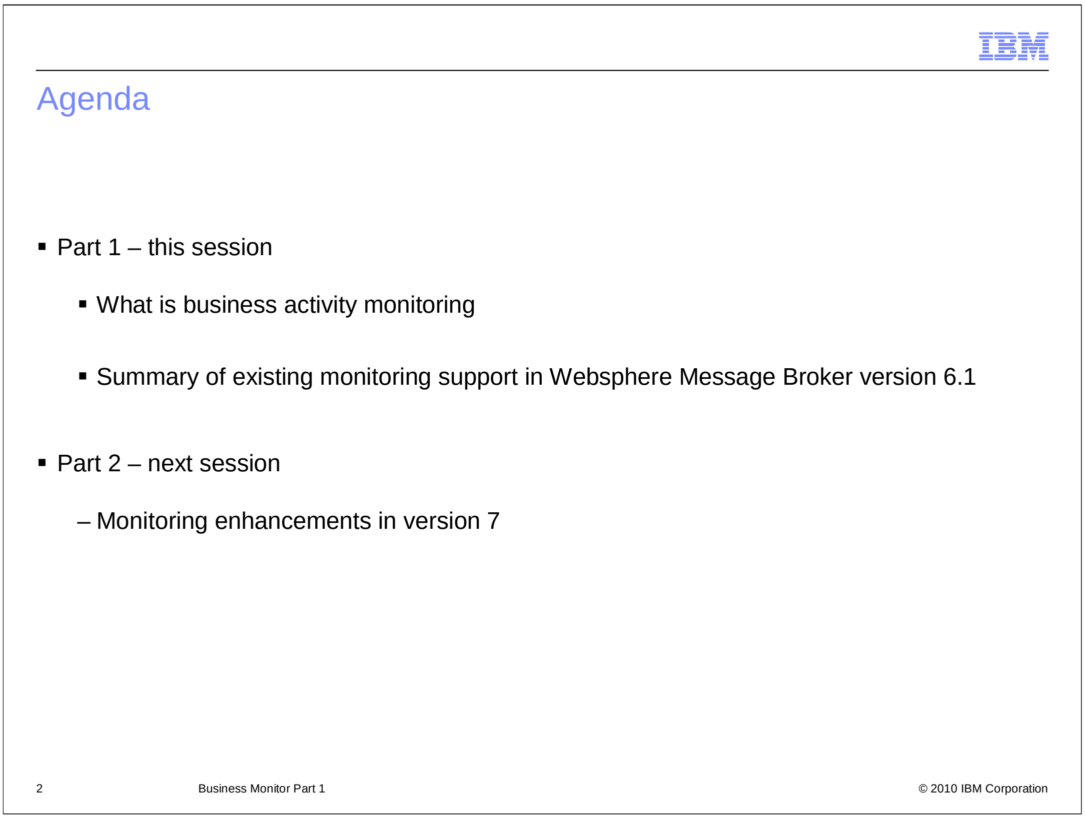


WebSphere Message Broker version 7

Message flow monitoring with WebSphere Business Monitor part 1 – Monitoring facilities in version 6.1

WebSphere software

This session is part 1 of the topic covering the integration with WebSphere Business Monitor. This part covers the existing facilities for business monitoring in WebSphere Message Broker version 6.1. The second part, in the next IBM Education Assistant session, covers the changes that have been made to this function in Message Broker version 7.



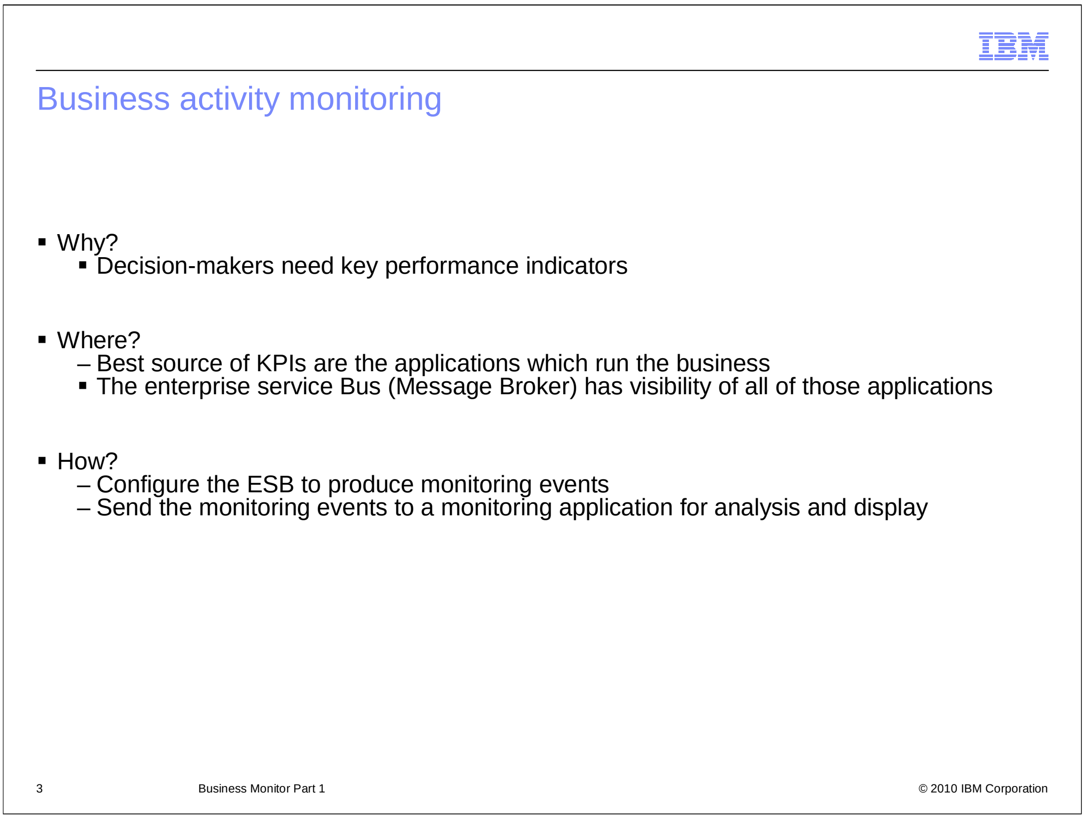
Agenda

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

2 Business Monitor Part 1 © 2010 IBM Corporation

WebSphere Message Broker introduced support for business monitoring in version 6.1. The capability was enhanced over several fixpacks in this version, and this session will describe this capability. This capability enables message flows deployed in Message Broker version 6.1 to emit events, and to have these events correlated and displayed by WebSphere Business Monitor.

The next session covers the enhancements that have been made to this function in Message Broker version 7, how these events are configured in the broker Toolkit, and how to configure the Business Monitor tools.



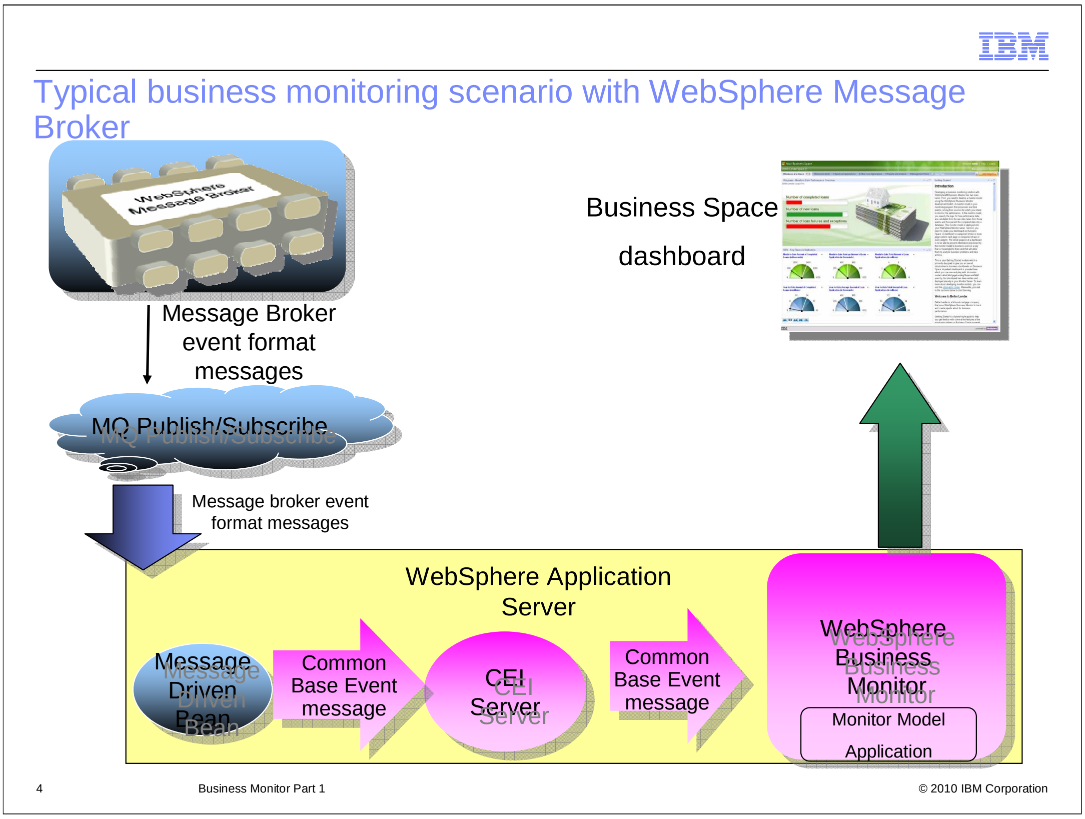
Business activity monitoring

- Why?
  - Decision-makers need key performance indicators
- Where?
  – Best source of KPIs are the applications which run the business
  - The enterprise service Bus (Message Broker) has visibility of all of those applications
- How?
  – Configure the ESB to produce monitoring events
  – Send the monitoring events to a monitoring application for analysis and display

3 Business Monitor Part 1 © 2010 IBM Corporation

The main usage of the monitoring capability within WebSphere Message Broker is business activity monitoring. This type of monitoring is designed to allow business decisions to be made, on the basis of information provided and collected by the monitoring system. Many components of a system can contribute to the data collected by the monitoring system, and data provided by Message Broker might be only a part of the complete monitoring picture.

For example, a business monitoring system for a WebSphere Commerce system might report on the number of orders placed over a certain period of time, or extract orders over a certain value for special authorization.



Typical business monitoring scenario with WebSphere Message Broker

Message Broker event format messages

MQ Publish/Subscribe

Message broker event format messages

Business Space dashboard

WebSphere Application Server

Message Driven Bean

Common Base Event message

CEI Server

Common Base Event message

WebSphere Business Monitor

Monitor Model Application

4 Business Monitor Part 1 © 2010 IBM Corporation

This slide shows a high-level view of how the primary components of this type of monitoring solution fit together.

Business events are published by the Message Broker application in the top left. These events are published using a publish/subscribe architecture with MQ.

A message-driven bean in WebSphere Application Server subscribes to the appropriate topic, and receives these published events. This requires the Application Server to have access to an MQ queue manager.

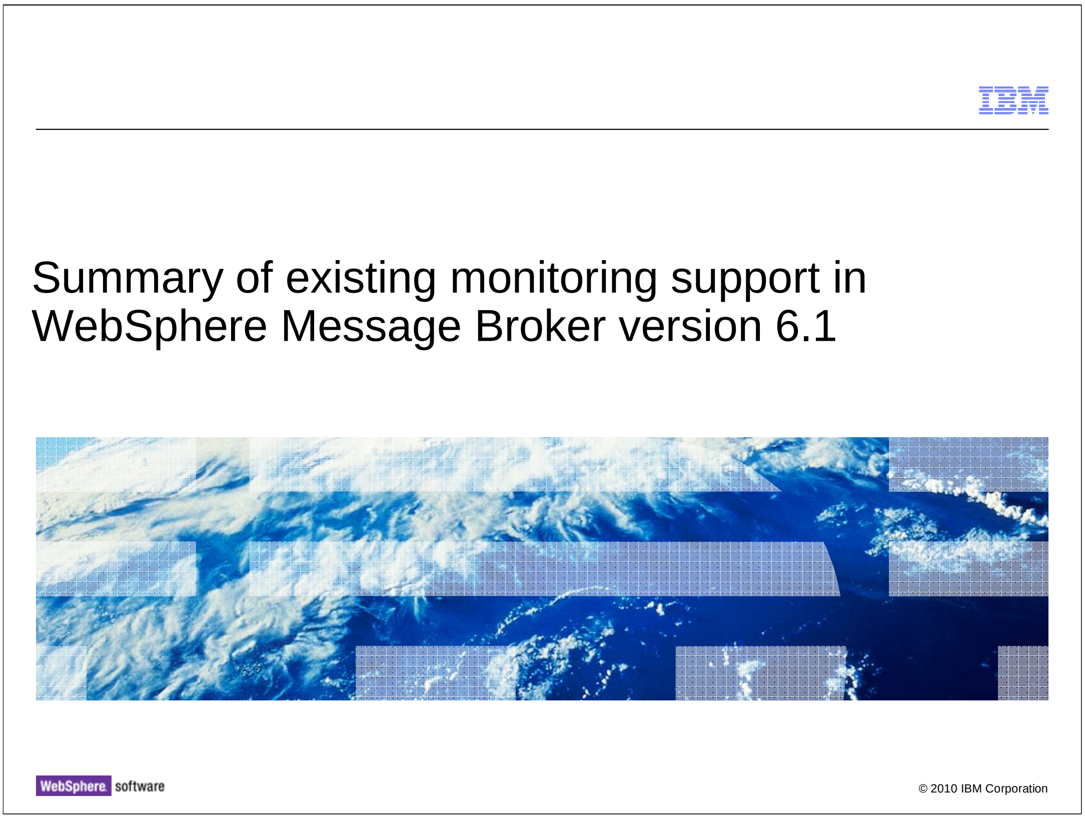
The message-driven bean formats the event message into the Common Business Event format.

This event message is then sent to WebSphere Business Monitor, and the information is used to define and display the required key performance indicators on the Business Space Dashboard, using a Web browser interface.

Events are published to an MQ topic to allow multiple subscribers, and to allow each subscriber to choose the level of granularity. This can be the broker domain, the broker itself, and execution group, or a message flow.

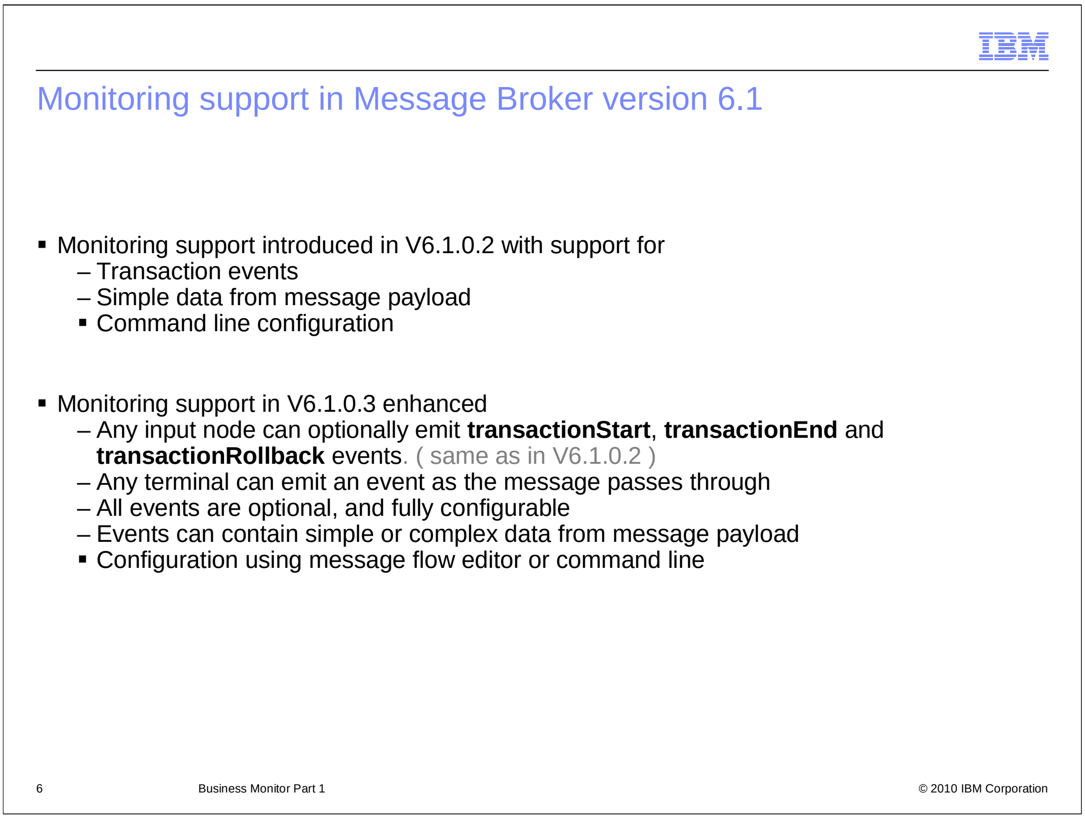
The format of the event message is defined by Message Broker and is compatible with the Common Base Event format. This allows Message Broker to integrate with other monitoring applications, which might have a different message format. It also allows the entire message to be captured and logged to a database for audit purposes.

The message-driven bean is provided with the monitoring sample in Message Broker. It is however fully supported.



Summary of existing monitoring support in WebSphere Message Broker version 6.1

This section describes the support for business monitoring that is already provided in WebSphere Message Broker version 6.1.



Monitoring support in Message Broker version 6.1

- Monitoring support introduced in V6.1.0.2 with support for
  - Transaction events
  - Simple data from message payload
  - Command line configuration

- Monitoring support in V6.1.0.3 enhanced
  - Any input node can optionally emit **transactionStart**, **transactionEnd** and **transactionRollback** events. (same as in V6.1.0.2)
  - Any terminal can emit an event as the message passes through
  - All events are optional, and fully configurable
  - Events can contain simple or complex data from message payload
  - Configuration using message flow editor or command line

6 Business Monitor Part 1 © 2010 IBM Corporation

Support for business monitoring was introduced in Message Broker version 6.1.0.2, or fixpack 2. At this stage, the monitoring functions enabled you to produce an event for a transaction event. An event was published when a transaction, or message flow, started, and when it ended. An event was also emitted if the message flow terminated abnormally, and performed a rollback.

The published message was the whole message payload, and this was configured using command line configuration.

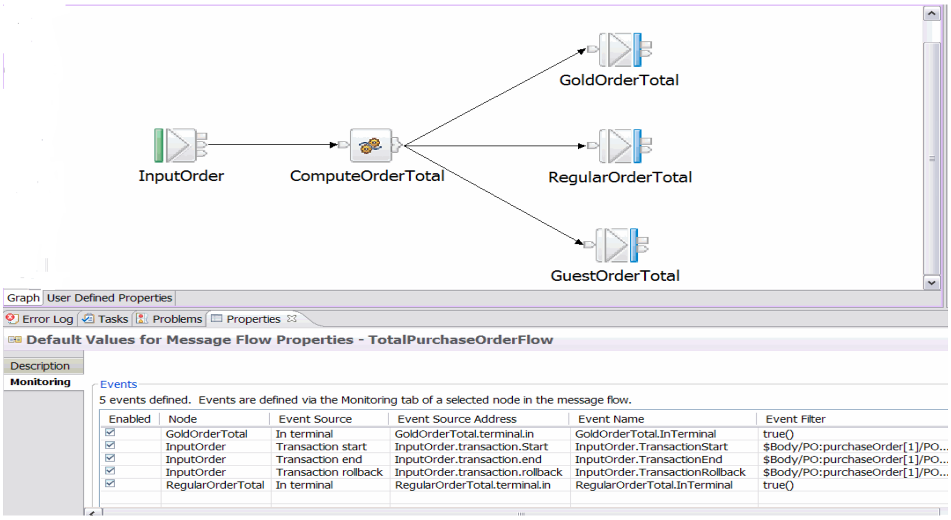
In Message Broker version 6.1.0.3, support was extended to “Terminal Events”. This enabled an event to be associated with any terminal on any node, so event messages can be published at many points throughout a message flow.

Events for error conditions should be configured on the nodes attached to the failure or catch terminals of the input node.

The event message formats can be complex, and you can specify any individual parts of the message tree to be included within the published event message.

The configuration of this monitoring is designed to be administered without the Message Broker Toolkit. The command-line interface and monitoring profile can do all the configuration that the toolkit can do, and the toolkit configuration can be exported as a monitoring profile to ease the command line interface.

Toolkit representation – Summary of events for message flow



Default Values for Message Flow Properties - TotalPurchaseOrderFlow

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	$Body/PO:purchaseOrder[1]/PO...
☑	InputOrder	Transaction end	InputOrder.transaction.end	InputOrder.TransactionEnd	$Body/PO:purchaseOrder[1]/PO...
☑	InputOrder	Transaction rollback	InputOrder.transaction.rollback	InputOrder.TransactionRollback	$Body/PO:purchaseOrder[1]/PO...
☑	RegularOrderTotal	In terminal	RegularOrderTotal.terminal.in	RegularOrderTotal.InTerminal	true()

7 Business Monitor Part 1 © 2010 IBM Corporation

This screen capture shows a message flow that has been defined to emit monitoring events.

If you click the message flow editor pane, rather than an individual node, you can see the properties of the message flow. Click the Monitoring tab, and you will see properties similar to those shown in this example.

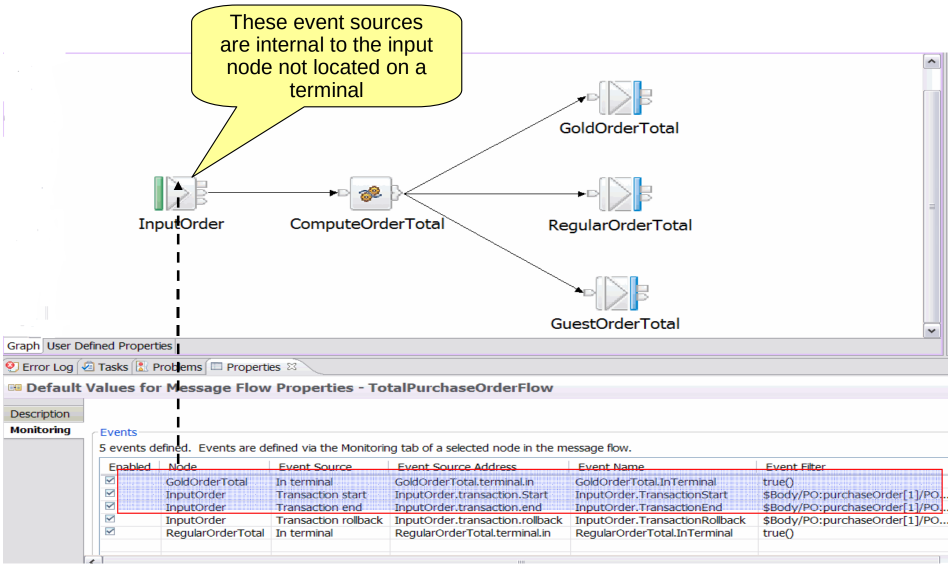
This will display all the monitoring events that have been defined for this message flow, and is a useful place to see a summary of these events.

Clicking an individual node shows the event sources for that node.

The “Event Source Address” column is used to address an event source from the command line, or from a monitoring profile. It is unique within a message flow, provided that the flow does not contain duplicate node names.

The “Enabled” column can be used to enable or disable the emission of events. The default value is for all events to be enabled, and these boxes will be ticked. Note that the emission of events can also be over-ridden at runtime, using a message broker monitoring command .

Monitoring events for input node



Default Values for Message Flow Properties - TotalPurchaseOrderFlow

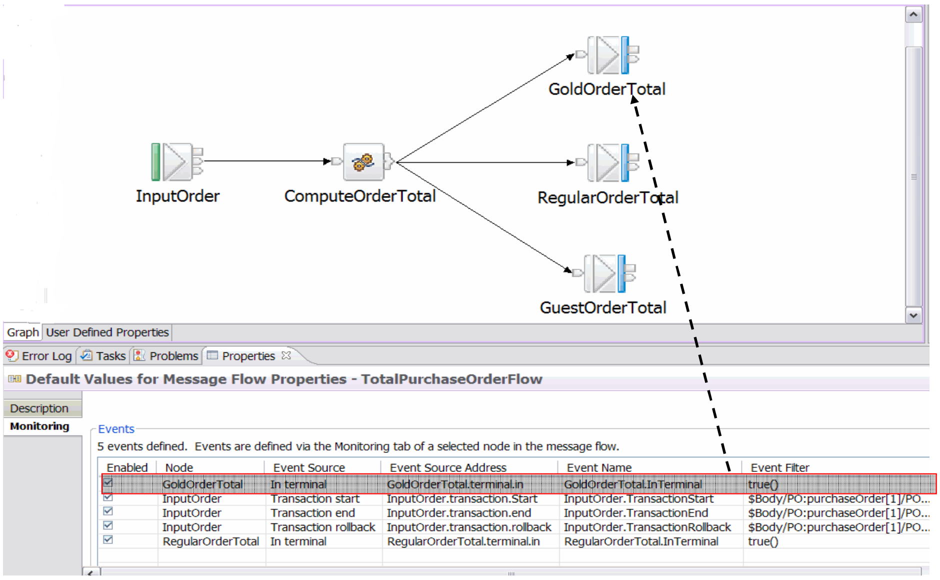
Events

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	$Body/PO:purchaseOrder[1]/PO...
☑	InputOrder	Transaction end	InputOrder.transaction.end	InputOrder.TransactionEnd	$Body/PO:purchaseOrder[1]/PO...
☑	InputOrder	Transaction rollback	InputOrder.transaction.rollback	InputOrder.TransactionRollback	$Body/PO:purchaseOrder[1]/PO...
☑	RegularOrderTotal	In terminal	RegularOrderTotal.terminal.in	RegularOrderTotal.InTerminal	true()

The three highlighted monitoring events are the events that have been defined for the InputOrder node, which is an MQ input node. These events represent the transaction start, transaction end, and transaction rollback points. These are not terminal events, but relate to the internal processing of the input node.

Monitoring event for gold order node



Default Values for Message Flow Properties - TotalPurchaseOrderFlow

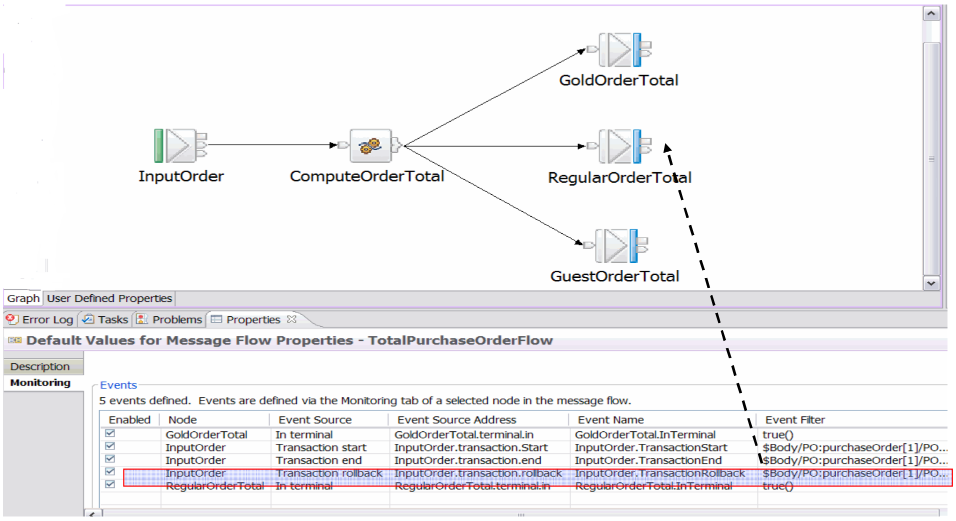
Events

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	$Body/PO:purchaseOrder[1]/PO...
☑	InputOrder	Transaction end	InputOrder.transaction.end	InputOrder.TransactionEnd	$Body/PO:purchaseOrder[1]/PO...
☑	InputOrder	Transaction rollback	InputOrder.transaction.rollback	InputOrder.TransactionRollback	$Body/PO:purchaseOrder[1]/PO...
☑	RegularOrderTotal	In terminal	RegularOrderTotal.terminal.in	RegularOrderTotal.InTerminal	true()

This highlighted monitoring event is the event that has been defined for the “Gold Order Total” node. This is a terminal event, and is defined to emit an event when the message flow execution passes through the “in terminal” of the Gold Order node.

Monitoring event for regular order node



Default Values for Message Flow Properties - TotalPurchaseOrderFlow

Events

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

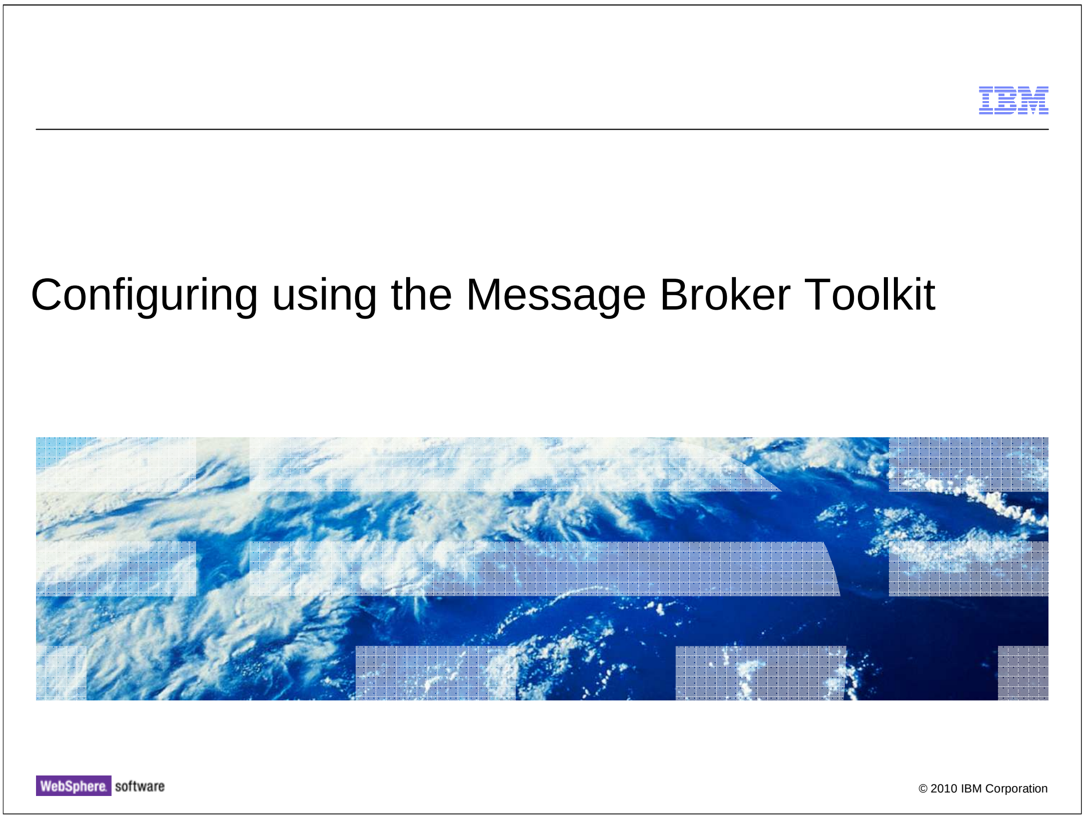
Enabled	Node	Event Source	Event Source Address	Event Name	Event Filter
☑	GoldOrderTotal	In terminal	GoldOrderTotal.terminal.in	GoldOrderTotal.InTerminal	true()
☑	InputOrder	Transaction start	InputOrder.transaction.Start	InputOrder.TransactionStart	$Body/PO:purchaseOrder[1]/PO...
☑	InputOrder	Transaction end	InputOrder.transaction.end	InputOrder.TransactionEnd	$Body/PO:purchaseOrder[1]/PO...
☑	InputOrder	Transaction rollback	InputOrder.transaction.rollback	InputOrder.TransactionRollback	$Body/PO:purchaseOrder[1]/PO...
☑	RegularOrderTotal	In terminal	RegularOrderTotal.terminal.in	RegularOrderTotal.InTerminal	true()

And finally, this highlighted monitoring event is the event that has been defined for the “Regular Order Total” node. This is another terminal event, and is defined to emit an event when the message flow execution passes through the “in terminal” of the Regular Order node.

When this message flow runs, if the flow processes a message successfully, then you can expect to see three published events. These are the transaction start event, a terminal event for either the gold or regular customer, and a transaction end event.

If the request is from a guest customer, then this will not produce a terminal event, so only the transaction start and end events will be published.

If the message flow fails to process the message correctly, a transaction start and transaction rollback events will be published.

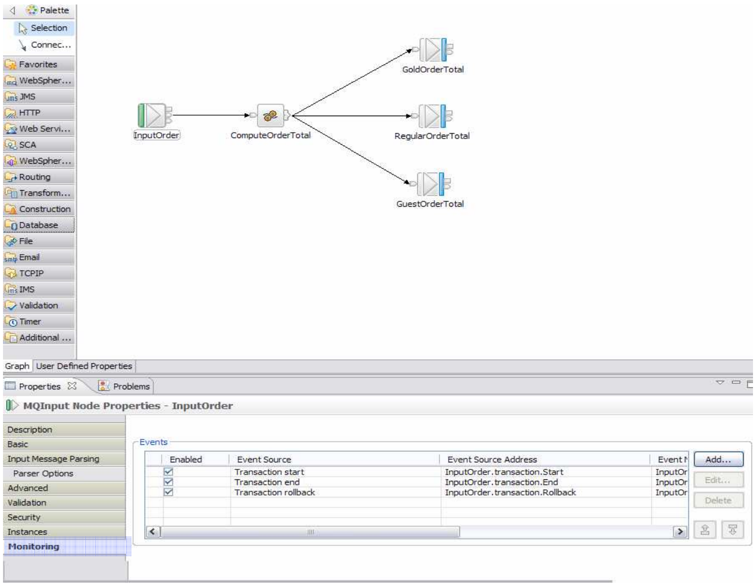


Configuring using the Message Broker Toolkit

WebSphere software

This section will describe how to define and configure monitoring events for message flows, and specific nodes within the message flow.

Node properties – the Monitoring tab



MQInput Node Properties - InputOrder

Events

Enabled	Event Source	Event Source Address	Event N
☑	Transaction start	InputOrder.transaction.Start	InputOr
☑	Transaction end	InputOrder.transaction.End	InputOr
☑	Transaction rollback	InputOrder.transaction.Rollback	InputOr

Monitoring tab available on selected node properties

12 Business Monitor Part 1 © 2010 IBM Corporation

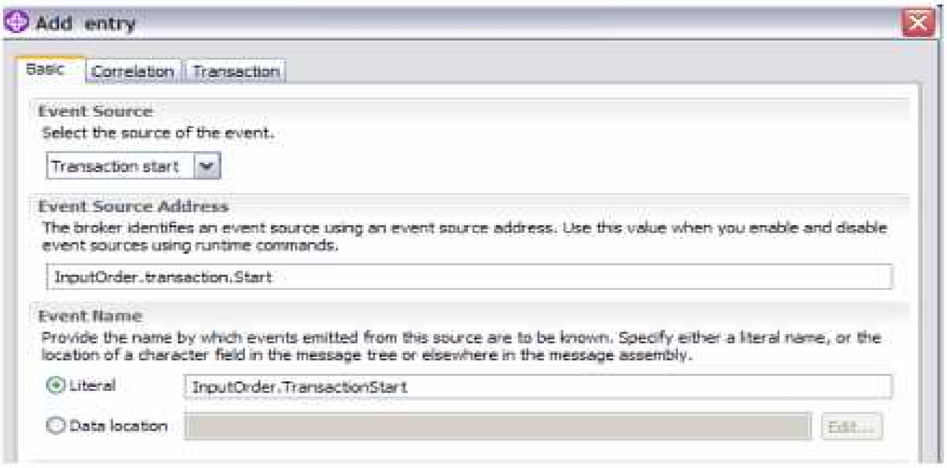
To define a monitoring event point for a particular node, click the required node in the message flow editor. This will show the properties for the selected node.

Then click the Monitoring tab, which is typically towards the bottom of the available tabs in the node properties. This will enable you to define a new monitoring event, and will display any existing events that have been defined for this node.

In the example shown here, the properties of the Input Order node are displayed, which show that there are events defined for transaction start, transaction end and transaction rollback.

To add a new event to this node, click the Add button. To modify or delete an existing event, select the required event by clicking directly on the event, and click Edit or Delete as required. As before, these events can be enabled or disable in the Toolkit, or can be controlled using the command interface in the running broker.

Adding an event to a node



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

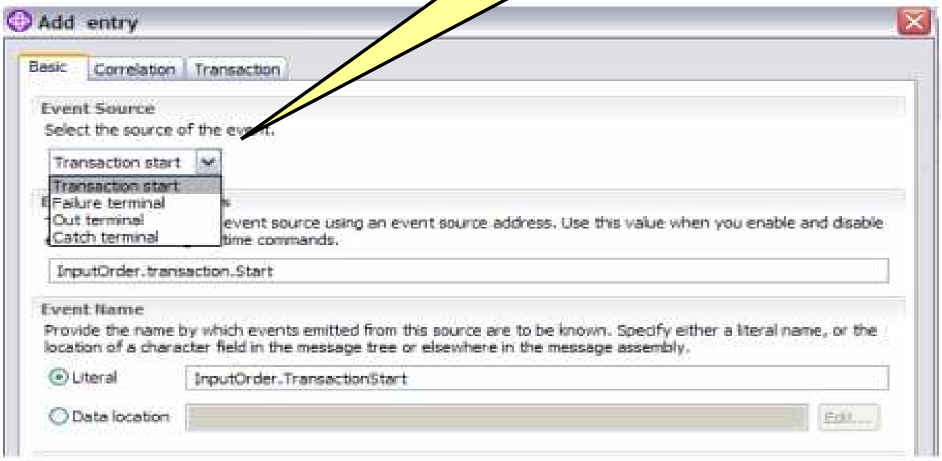
○ Data location

To define a new monitoring event point, click the Add button. This will open a new wizard dialogue, which will look similar to the screen capture shown here.

To edit an existing monitoring event, click Edit. This will open the same dialogue, but the event values that were defined for this event are shown in the appropriate fields in the dialogue.

Select the event type

Select the event source. Input nodes include the special ‘transaction’ event sources


Add entry

Basic | Correlation | Transaction

Event Source

Select the source of the event.

Transaction start

- Transaction start
- Failure terminal
- Out terminal
- Catch terminal

event source using an event source address. Use this value when you enable and disable time 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.

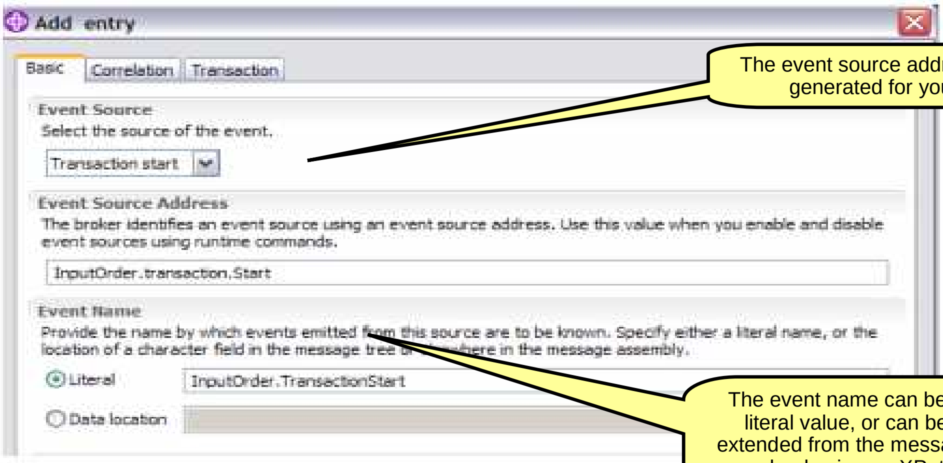
● Literal InputOrder.TransactionStart

○ Data location

Edit...

The first task is to select the type of event monitoring point that is required. The pull-down menu shows the available values for the event type for the particular node. If the node is an input terminal, then the available sources will also include a transaction start event. If selected this will automatically include the transaction end and rollback events.

Specify the event name



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

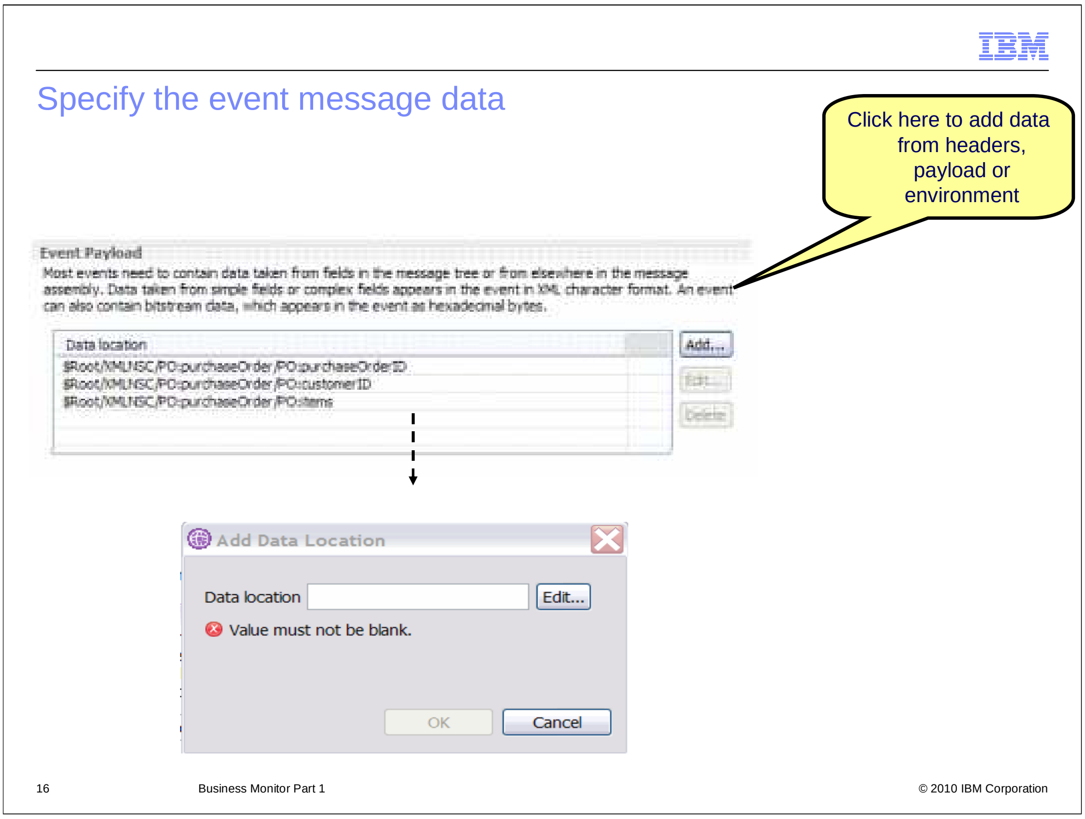
○ Data location

The event source address is generated for you

The event name can be a literal value, or can be extended from the message payload using an XPath expression

The event source address is generated, based on your choice of event type. In this example, the event source address is derived from the transaction start event. This is used by Message Broker to correlate different events in the message flow.

The event name can be specified directly, or can be left to default. The Toolkit provides a suggested value, which can be changed to a literal value of your choice, using the Literal radio button. Alternatively, you can select the “Data location” radio button, and use an X-Path expression to select a value from within the message tree.



Specify the event message data

Click here to add data from headers, payload or environment

Event Payload

Most events need to contain data taken from fields in the message tree or from elsewhere in the message assembly. Data taken from simple fields or complex fields appears in the event in XML character format. An event can also contain bitstream data, which appears in the event as hexadecimal bytes.

Data location
$Root/XMLNSC/PO:purchaseOrder/PO:purchaseOrderID
$Root/XMLNSC/PO:purchaseOrder/PO:customerID
$Root/XMLNSC/PO:purchaseOrder/PO:items

Add Data Location

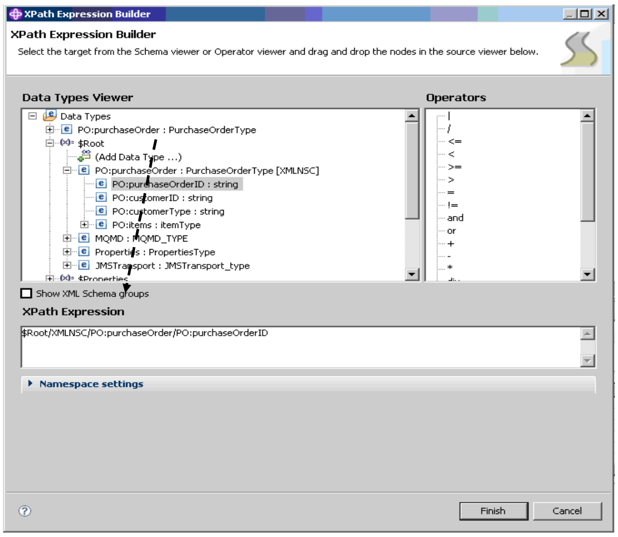
Data location

Value must not be blank.

The next task is to specify is the message content to be contained within this event. This is known as the event payload. To add data definitions to the event message, click the Add button in the event payload section, as shown in this example. This will open a dialogue window where you can specify the location of the data, within the message tree. This dialogue is the same type of dialogue that is used in many other message flow node types.

Click Edit to start the dialogue to specify the event payload.

Use XPath expression builder to specify event payload



XPath Expression Builder

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

Data Types Viewer

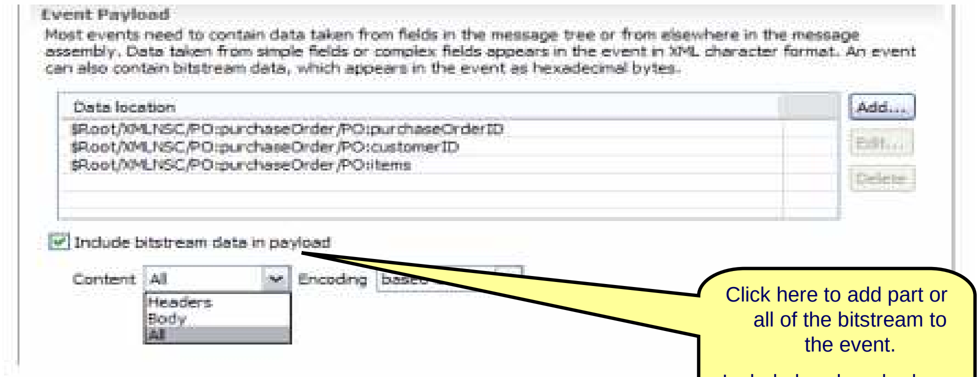
XPath Expression

$Root/XMLNSC/PO:purchaseOrder/PO:purchaseOrderID

17 Business Monitor Part 1 © 2010 IBM Corporation

The XPath expression builder dialogue window is shown. This is the same dialogue that is used to specify XPath expressions for other nodes, such as the database retrieve node, or the Route node. This example shows a field called “purchase Order ID” is going to be included in the event message data.

Customizing an event – Bitstream data



Event Payload

Most events need to contain data taken from fields in the message tree or from elsewhere in the message assembly. Data taken from simple fields or complex fields appears in the event in XML character format. An event can also contain bitstream data, which appears in the event as hexadecimal bytes.

Data location
$Root/XMLNSC/PO:purchaseOrder/PO:purchaseOrderID
$Root/XMLNSC/PO:purchaseOrder/PO:customerID
$Root/XMLNSC/PO:purchaseOrder/PO:items

☑ Include bitstream data in payload

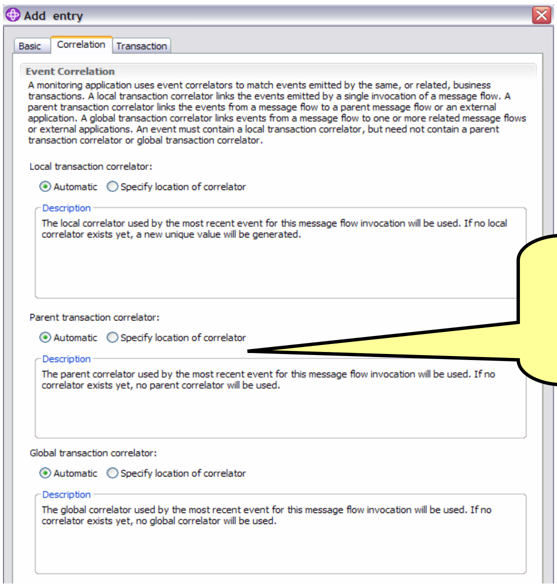
Content: All — Encoding: base

Click here to add part or all of the bitstream to the event.

Include headers, body or entire bitstream

Finally on the basic tab, you can specify whether to include the bit-stream data in the payload of the event message. If you select this, you can specify the precise content and format of this data. You can select the whole bit-stream, or just the body, or just the headers.

Customizing an event - Correlators



Add entry

Basic | Correlation | Transaction

Event Correlation

A monitoring application uses event correlators to match events emitted by the same, or related, business transactions. A local transaction correlator links the events emitted by a single invocation of a message flow. A parent transaction correlator links the events from a message flow to a parent message flow or an external application. A global transaction correlator links events from a message flow to one or more related message flows or external applications. An event must contain a local transaction correlator, but need not contain a parent transaction correlator or global transaction correlator.

Local transaction correlator:

● Automatic ○ Specify location of correlator

Description

The local correlator used by the most recent event for this message flow invocation will be used. If no local correlator exists yet, a new unique value will be generated.

Parent transaction correlator:

● Automatic ○ Specify location of correlator

Description

The parent correlator used by the most recent event for this message flow invocation will be used. If no correlator exists yet, no parent correlator will be used.

Global transaction correlator:

● Automatic ○ Specify location of correlator

Description

The global correlator used by the most recent event for this message flow invocation will be used. If no correlator exists yet, no global correlator will be used.

Use this option when your events must be correlated with events from an external process

19 Business Monitor Part 1 © 2010 IBM Corporation

The next task is to specify the correlators for the published events. Correlators enable you to group together the events generated from different points within a message flow, or from different instances of the message flow. This is done using the correlation tab on the event specification wizard.

Each event contains up to three correlation fields.

First, the “local Transaction ID” correlation. This enables you to group all the events generated by a particular instance of the message flow.

It is automatically populated with a unique identifier which is the same for all events emitted during a single invocation of the message flow.

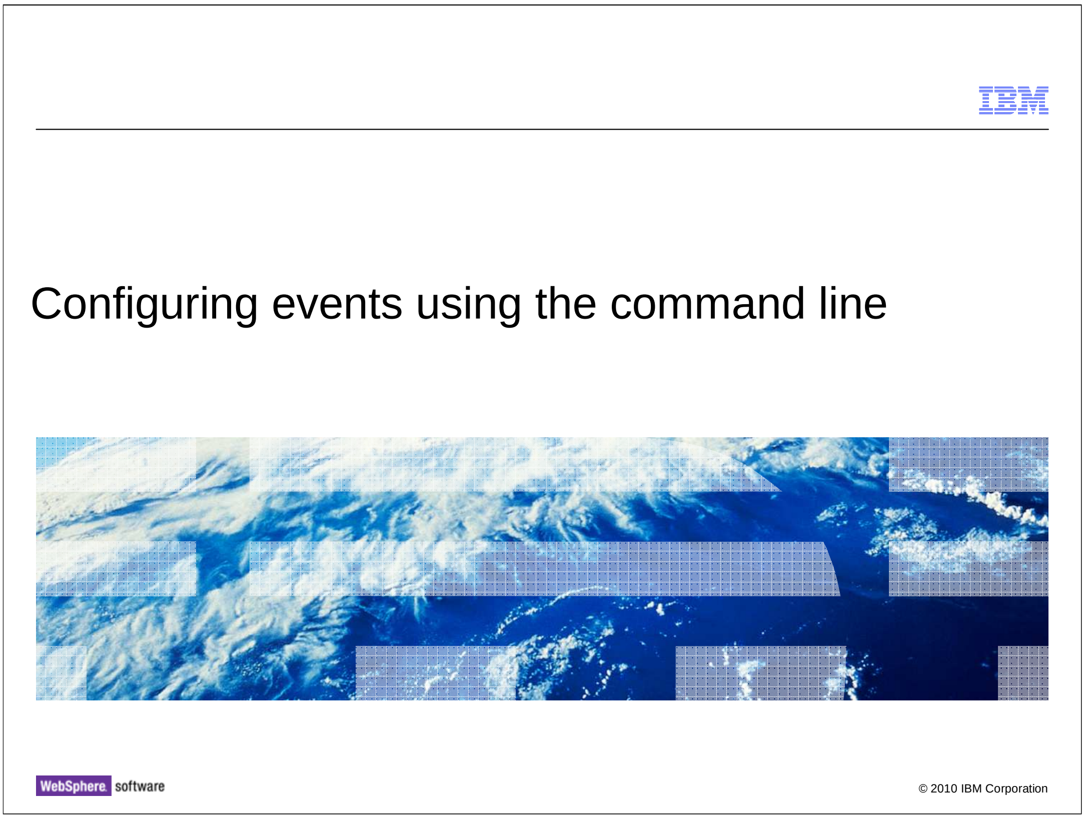
Its value can be set from a field in the message, often from a header. Once set, later events inherit the same value. The value can be specified automatically, or you can specify your own value by clicking the “Specify location” radio button, and using the XPath expression builder as before. If you do not provide your own specification for this, then the value will be populated automatically.

Secondly, the “parent Transaction ID” and “global Transaction ID” can both be used to group events emitted from different instances of the message flow. You should choose one of these values, based on the precise design requirements of your application, and the availability of the information required for this. For example, if a message flow processes multiple order lines as individual messages, then the order number might be used as the correlator, and use the “parent transaction ID” field.

If you do not specify a value for these correlation fields, then the default value is empty.

The value can be set from a field in the message, and this can be in the message header or message data.

If the correlator is derived from a message, then you should only specify it once. This will typically be on the “transaction Start” event. You should not copy the XPath or ESQL expression to other event definitions. This will work, but might incur a performance penalty. The value of the correlator data values are cached in the Environment tree, and later event sources automatically reuse them if they have been set.



Configuring events using the command line

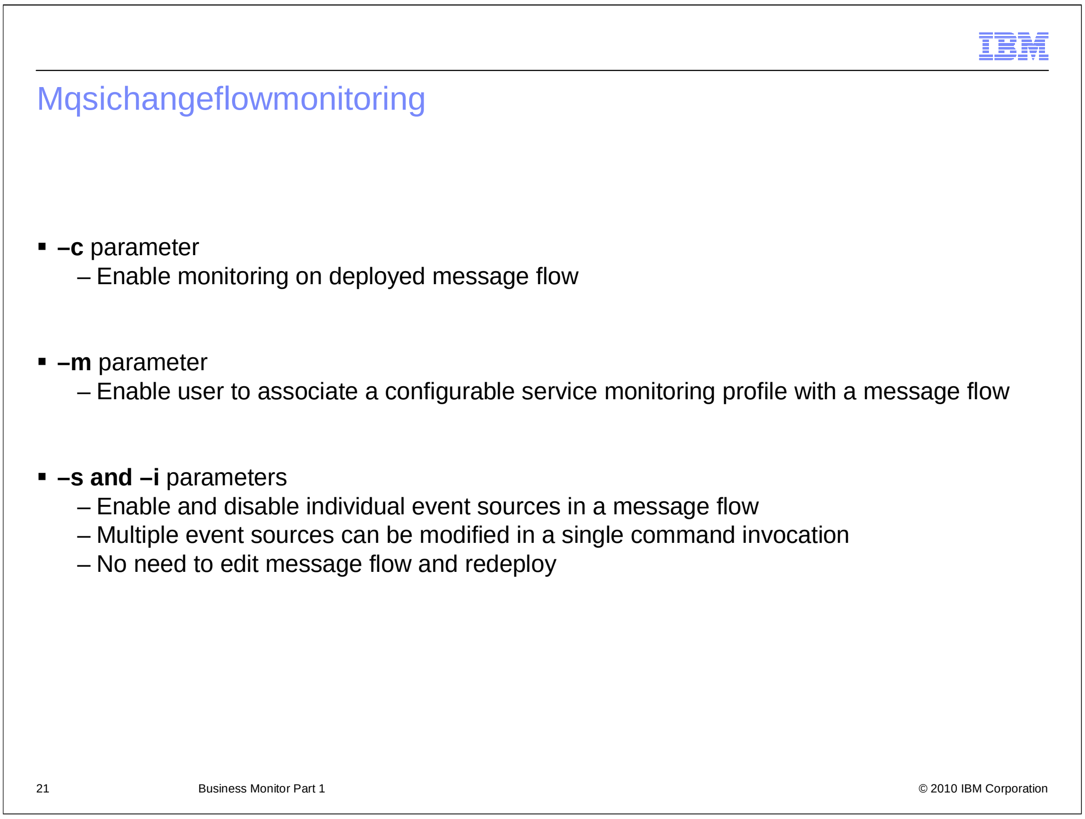
WebSphere software

The “mqsi report flow monitoring” command enables you to report on the status of your message flow monitoring. As with the earlier Change command, you specify the name of the broker, execution group and flow name.

The “-n” parameters reports all of the configured events for the specified message flow. This is equivalent to selecting the message flow editor in the Toolkit, and selecting the Monitoring tab for the entire message flow.

The “-a” parameters reports the available event sources in the specified flow.

The “-x” and “-p” parameters create a monitoring profile, based on the current monitoring properties. This allows you to avoid creating monitoring profile by hand.



Mqsichangeflowmonitoring

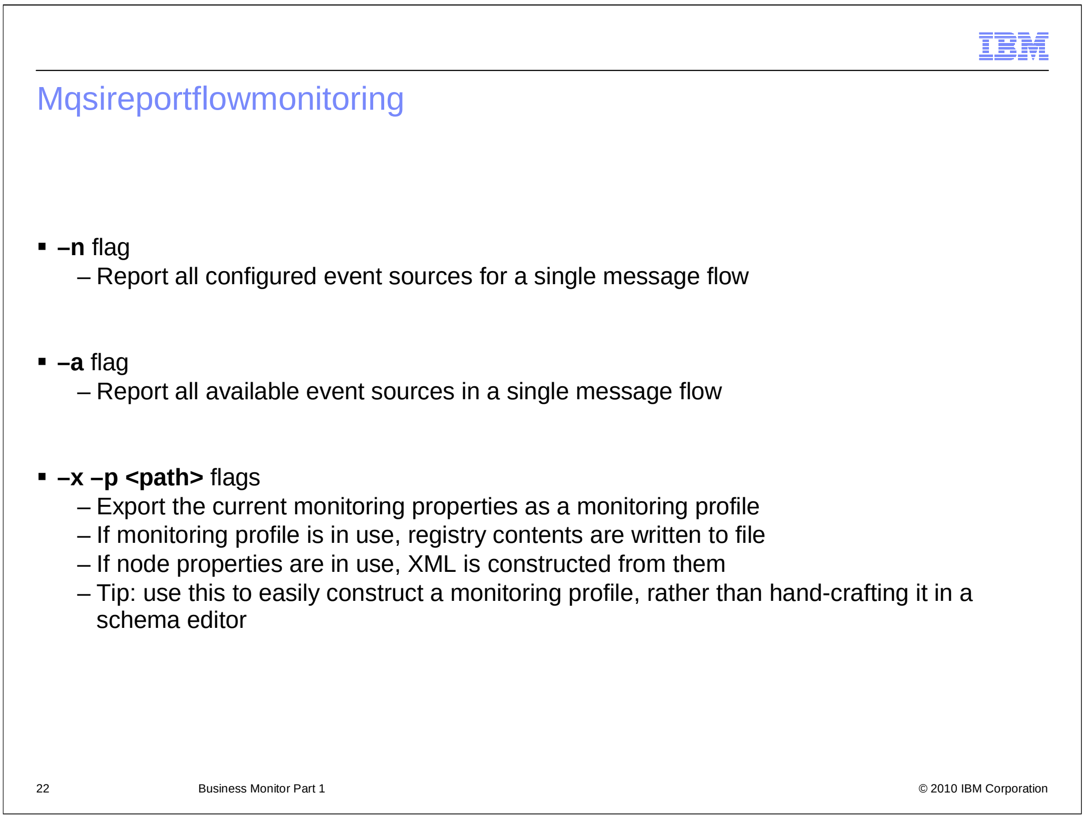
- **–c** parameter
  - Enable monitoring on deployed message flow
- **–m** parameter
  - Enable user to associate a configurable service monitoring profile with a message flow
- **–s and –i** parameters
  - Enable and disable individual event sources in a message flow
  - Multiple event sources can be modified in a single command invocation
  - No need to edit message flow and redeploy

The “mqsi report flow monitoring” command enables you to report on the status of your message flow monitoring. As with the earlier Change command, you specify the name of the broker, execution group and flow name.

The “-n” parameters reports all of the configured events for the specified message flow. This is equivalent to selecting the message flow editor in the Toolkit, and selecting the Monitoring tab for the entire message flow.

The “-a” parameters reports the available event sources in the specified flow.

The “-x” and “-p” parameters create a monitoring profile, based on the current monitoring properties. This allows you to avoid creating monitoring profile by hand.



Mqsireportflowmonitoring

- **–n** flag
  - Report all configured event sources for a single message flow
- **–a** flag
  - Report all available event sources in a single message flow
- **–x –p <path>** flags
  - Export the current monitoring properties as a monitoring profile
  - If monitoring profile is in use, registry contents are written to file
  - If node properties are in use, XML is constructed from them
  - Tip: use this to easily construct a monitoring profile, rather than hand-crafting it in a schema editor

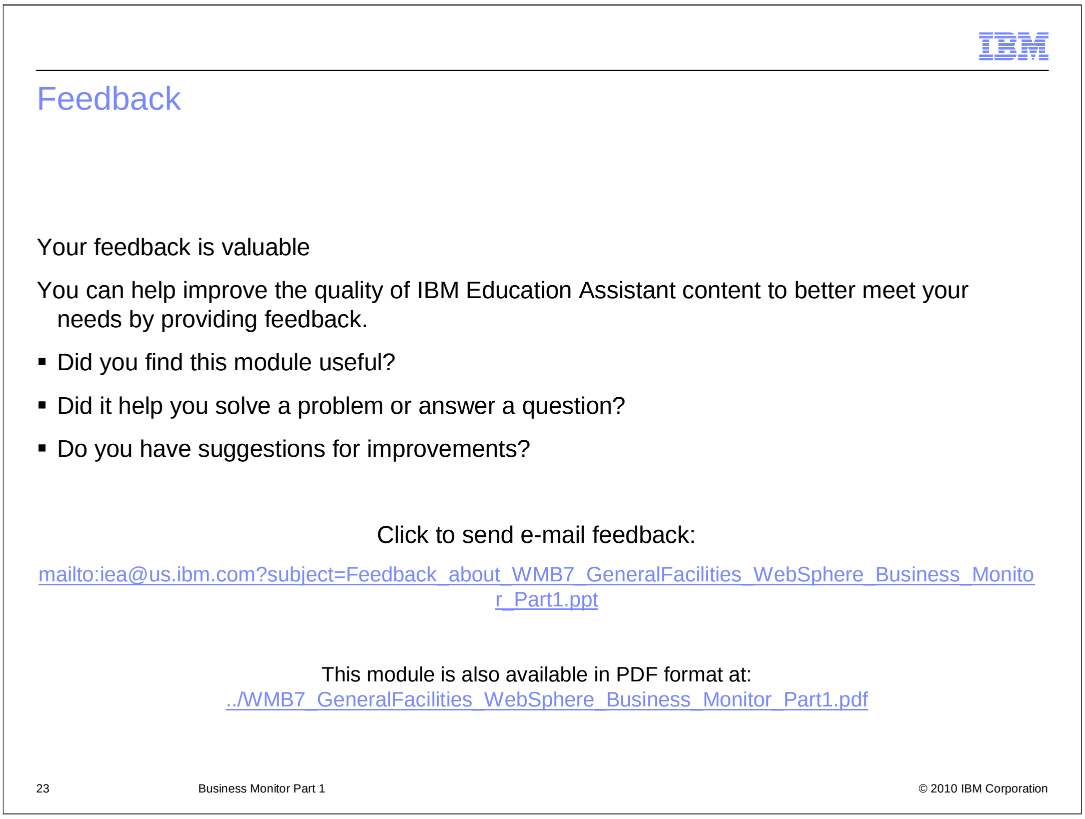
22 Business Monitor Part 1 © 2010 IBM Corporation

The “mqsi report flow monitoring” command enables you to report on the status of your message flow monitoring. As with the earlier Change command, you specify the name of the broker, execution group and flow name.

The “-n” parameters reports all of the configured events for the specified message flow. This is equivalent to selecting the message flow editor in the Toolkit, and selecting the Monitoring tab for the entire message flow.

The “-a” parameters reports the available event sources in the specified flow.

The “-x” and “-p” parameters create a monitoring profile, based on the current monitoring properties. This allows you to avoid creating monitoring profile by hand.



Feedback

Your feedback is valuable

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

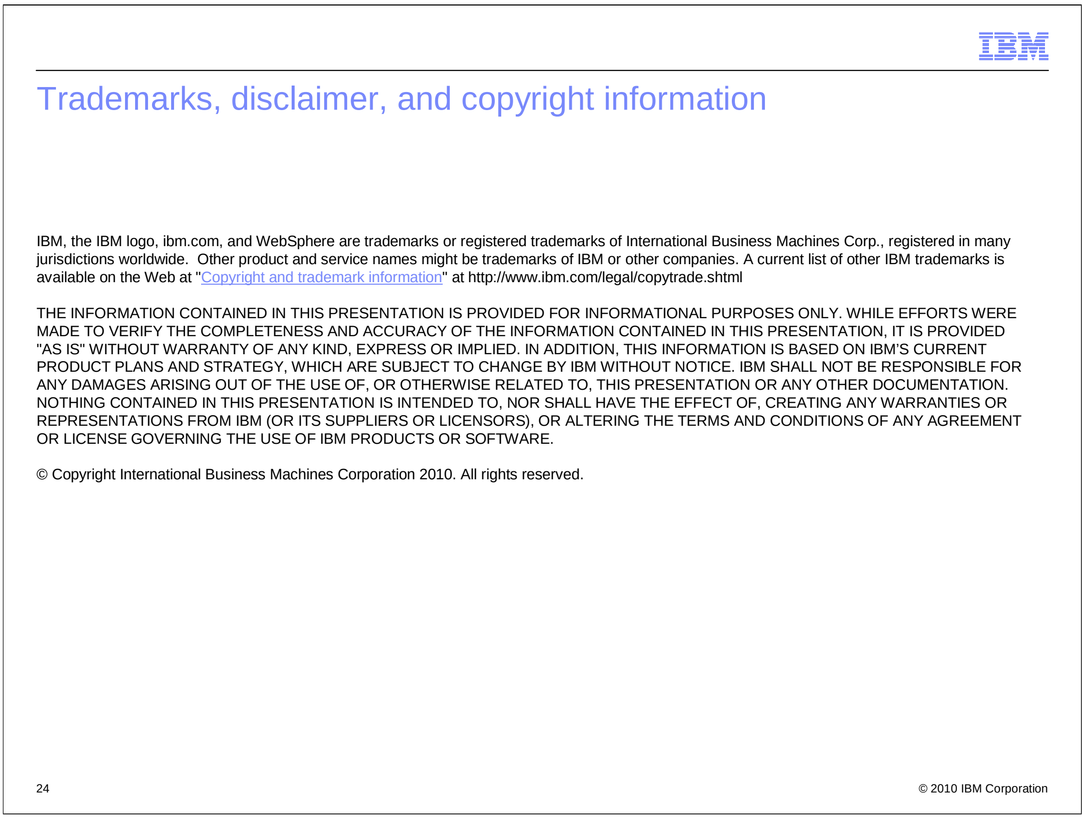
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WMB7_GeneralFacilities_WebSphere_Business_Monitor_Part1.ppt

This module is also available in PDF format at:
../WMB7_GeneralFacilities_WebSphere_Business_Monitor_Part1.pdf

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.