IBM Integration Bus

Trades Application
Viewing data with Record and Replay

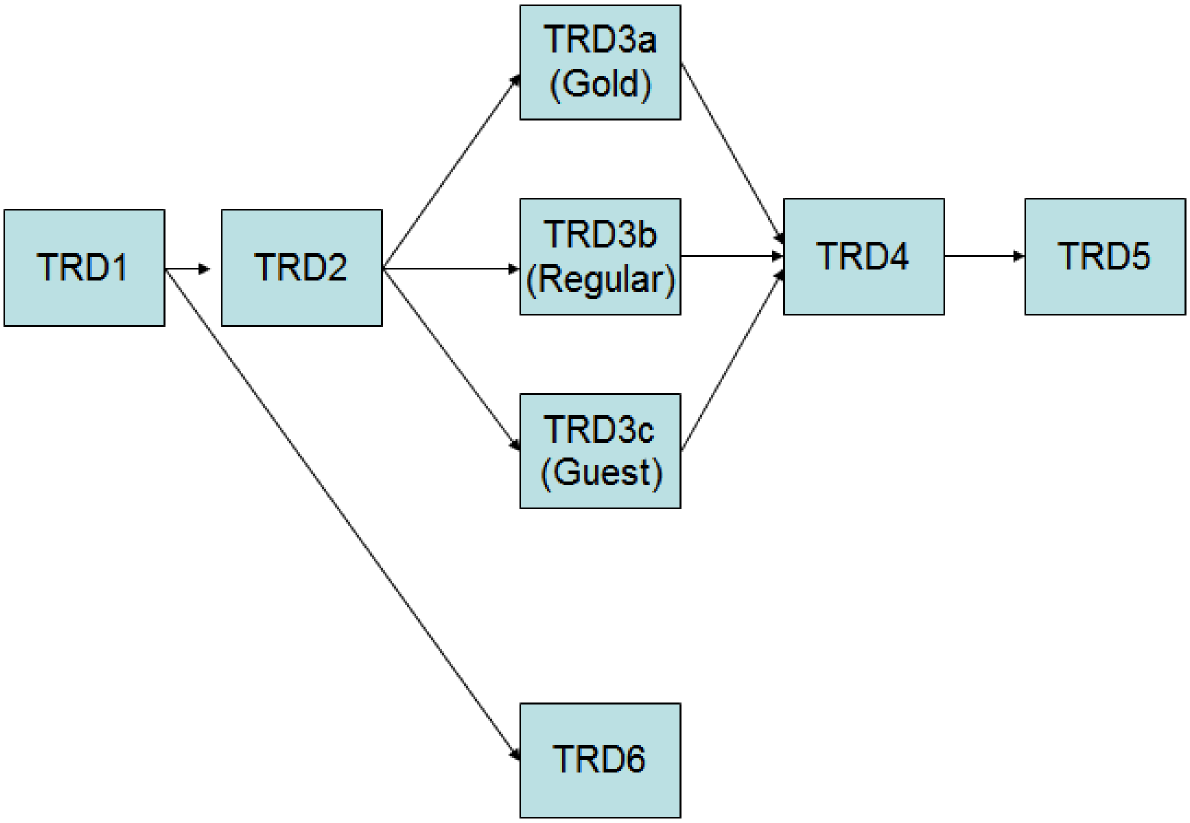
June, 2013

Hands-on lab built at product
code level Version 9.0.0.0

1. Introduction to Record and Replay

This lab will use an application that comprises several message flows. The application process stock trade requests, and each trade executes five message flows, TRD1, TRD2, TRD3*, TRD4 and TRD5. The TRD3 message flow is executed on the basis of customer type (Gold, Regular or Guest). Several of the nodes of the message flows have monitoring points defined on them, using the Monitoring tab on the node properties. These monitoring points publish certain items of the message payload data, and this data is used by the Record and Replay web browser to view messages that have been processed, and to re-submit (replay) the message for re-processing.

The application contains the following message flow structure. The TRD3 flows are selected on the basis of the customer type. The TRD6 flow is executed if a validation failure occurs in TRD1.



2. Setting up the Application for Record and Replay

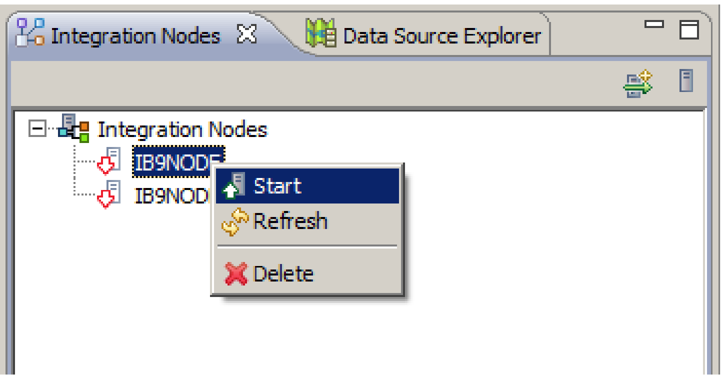
This lab assumes that the default broker, IB9NODE, is available. You may use other brokers, but you will need to adjust the supplied command files accordingly.

2.1 Import the Application

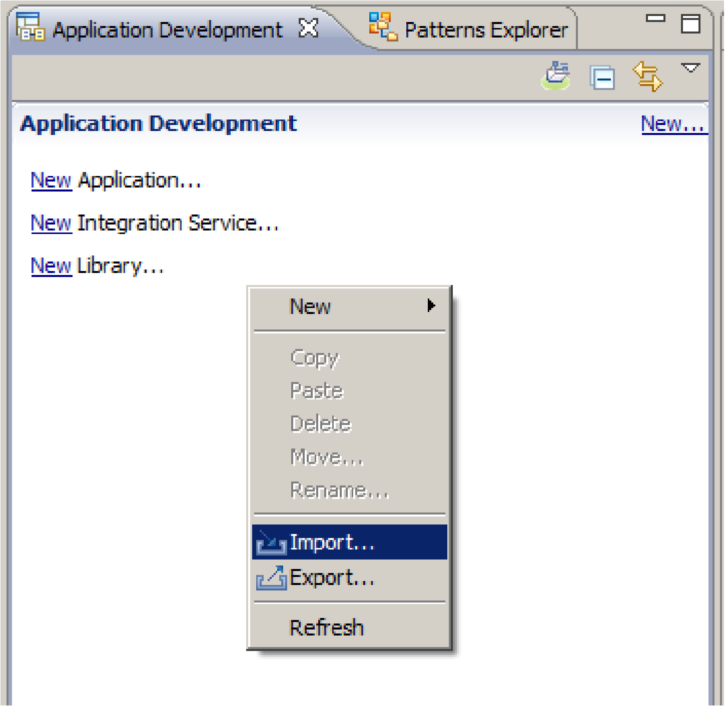
1. If not open, start the Integration Bus Toolkit and the IB Explorer.

   For the Toolkit, use the workspace c:\workspaces\IBWorkshop.

2. In the toolkit's lower left, in the Broker's view, ensure that IB9NODE is started. If it is stopped, start it by selecting IB9NODE, right-clicking, then selecting Start.

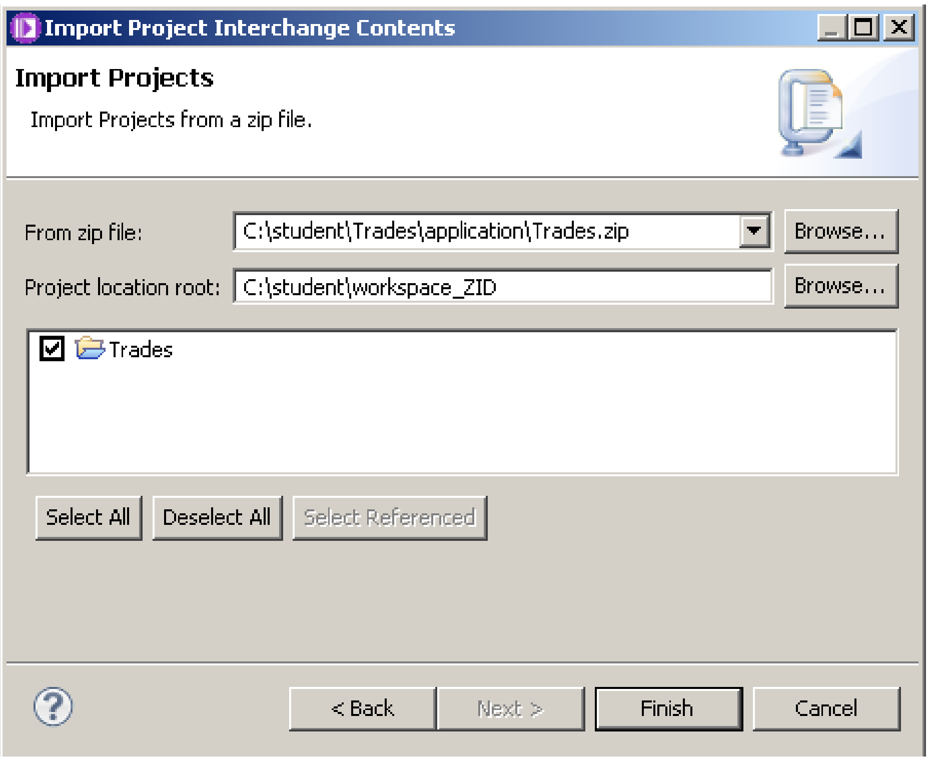


3. Right-click in the white space of the Application Development navigator, and select Import.



4. Select a Project Interchange file (Other, Project Interchange), and select the file c:\student\Trades\application\Trades.zip.

   Ensure the Trades project is checked, and click Finish.



Import Project Interchange Contents

Import Projects

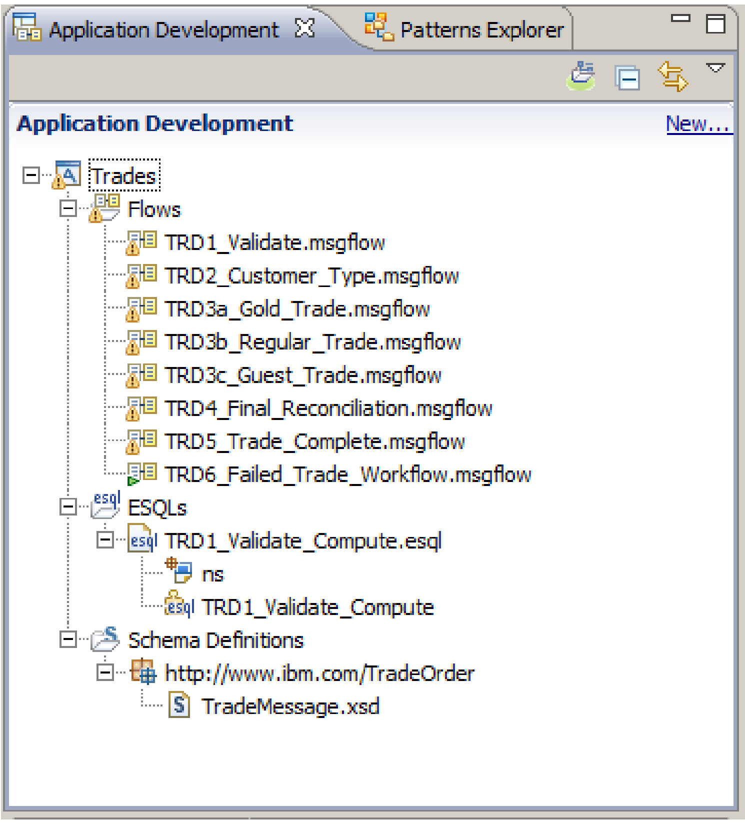
Import Projects from a zip file.

From zip file: C:\student\Trades\application\Trades.zip

Project location root: C:\student\workspace_ZID

☑ Trades

5. The Trades Application will be imported. Expand the application; the application consists of eight message flows. The message flows are executed in sequence, with just one of the TRD3* messages flows being used, depending on the type of customer.



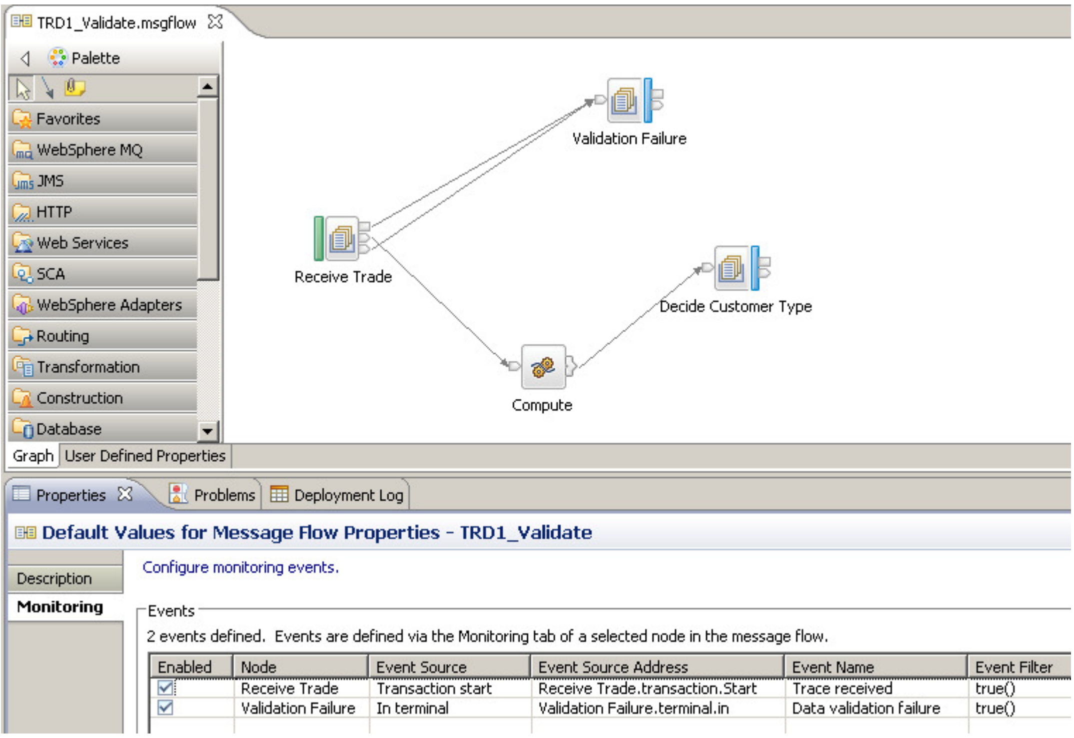
- Trades
  - Flows
    - TRD1_Validate.msgflow
    - TRD2_Customer_Type.msgflow
    - TRD3a_Gold_Trade.msgflow
    - TRD3b_Regular_Trade.msgflow
    - TRD3c_Guest_Trade.msgflow
    - TRD4_Final_Reconciliation.msgflow
    - TRD5_Trade_Complete.msgflow
    - TRD6_Failed_Trade_Workflow.msgflow
  - ESQLs
    - TRD1_Validate_Compute.esql
      - ns
      - TRD1_Validate_Compute
  - Schema Definitions
    - http://www.ibm.com/TradeOrder
      - TradeMessage.xsd

2.2 Review the Flow Monitoring Event Points

The imported message flows contained in the Trades application already has several monitoring event points defined.

1. Open the TRD1_validate message flow, and click on the flow editor canvas (not on a node).

   Clicking the Monitoring tab in the Properties pane will shows all the flow monitoring points in the flow, and their initial status.



Default Values for Message Flow Properties - TRD1_Validate

Configure monitoring events.

Events

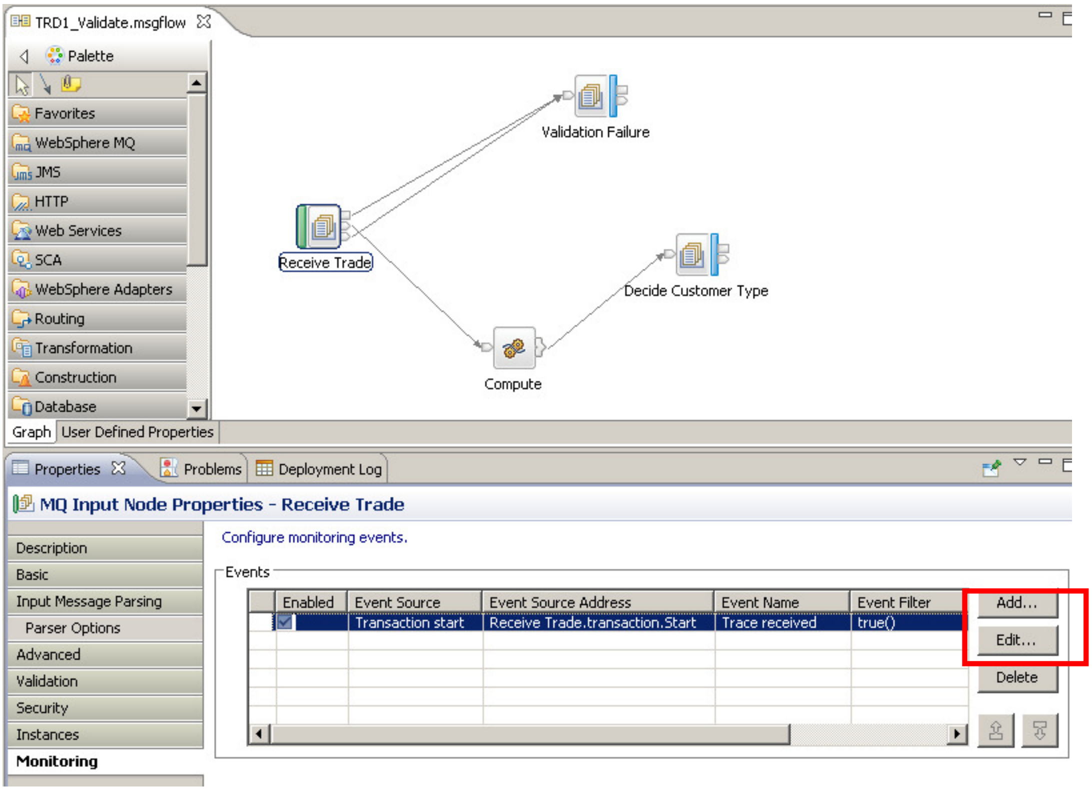
2 events defined. Events are defined via the Monitoring tab of a selected node in the message flow.

Enabled	Node	Event Source	Event Source Address	Event Name	Event Filter
☑	Receive Trade	Transaction start	Receive Trade.transaction.Start	Trace received	true()
☑	Validation Failure	In terminal	Validation Failure.terminal.in	Data validation failure	true()

2. Click on the Input node.

   The Monitoring properties now show just the events that have been defined for the Receive Trade input node. Just a single event is defined, for the Transaction Start monitoring point.

   Highlight the Transaction Start event, and then click the Edit Button.



MQ Input Node Properties - Receive Trade

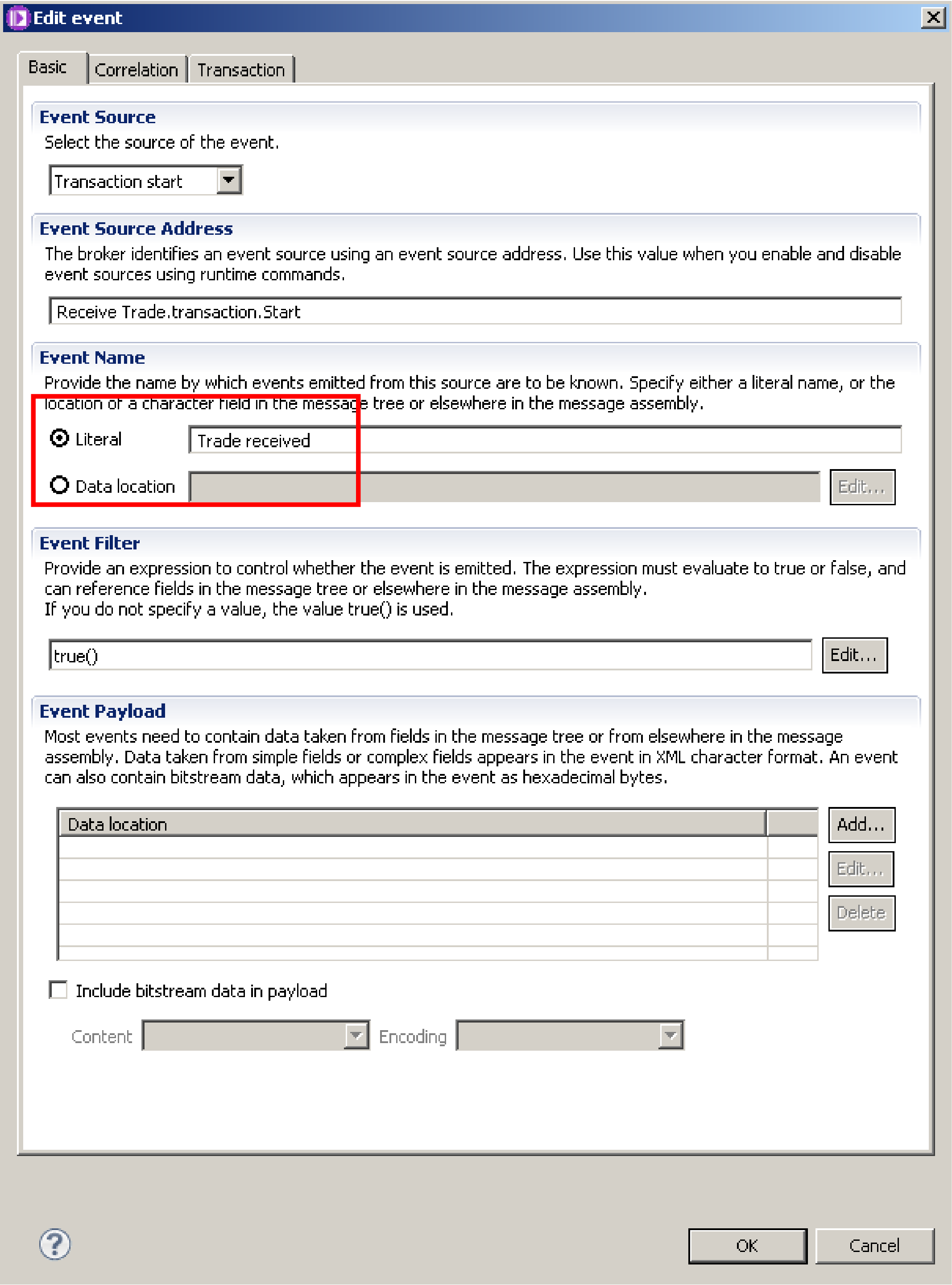
Configure monitoring events.

Events

Enabled	Event Source	Event Source Address	Event Name	Event Filter
☑	Transaction start	Receive Trade.transaction.Start	Trace received	true()

3. On the Basic tab, for the Event Name, the Literal option has been selected, and the Event Name to “Trade instruction received”.

   Note that the Event Filter can be used to dynamically determine whether to emit a monitoring event, based on the value of a message element. The edit button can be used to generate the required XPath syntax for this. This example will emit events for all messages that it processes.



Edit event

Basic | Correlation | Transaction

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.

Receive Trade.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: Trade received

○ 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.

true()

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	

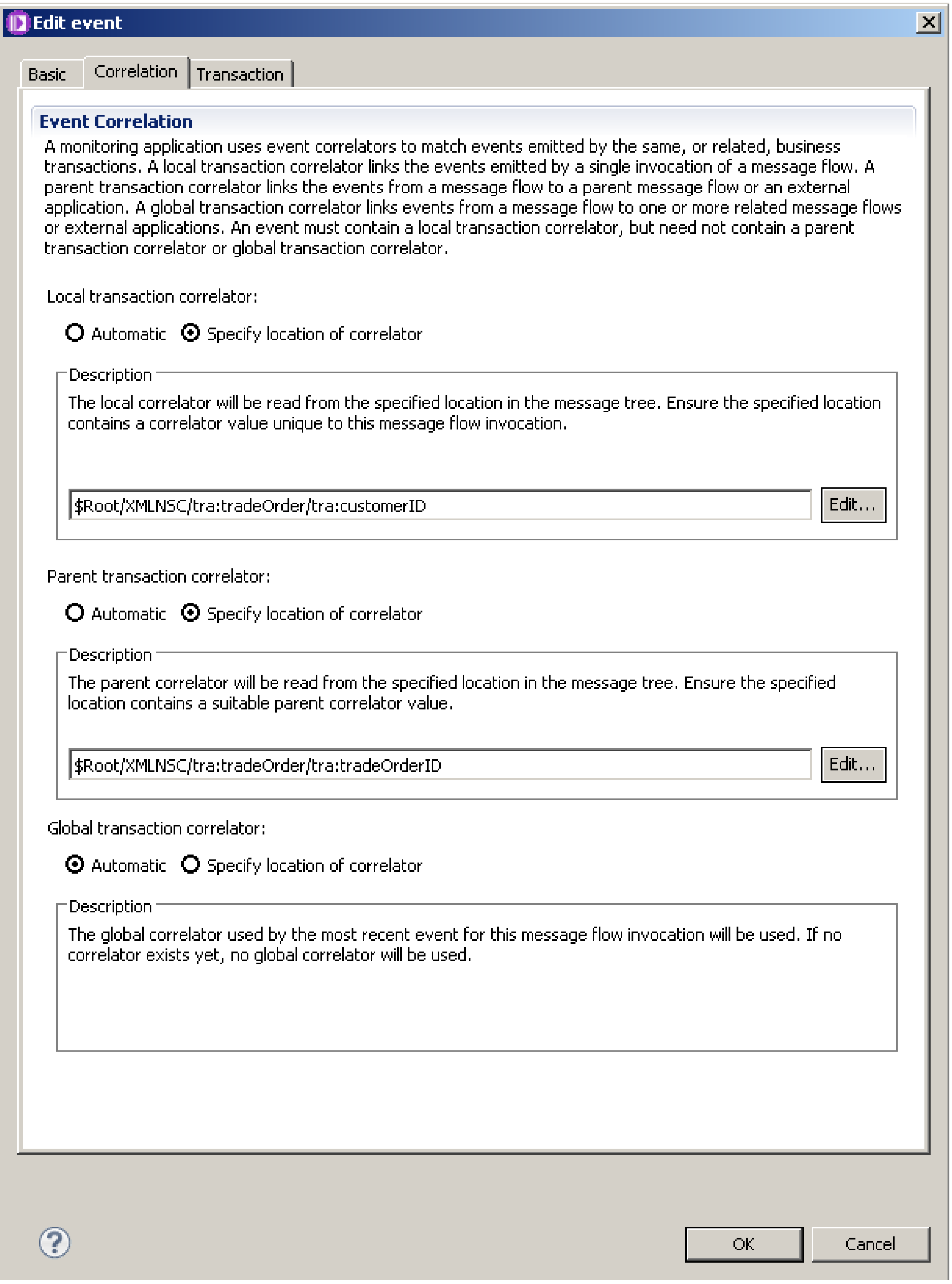
□ Include bitstream data in payload

Content | Encoding

4. On the Correlation tab, the Local transaction correlator has been set to “customerID”, and the Parent transaction correlator has been set to “tradeOrderID”.

   The Global transaction correlator has been left for values in later message flows.

   If you want to change these values, you can use the Edit button to start the XPath editor. Or, you can simply edit the correlation fields directly.



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 will be read from the specified location in the message tree. Ensure the specified location contains a correlator value unique to this message flow invocation.

$Root/XMLNSC/tra:tradeOrder/tra:customerID

Parent transaction correlator:

○ Automatic ● Specify location of correlator

Description

The parent correlator will be read from the specified location in the message tree. Ensure the specified location contains a suitable parent correlator value.

$Root/XMLNSC/tra:tradeOrder/tra:tradeOrderID

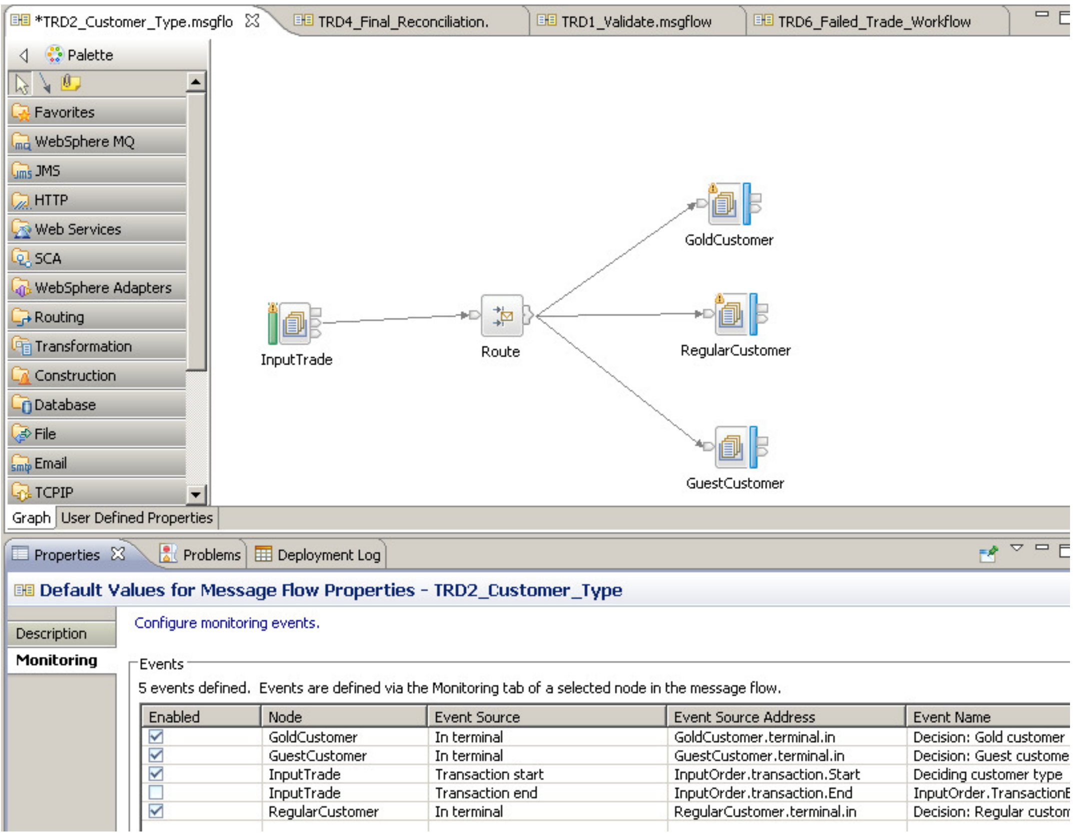
Global transaction correlator:

● 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.

5. Cancel the monitor dialogue windows for TRD1, and then open the TRD2_Customer_Type message flow, and review the monitoring configuration:



Default Values for Message Flow Properties - TRD2_Customer_Type

Configure monitoring events.

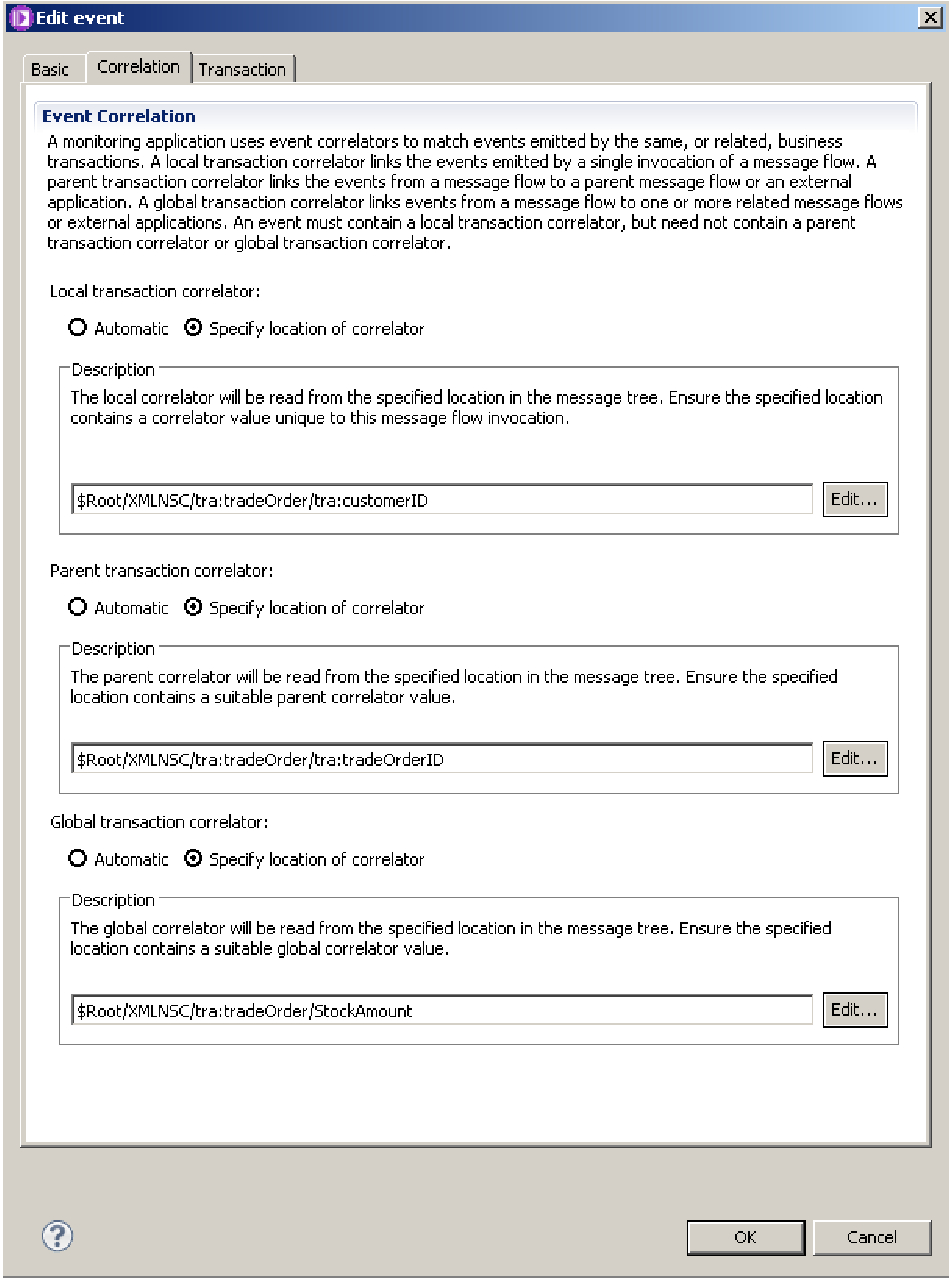
Events

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
☑	GoldCustomer	In terminal	GoldCustomer.terminal.in	Decision: Gold customer
☑	GuestCustomer	In terminal	GuestCustomer.terminal.in	Decision: Guest custome
☑	InputTrade	Transaction start	InputOrder.transaction.Start	Deciding customer type
☐	InputTrade	Transaction end	InputOrder.transaction.End	InputOrder.TransactionE
☑	RegularCustomer	In terminal	RegularCustomer.terminal.in	Decision: Regular custon

6. Open the monitoring event for the InputTrade node, and select the Correlation tab.

   For the Global Transaction Correlator, you will see that the value has been set to $Root/XMLNSC/tra.tradeOrder/StockAmount. StockAmount is a new message tree element that has been created by the TRD1_Validate message flow, and is a combination of the trade and amount elements.



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 will be read from the specified location in the message tree. Ensure the specified location contains a correlator value unique to this message flow invocation.

$Root/XMLNSC/tra:tradeOrder/tra:customerID Edit...

Parent transaction correlator:

○ Automatic ● Specify location of correlator

Description

The parent correlator will be read from the specified location in the message tree. Ensure the specified location contains a suitable parent correlator value.

$Root/XMLNSC/tra:tradeOrder/tra:tradeOrderID Edit...

Global transaction correlator:

○ Automatic ● Specify location of correlator

Description

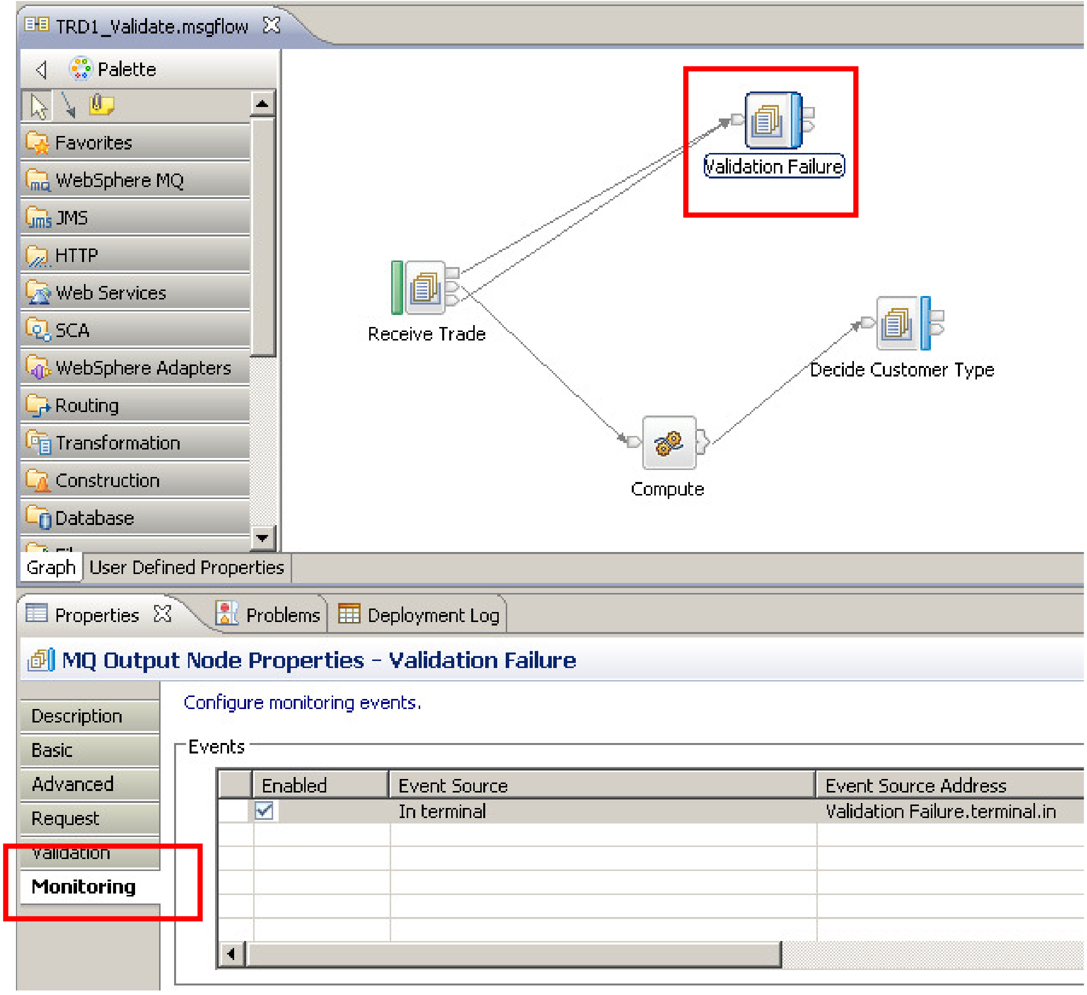
The global correlator will be read from the specified location in the message tree. Ensure the specified location contains a suitable global correlator value.

$Root/XMLNSC/tra:tradeOrder/StockAmount Edit...

2.3 Configure the monitoring event to emit the message payload

The Replay function uses the raw message payload (bitstream) that can be emitted by a event monitoring point. If you do not plan to use the Replay function, do not configure the monitoring event to emit the bitstream. In this example, we will emit the bitstream on the node that processes the Validation failures.

1. In the TRD1_Validate message flow, click the Validation Failure node.
   Select the Monitoring property.



MQ Output Node Properties - Validation Failure

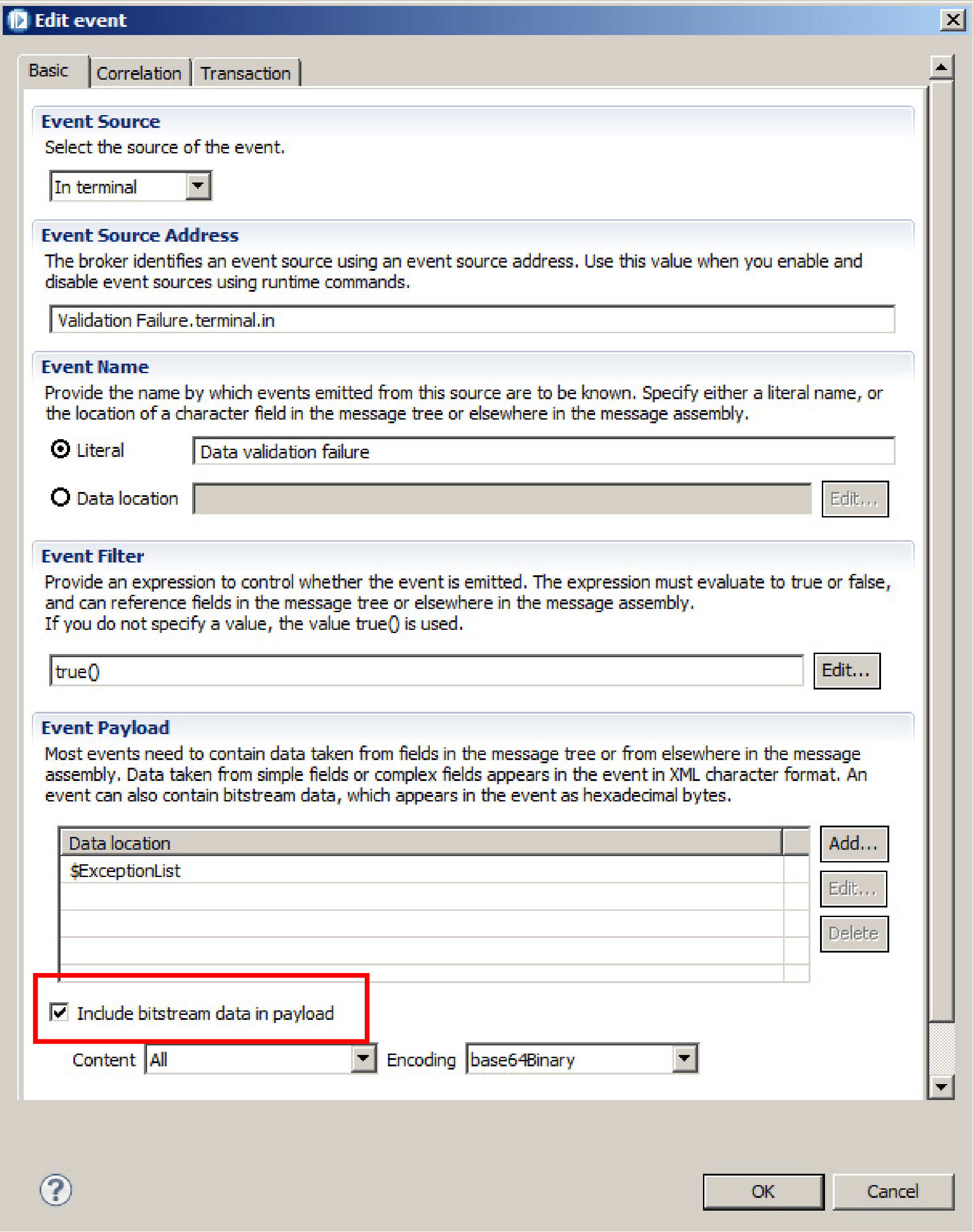
Configure monitoring events.

Events

Enabled	Event Source	Event Source Address
☑	In terminal	Validation Failure.terminal.in

2. Click Edit to open the Monitoring Event editor for this node.

   On the Basic tab, in the Event Payload section, note that “Include bitstream data in payload” has been selected.
   Click OK, and then close the message flow.



Edit event

Basic | Correlation | Transaction

Event Source

Select the source of the event.

In terminal

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.

Validation Failure.terminal.in

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: Data validation failure

○ 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.

true()

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.

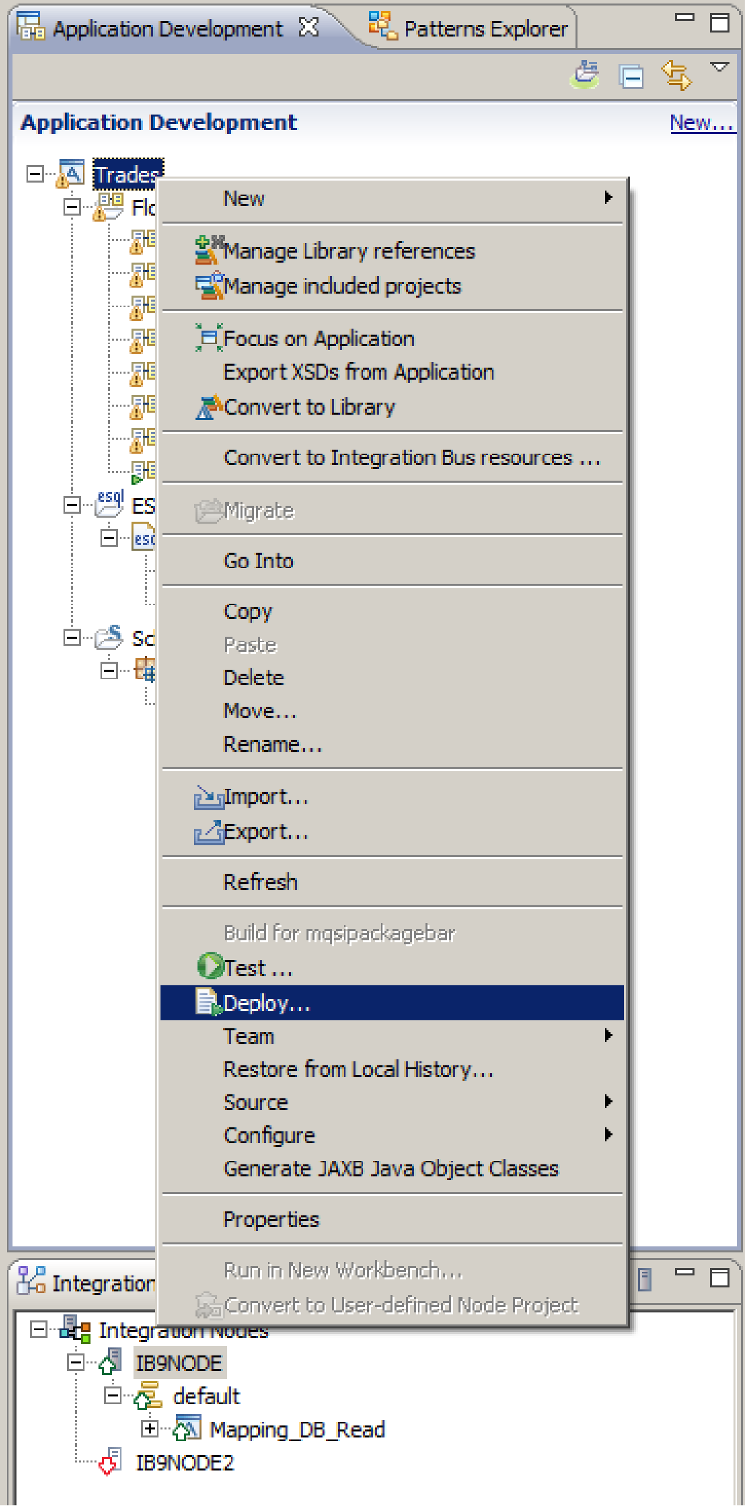
Data location
$ExceptionList

☑ Include bitstream data in payload

Content: All Encoding: base64Binary

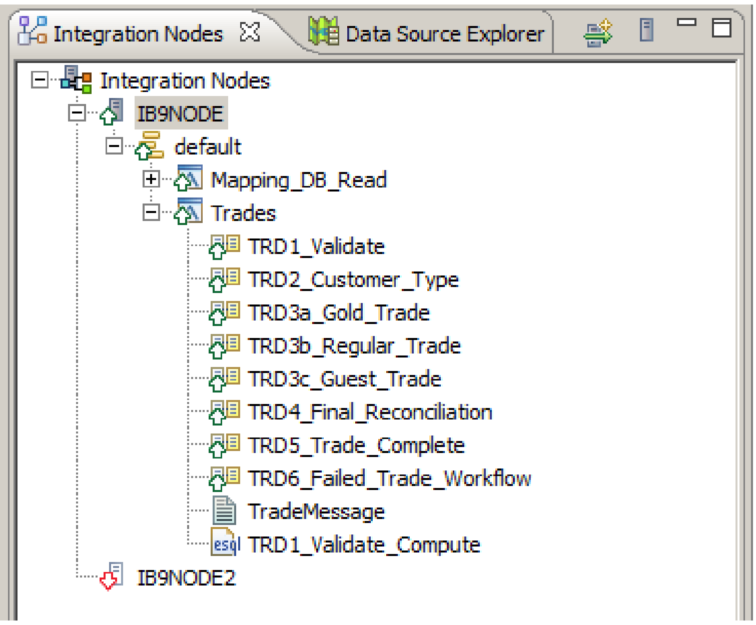
2.4 Deploy the Application

1. Deploy the Trades Application, by right-clicking on the Application (not the message flows), and selecting Deploy. Deploy it to the default Integration Server.



2. Select the default Integration Server.

3. Validate that the Trades application has been deployed.



2.5 Recreate the monitoring tables and enable flow monitoring

1. In an Integration Bus Command Console, change directory to c:\student\Trades\install\DBSetup.

   Run the command : **`CreateTRADES_Tables.bat`**

   This will open a new DOS window and recreate the Record/Replay tables in the TRADES database. This is required so that your scenario starts with a clean display of monitor events, and so that new events are easily viewable on the web browser.

2. Change directory to c:\student\Trades\monitoring.

   Note. In this example, we have configured monitoring points on the message flow, using the Monitoring function on the flow and node properties. Flow monitoring can also be achieved non-invasively by using monitoring templates, not shown in this example.

   Run the command: **`enableMonitoringTrades`**

   This will invoke the Integration Node command:

```
mqsichangeflowmonitoring IB9NODE
      -e default
      -k Trades
      -f TRD1_Validate
      -c active
```

   with equivalent commands for the other message flows in the Trades application.

   NB: If you redeploy the Application, this will reset the flow monitoring status, and you must reissue the command above to reactivate flow monitoring.

2.6 Activate Security for the IB9NODE

1. In an Integration Bus Command Console, issue the commands:

```
mqsistop IB9NODE
```

```
mqsichangebroker IB9NODE -s active
```

   The Integration node must now be restarted:

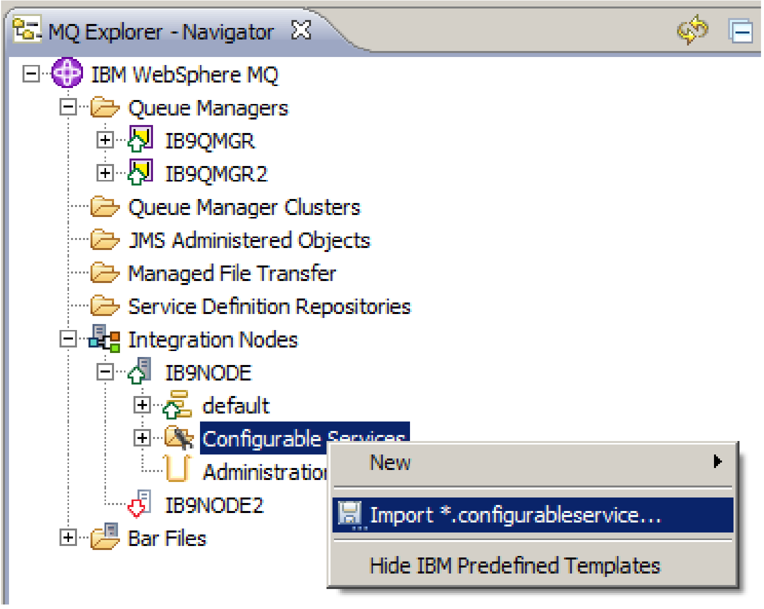
```
mqsistart IB9NODE
```

   This will require any users of the Web Admin browser to login, to see the required items.

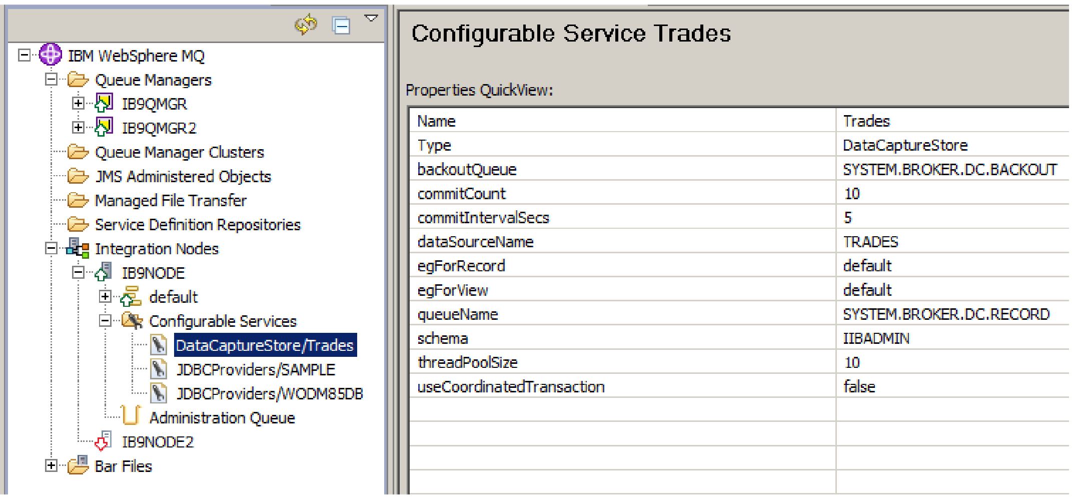
2.7 Import the Data Store configurable services

The record/replay function uses a Data Store, which represents the database which holds the captured monitoring events. The data Store is defined to the Integration bus by using configurable services.
You can define these manually, but for this lab we have already created these. These services need to be imported into IB9NODE.

1. In an Integration Bus Explorer, expand IB9NODE, right-click the configurable services and select Import.



2. Navigate to c:\student\trades\configurable_services, and import Trades_data_capture_store.configurableservice.



Configurable Service Trades

Properties QuickView:

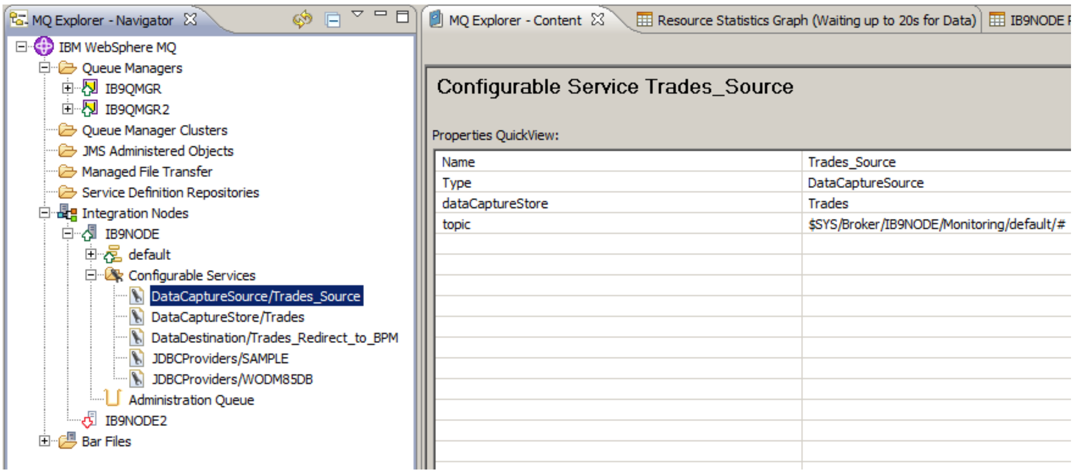
Name	Trades
Type	DataCaptureStore
backoutQueue	SYSTEM.BROKER.DC.BACKOUT
commitCount	10
commitIntervalSecs	5
dataSourceName	TRADES
egForRecord	default
egForView	default
queueName	SYSTEM.BROKER.DC.RECORD
schema	IIBADMIN
threadPoolSize	10
useCoordinatedTransaction	false

3. Perform the same import for:
Trades_source.configurableservice

Trades_BPM_Data_Destination.configurableservice

When complete, you will have added three new configurable services, as shown.

Note that the Trades_source service subscribes to the topic $SYS/Broker/IB9NODE/Monitoring/default/#. This means that this data source will collect all monitoring events that are generated by applications in the default server (execution group) of the IB9NODE integration node. Events emitted in other nodes or servers will not be collected by this data source.



Configurable Service Trades_Source

Properties QuickView:

Name	Trades_Source
Type	DataCaptureSource
dataCaptureStore	Trades
topic	$SYS/Broker/IB9NODE/Monitoring/default/#

2.8 Define a web user for Record/Replay

4. In an Integration Bus Command Console, change directory to c:\student\Trades\webadmin

Run the command : `set_record_replay_users_MQauth`

This will create the required MQ authorities for the profile ESBProfile3

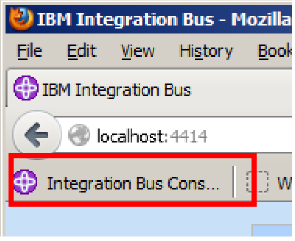
Run the command : `define_record_replay_users`

This will create new web users for the Integration Bus. The “record3” user will be defined with the ESBProfile3, which provides full access to the record/replay function.

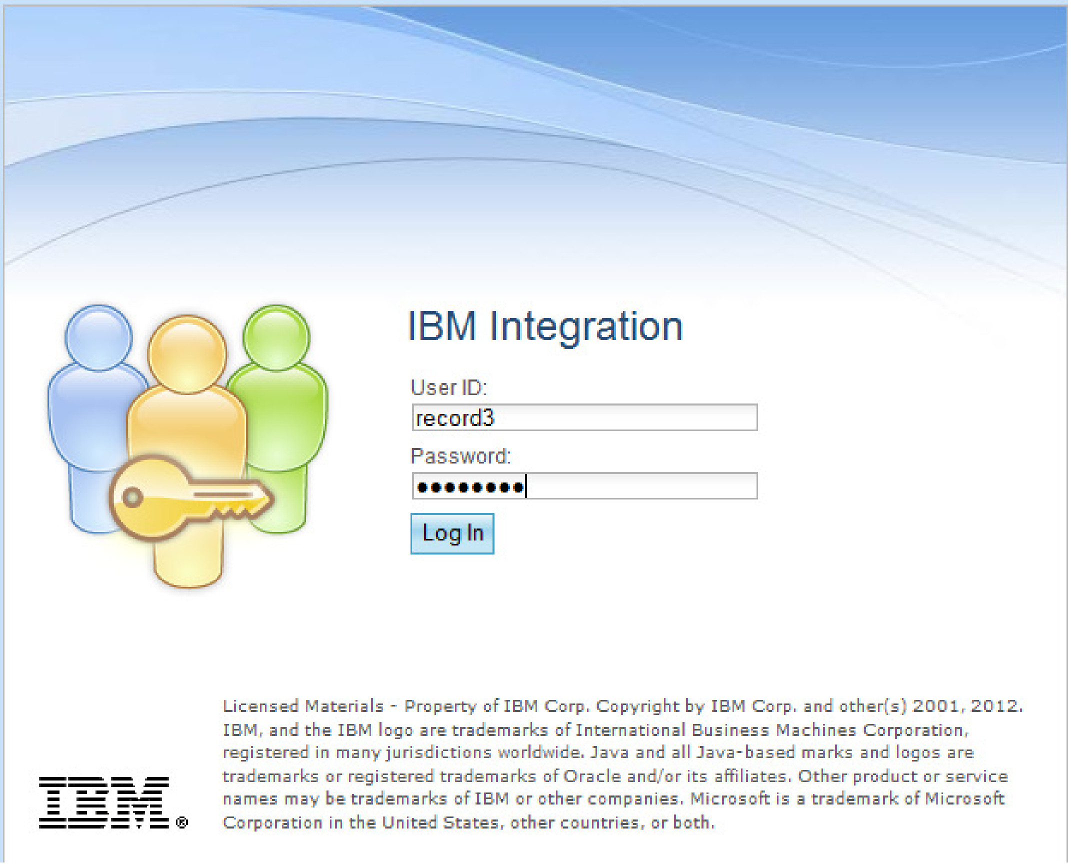
3. View messages in the Web Browser

1. Open a web browser; Firefox is installed on the prebuilt system.

   Use the URL: http://localhost:4414 (a configured link, "`Integration Bus Console`", is provided on the prebuilt vmware image.



2. The following login screen will appear. Log in to the web interface using the `record3` user. The password is "`passw0rd`" (using a zero character).



User ID: record3

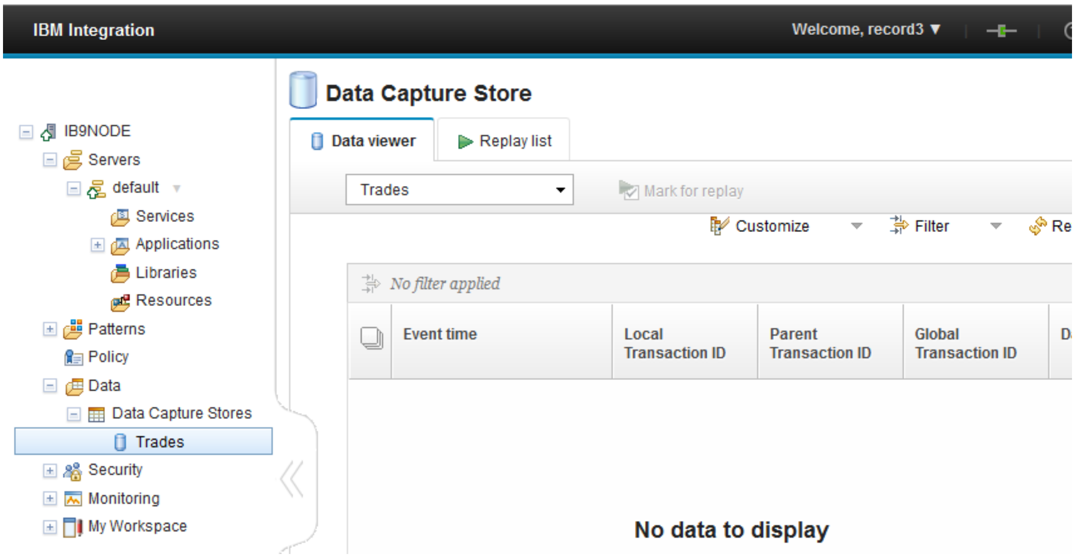
Password: ••••••••

Licensed Materials - Property of IBM Corp. Copyright by IBM Corp. and other(s) 2001, 2012. IBM, and the IBM logo are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Java and all Java-based marks and logos are trademarks or registered trademarks of Oracle and/or its affiliates. Other product or service names may be trademarks of IBM or other companies. Microsoft is a trademark of Microsoft Corporation in the United States, other countries, or both.

3. The “record3” user has full Integration Bus administrator access (because it uses the ESBProfile3 MQ user), so in the navigator you will see full details of the broker, the deployed applications, and you will be able to perform various operations, such as Start and Stop on these applications.

   Expand the Data item in the navigator, expand Data Capture Stores, and select the Trades data capture store.

   If you have recreated the record/replay tables, you will not see any items in the event list for Trades.



4. Use RFHUtil (on the Start menu if not already open) to send some new events to the Trades application.

   In RFHUtil, open the file c:\student\Trades\data\TradeMessageGold_BNY347290.xml (a gold customer).

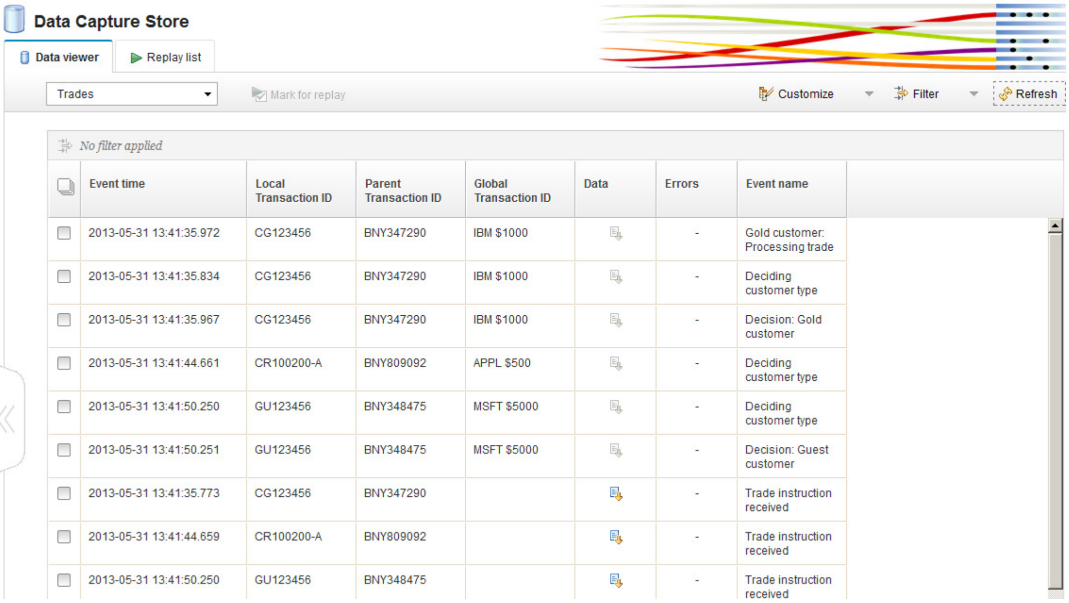
   Send one instance of the data to the queue TRADE.VALIDATE.IN.

   Repeat with the file ….\TradeMessageRegular…(Regular customer).

   Repeat with the file ….. \TradeMessageGuest ….. (Guest customer).

5. On the Data Viewer window, click Refresh.

The default display in the web browser will use the standard column heading names.



□	Event time	Local Transaction ID	Parent Transaction ID	Global Transaction ID	Data	Errors	Event name
□	2013-05-31 13:41:35.972	CG123456	BNY347290	IBM $1000	🗎	-	Gold customer: Processing trade
□	2013-05-31 13:41:35.834	CG123456	BNY347290	IBM $1000	🗎	-	Deciding customer type
□	2013-05-31 13:41:35.967	CG123456	BNY347290	IBM $1000	🗎	-	Decision: Gold customer
□	2013-05-31 13:41:44.661	CR100200-A	BNY809092	APPL $500	🗎	-	Deciding customer type
□	2013-05-31 13:41:50.250	GU123456	BNY348475	MSFT $5000	🗎	-	Deciding customer type
□	2013-05-31 13:41:50.251	GU123456	BNY348475	MSFT $5000	🗎	-	Decision: Guest customer
□	2013-05-31 13:41:35.773	CG123456	BNY347290		🗎	-	Trade instruction received
□	2013-05-31 13:41:44.659	CR100200-A	BNY809092		🗎	-	Trade instruction received
□	2013-05-31 13:41:50.250	GU123456	BNY348475		🗎	-	Trade instruction received

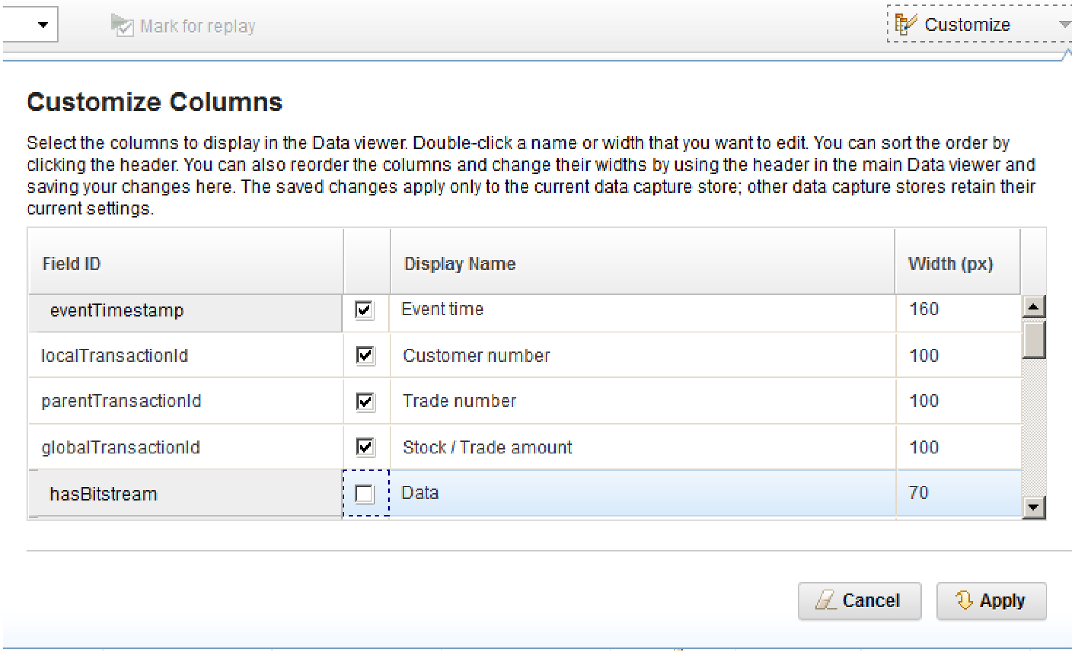
6. Click the Customize button.

   Several facilities are available here:

   1. Change the display name of each column by double-clicking the required display name, and change to the appropriate name. These changes are stored in the Broker Registry, and are retained uniquely for each data capture store. All users who display data from the same data capture store will see the changes made by this user. If you wish to record and view data which has a different meaning, you should record this in a separate data capture store.
   2. Select or de-select any of the recorded fields for display.
   3. Override the width of the displayed column (widths can still be changed using the divider bars).

   Make the following changes:

   localTransactionId field has display name “Customer number”.
   parentTransactionId has display name “Trade number”.
   globalTransactionId has display name “Stock / Trade amount”.
   eventName has display name “Trade processing stage”.
   eventName (Trade Processing stage) has field width 160.
   hasBitstream – deselect.

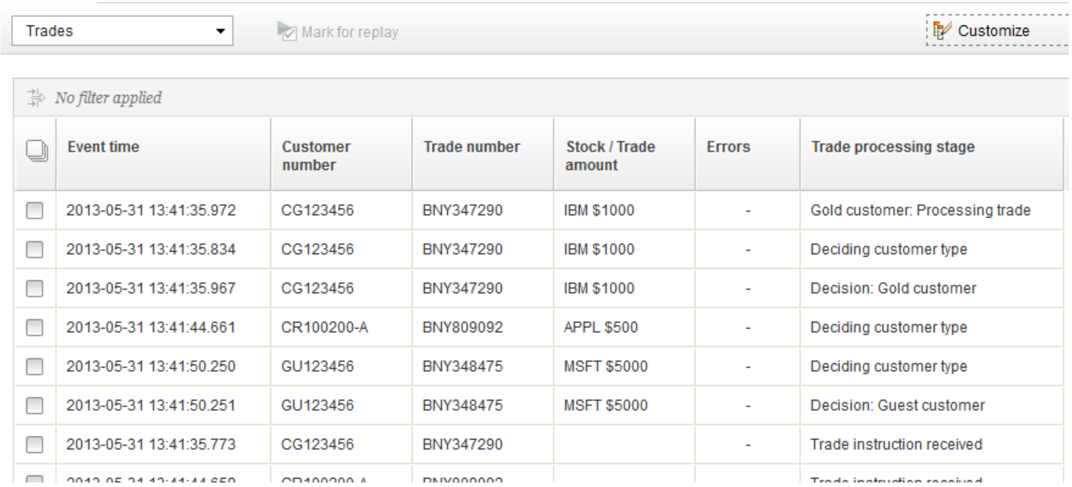


Customize Columns

Select the columns to display in the Data viewer. Double-click a name or width that you want to edit. You can sort the order by clicking the header. You can also reorder the columns and change their widths by using the header in the main Data viewer and saving your changes here. The saved changes apply only to the current data capture store; other data capture stores retain their current settings.

Field ID		Display Name	Width (px)
eventTimestamp	☑	Event time	160
localTransactionId	☑	Customer number	100
parentTransactionId	☑	Trade number	100
globalTransactionId	☑	Stock / Trade amount	100
hasBitstream	□	Data	70

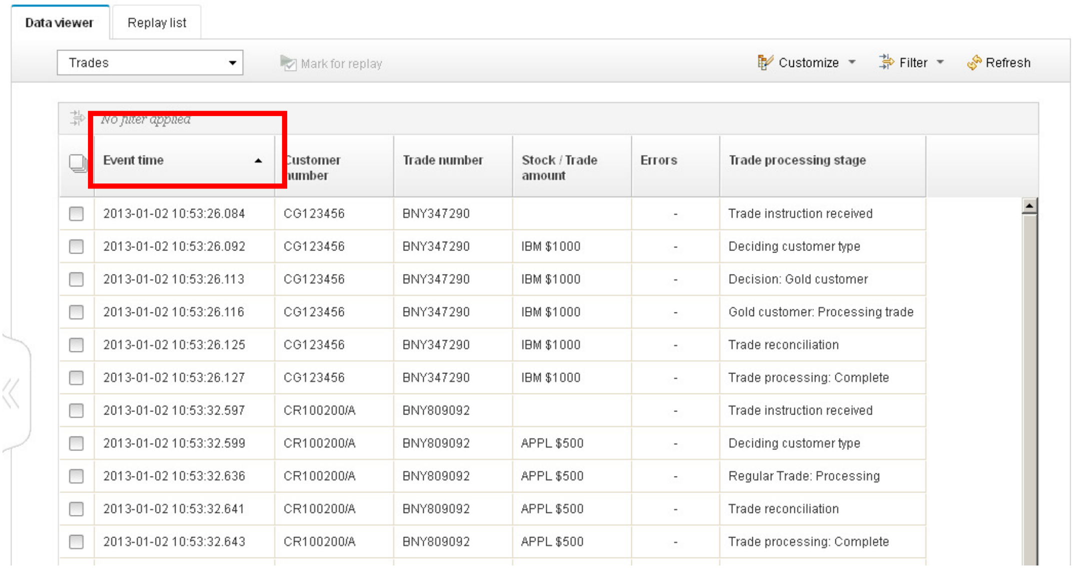
7. Click Apply. The updated display will look something like this:



No filter applied

□	Event time	Customer number	Trade number	Stock / Trade amount	Errors	Trade processing stage
□	2013-05-31 13:41:35.972	CG123456	BNY347290	IBM $1000	-	Gold customer: Processing trade
□	2013-05-31 13:41:35.834	CG123456	BNY347290	IBM $1000	-	Deciding customer type
□	2013-05-31 13:41:35.967	CG123456	BNY347290	IBM $1000	-	Decision: Gold customer
□	2013-05-31 13:41:44.661	CR100200-A	BNY809092	APPL $500	-	Deciding customer type
□	2013-05-31 13:41:50.250	GU123456	BNY348475	MSFT $5000	-	Deciding customer type
□	2013-05-31 13:41:50.251	GU123456	BNY348475	MSFT $5000	-	Decision: Guest customer
□	2013-05-31 13:41:35.773	CG123456	BNY347290		-	Trade instruction received

8. Click on the Event Time column to display the oldest events first.



No filter applied

□	Event time ▲	Customer number	Trade number	Stock / Trade amount	Errors	Trade processing stage
□	2013-01-02 10:53:26.084	CG123456	BNY347290		-	Trade instruction received
□	2013-01-02 10:53:26.092	CG123456	BNY347290	IBM $1000	-	Deciding customer type
□	2013-01-02 10:53:26.113	CG123456	BNY347290	IBM $1000	-	Decision: Gold customer
□	2013-01-02 10:53:26.116	CG123456	BNY347290	IBM $1000	-	Gold customer: Processing trade
□	2013-01-02 10:53:26.125	CG123456	BNY347290	IBM $1000	-	Trade reconciliation
□	2013-01-02 10:53:26.127	CG123456	BNY347290	IBM $1000	-	Trade processing: Complete
□	2013-01-02 10:53:32.597	CR100200/A	BNY809092		-	Trade instruction received
□	2013-01-02 10:53:32.599	CR100200/A	BNY809092	APPL $500	-	Deciding customer type
□	2013-01-02 10:53:32.636	CR100200/A	BNY809092	APPL $500	-	Regular Trade: Processing
□	2013-01-02 10:53:32.641	CR100200/A	BNY809092	APPL $500	-	Trade reconciliation
□	2013-01-02 10:53:32.643	CR100200/A	BNY809092	APPL $500	-	Trade processing: Complete

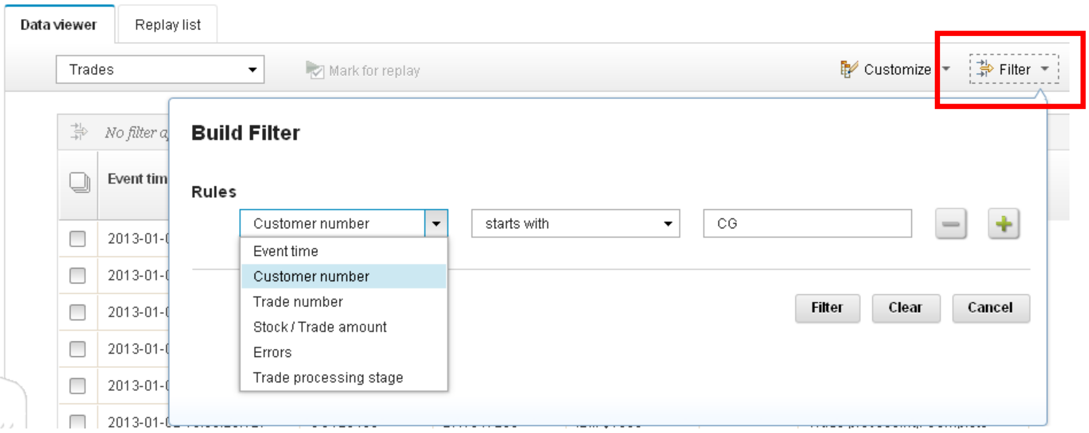
9. Click on any other column to order the events depending on the content of the clicked column (eg. Customer number or Trade number).

10. Filtering.

You can limit the displayed data by using the Filter function. Click Filter, and then specify some filter criteria.

In this example, select “Customer number” starts with “CG”.

Click Filter to activate the defined filters.

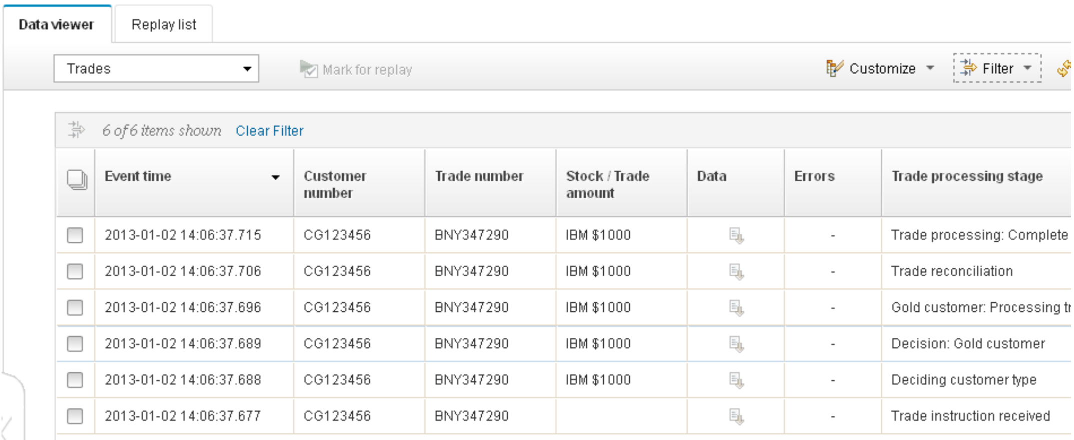


Build Filter

Rules

Customer number | starts with | CG

Activating a filter similar to the above will result in a reduced data display similar to this:

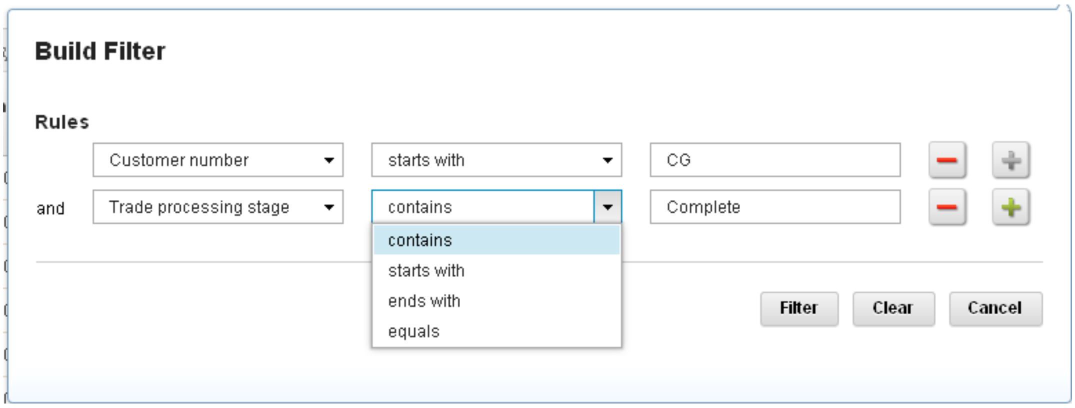


6 of 6 items shown

	Event time	Customer number	Trade number	Stock / Trade amount	Data	Errors	Trade processing stage
□	2013-01-02 14:06:37.715	CG123456	BNY347290	IBM $1000		-	Trade processing: Complete
□	2013-01-02 14:06:37.706	CG123456	BNY347290	IBM $1000		-	Trade reconciliation
□	2013-01-02 14:06:37.696	CG123456	BNY347290	IBM $1000		-	Gold customer: Processing tr
□	2013-01-02 14:06:37.689	CG123456	BNY347290	IBM $1000		-	Decision: Gold customer
□	2013-01-02 14:06:37.688	CG123456	BNY347290	IBM $1000		-	Deciding customer type
□	2013-01-02 14:06:37.677	CG123456	BNY347290			-	Trade instruction received

11. You can add additional filters to the display. Click Filter again, and click the green plus sign to add a second filter, as shown. Note that the values for each filter are case-sensitive.

    Click Filter to activate the new filter.

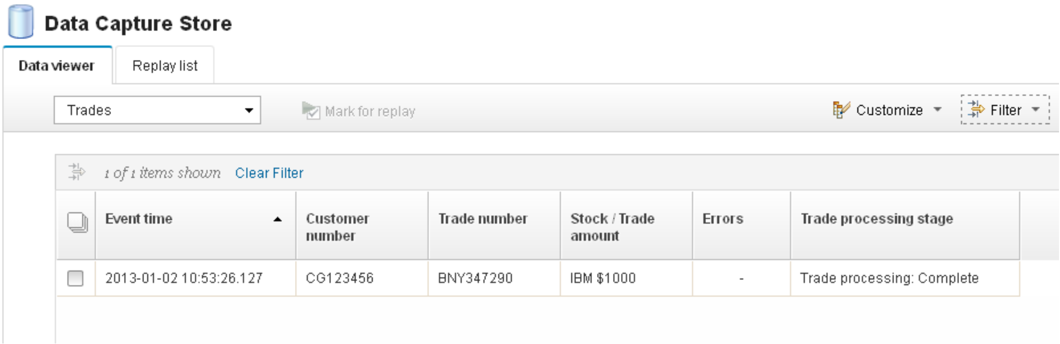


Build Filter

Rules

	Customer number	starts with	CG
and	Trade processing stage	contains	Complete

12. The data viewer will now show just one matching record.



Data Capture Store

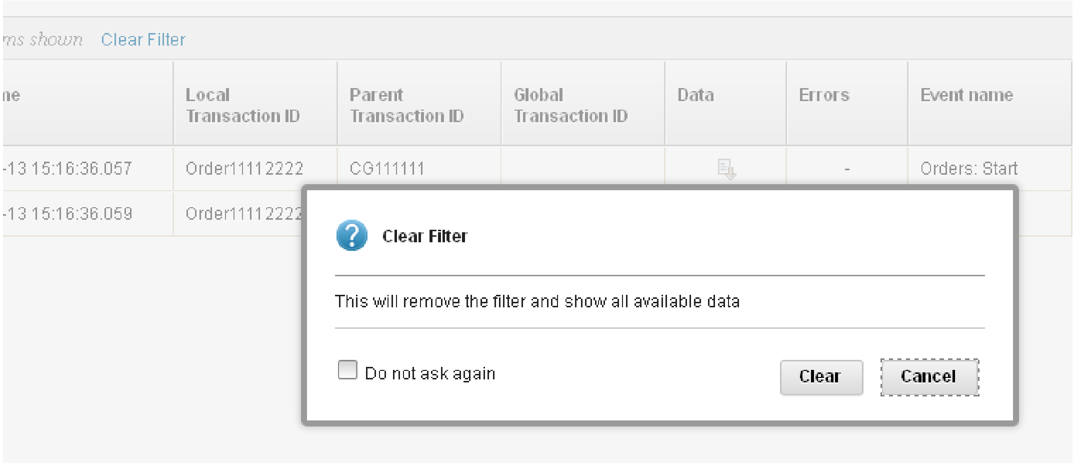
1 of 1 items shown Clear Filter

□	Event time	Customer number	Trade number	Stock / Trade amount	Errors	Trade processing stage
□	2013-01-02 10:53:26.127	CG123456	BNY347290	IBM $1000	-	Trade processing: Complete

13. Clear the filter.

    You can clear all filters by clicking “Clear Filter”.

    Click Clear to confirm, and optionally set the check box to not see the message again.



ms shown Clear Filter

ne	Local Transaction ID	Parent Transaction ID	Global Transaction ID	Data	Errors	Event name
-13 15:16:36.057	Order11112222	CG111111			-	Orders: Start
-13 15:16:36.059	Order1111222					

Clear Filter

This will remove the filter and show all available data

□ Do not ask again

4. Replaying Messages

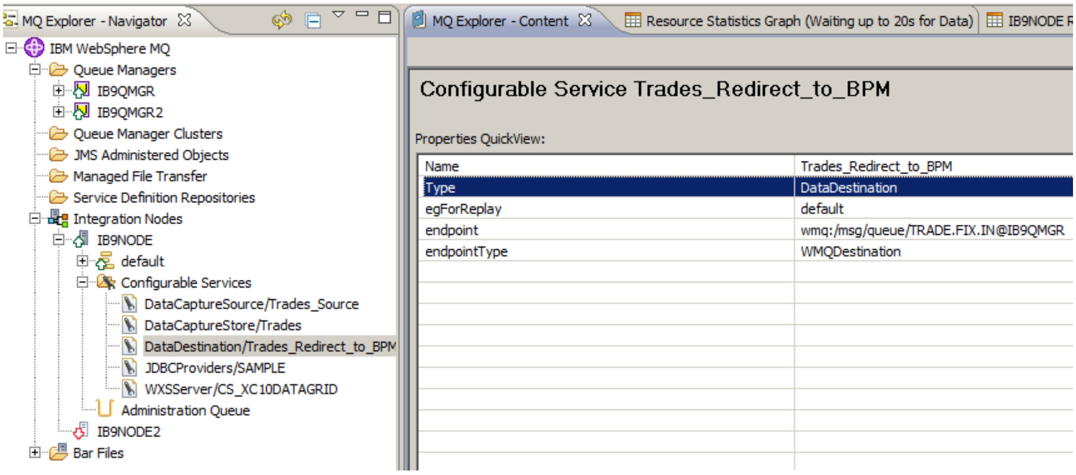
We will now configure the IB9NODE to allow messages to be replayed. This means that messages displayed on the web browser can be selected and resent to the same, or a different, message flow for further processing.

In this scenario, you will just replay the message by sending it to a separate MQ queue, although it will not be processed by any additional application.

1. First, review the configurable service that has been defined to enable the replay.

   In the IB Explorer, expand the IB9NODE, and expand Configurable Services. Select the appropriate Data Destination configurable service. You will see that this destination will enable messages to be routed to the TRADE.FIX.IN queue. In this example, this queue is processed by a simple message flow in the Trades application, but in another scenario it might be processed by a BPM application which could provide the facilities to make detailed amendments to the messages, prior to resending to the Trades application again.

   Click Finish.

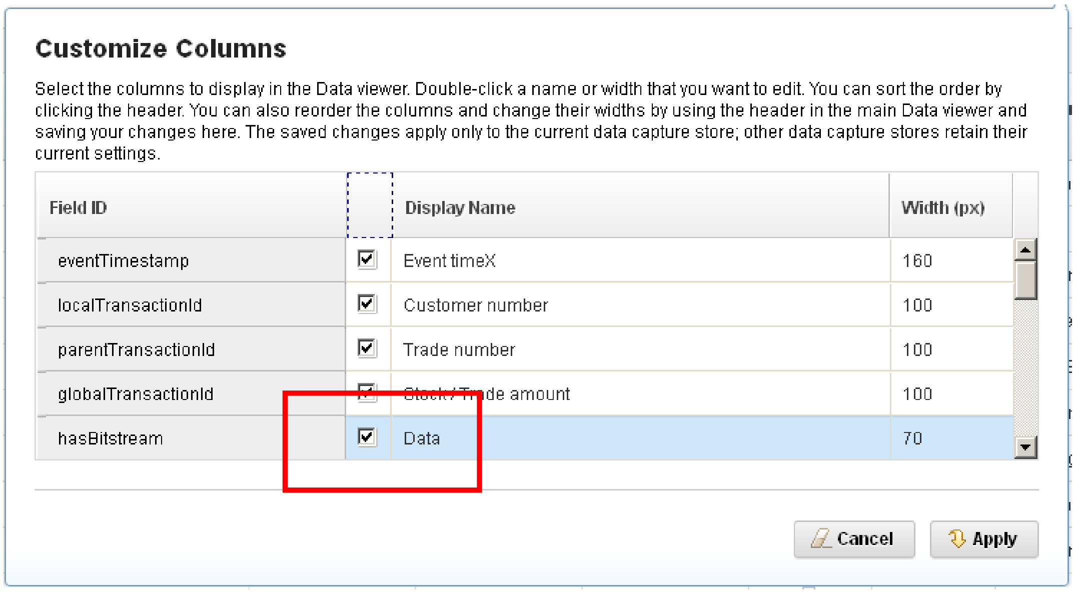


Configurable Service Trades_Redirect_to_BPM

Properties QuickView:

Name	Trades_Redirect_to_BPM
Type	DataDestination
egForReplay	default
endpoint	wmq:/msg/queue/TRADE.FIX.IN@IB9QMGR
endpointType	WMQDestination

2. In the web browser Data Viewer, again select the Customise button. Check the “hasBitstream” item, and click Apply.

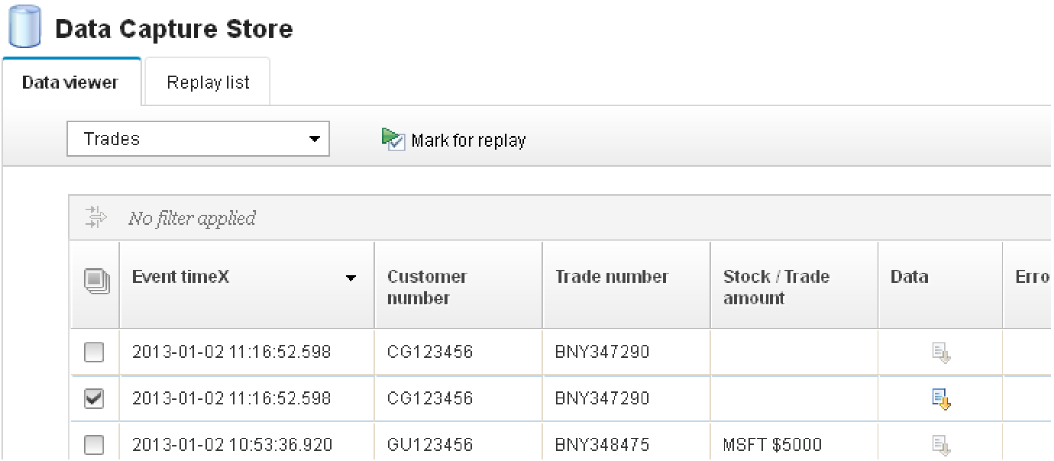


Customize Columns

Select the columns to display in the Data viewer. Double-click a name or width that you want to edit. You can sort the order by clicking the header. You can also reorder the columns and change their widths by using the header in the main Data viewer and saving your changes here. The saved changes apply only to the current data capture store; other data capture stores retain their current settings.

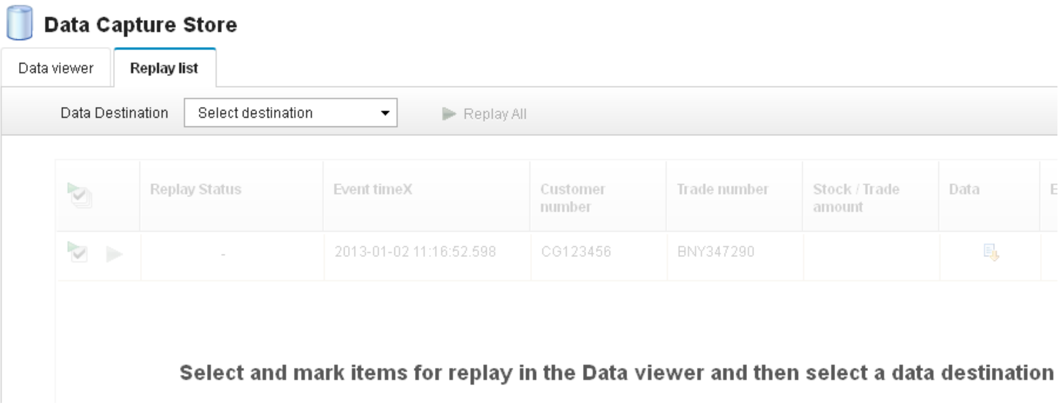
Field ID		Display Name	Width (px)
eventTimestamp	☑	Event timeX	160
localTransactionId	☑	Customer number	100
parentTransactionId	☑	Trade number	100
globalTransactionId	☑	Stock / Trade amount	100
hasBitstream	☑	Data	70

3. Select some of the messages for replay by placing a tick in the checkbox and clicking the “Mark for replay” button, which will now be active. Make sure at least one of the selected messages shows the coloured bitstream icon.



☐	Event timeX	Customer number	Trade number	Stock / Trade amount	Data	Erro
☐	2013-01-02 11:16:52.598	CG123456	BNY347290			
☑	2013-01-02 11:16:52.598	CG123456	BNY347290			
☐	2013-01-02 10:53:36.920	GU123456	BNY348475	MSFT $5000		

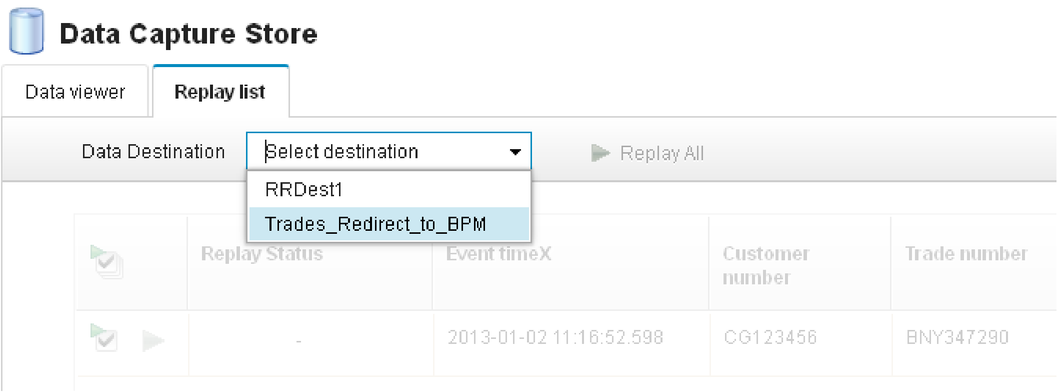
4. Clicking the “Mark for replay” button will take you to the Replay List. However, you will still not be able to invoke the Replay function.



	Replay Status	Event timeX	Customer number	Trade number	Stock / Trade amount	Data	E
	-	2013-01-02 11:16:52.598	CG123456	BNY347290			

Select and mark items for replay in the Data viewer and then select a data destination

5. On the Data Destination, select the destination Trades_Redirect_to_BPM.



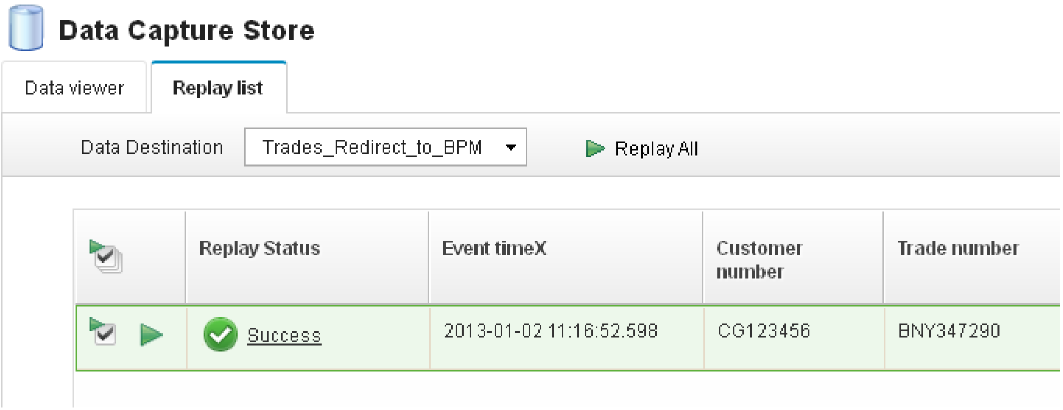
Data Destination: Select destination

	Replay Status	Event timeX	Customer number	Trade number
	-	2013-01-02 11:16:52.598	CG123456	BNY347290

6. On the Replay List, click Replay All (or you can replay each item individually by clicking the green arrow against each item).

You will see that the item that contained the data bitstream has been successfully sent to the replay destination. Note however, that this does not mean that it has been successfully processed by that application; it has simply been sent to the receiving destination.

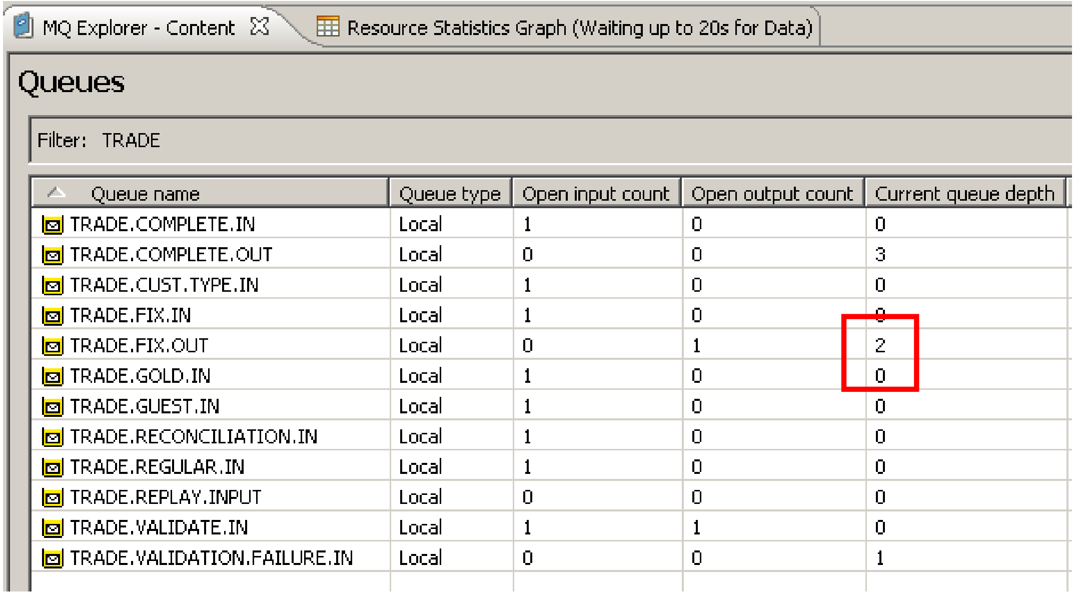
Items that did not contain a data bitstream cannot be replayed, as described by the error message seen on the Replay List view.



Data Destination: Trades_Redirect_to_BPM

	Replay Status	Event timeX	Customer number	Trade number
	✓ Success	2013-01-02 11:16:52.598	CG123456	BNY347290

7. Confirm the messages have been sent to the Replay queue. In MQ Explorer, select Queues under IB9QMGR. The queue depth of TRADES.FIX.OUT should have increased by 2 (or however many messages you sent for replay).



Queues

Filter: TRADE

Queue name	Queue type	Open input count	Open output count	Current queue depth
TRADE.COMPLETE.IN	Local	1	0	0
TRADE.COMPLETE.OUT	Local	0	0	3
TRADE.CUST.TYPE.IN	Local	1	0	0
TRADE.FIX.IN	Local	1	0	0
TRADE.FIX.OUT	Local	0	1	2
TRADE.GOLD.IN	Local	1	0	0
TRADE.GUEST.IN	Local	1	0	0
TRADE.RECONCILIATION.IN	Local	1	0	0
TRADE.REGULAR.IN	Local	1	0	0
TRADE.REPLAY.INPUT	Local	0	0	0
TRADE.VALIDATE.IN	Local	1	1	0
TRADE.VALIDATION.FAILURE.IN	Local	0	0	1

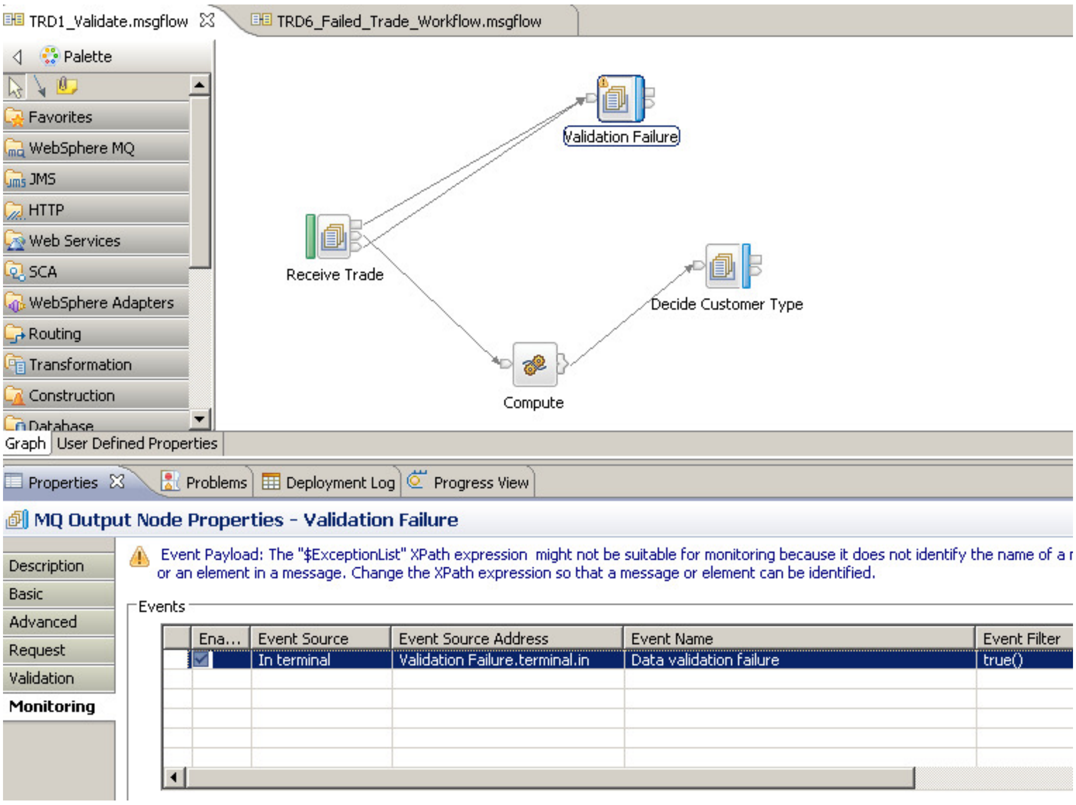
8. Depending on the nature of the message flow, and the types of events that you decide are eligible for replaying, the replay queue could be the same input queue that the original message flow uses, or a separate queue and a separate (and different) message flow.

5. Failed Messages

If a message flow encounters an error during processing, this can be captured and reported using the web browser.

To do this, the monitoring point on the message flow node must be configured to include the $ExceptionList in the monitoring event message. The TRD1_Validate message flow has been configured in this way.

1. In the Broker Toolkit, click the Validation Failure node.



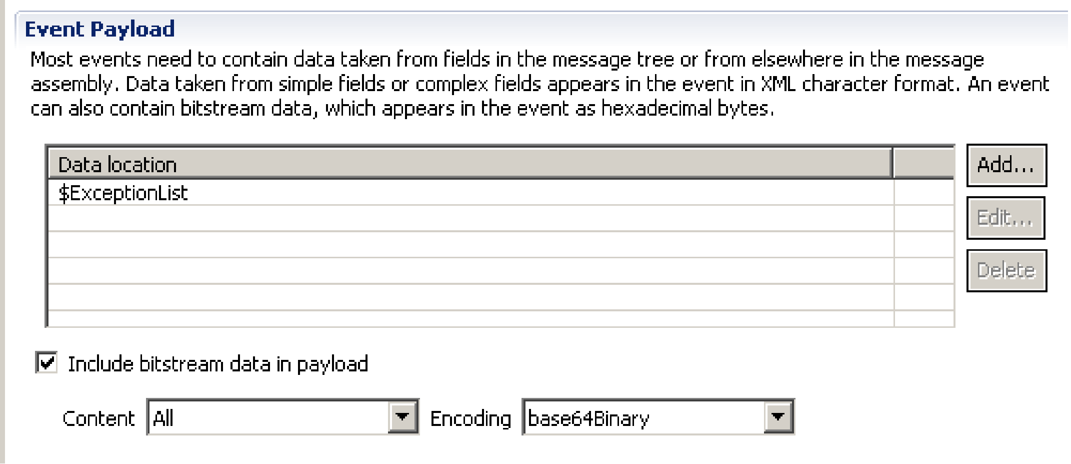
MQ Output Node Properties - Validation Failure

⚠ Event Payload: The "$ExceptionList" XPath expression might not be suitable for monitoring because it does not identify the name of a r
or an element in a message. Change the XPath expression so that a message or element can be identified.

Events

Ena...	Event Source	Event Source Address	Event Name	Event Filter
☑	In terminal	Validation Failure.terminal.in	Data validation failure	true()

2. In the Monitoring properties, click Edit, and in the Event Payload section, you will see that the Data Location has been configured to include the $ExceptionList.



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	
$ExceptionList	

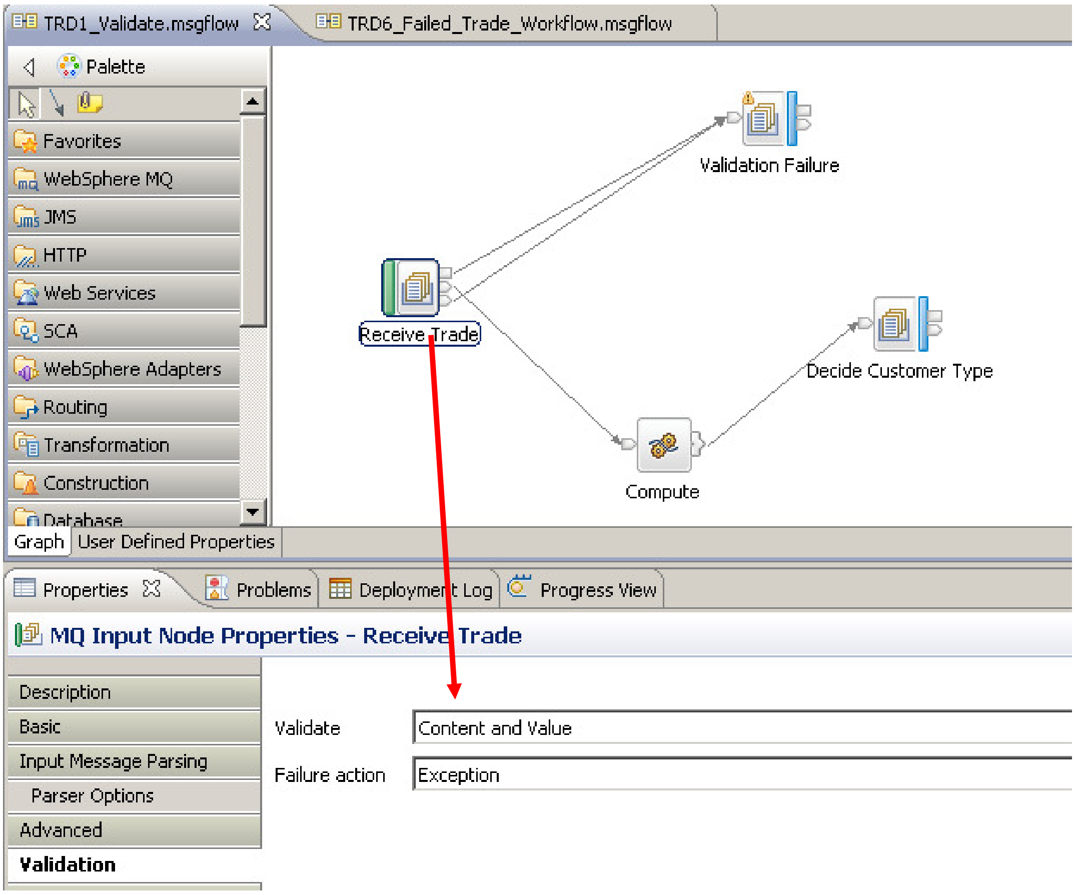
☑ Include bitstream data in payload

Content: All

Encoding: base64Binary

3. Using RFHUtil, send further events to the Trades application.

Open the file c:\student\Trades\data\TradeMessage_BadMessage.xml to the queue TRADE.VALIDATE.IN. Although this message is a valid XML message, it has one of the required xml elements missing, and will fail validation on the ReceiveTrade node.

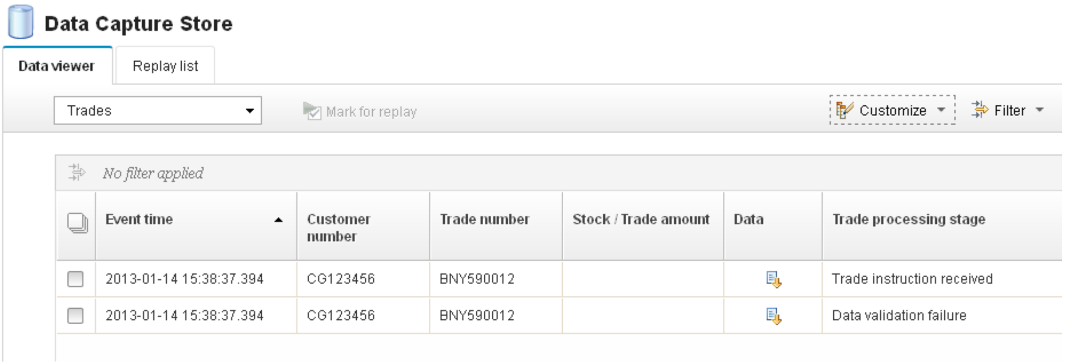


MQ Input Node Properties - Receive Trade

Validate: Content and Value

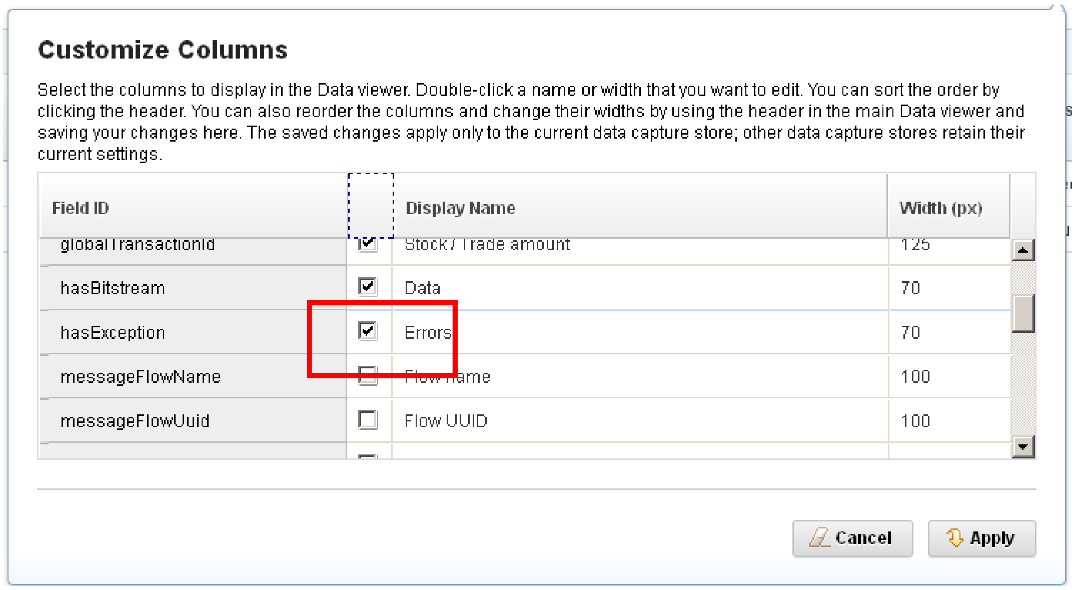
Failure action: Exception

4. Back in the web browser, again with the record3 user, you will see two new entries like this (note earlier entries have been removed for clarity).



□	Event time ▲	Customer number	Trade number	Stock / Trade amount	Data	Trade processing stage
□	2013-01-14 15:38:37.394	CG123456	BNY590012			Trade instruction received
□	2013-01-14 15:38:37.394	CG123456	BNY590012			Data validation failure

5. Failed events are highlighted by customizing the displayed columns. Click the Customize button, and (if not already check marked) tick the "hasException" box. Click Apply.

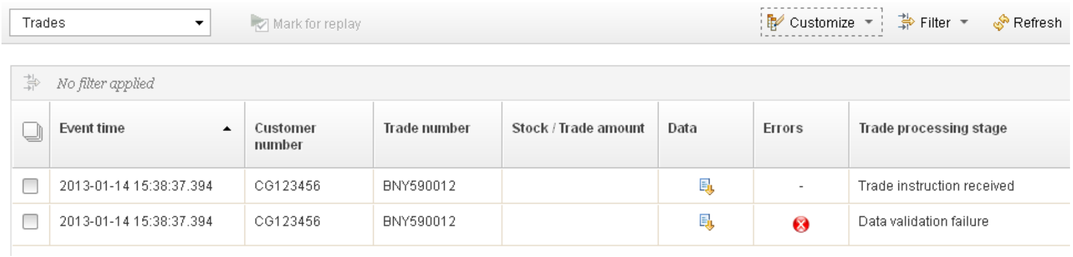


Customize Columns

Select the columns to display in the Data viewer. Double-click a name or width that you want to edit. You can sort the order by clicking the header. You can also reorder the columns and change their widths by using the header in the main Data viewer and saving your changes here. The saved changes apply only to the current data capture store; other data capture stores retain their current settings.

Field ID		Display Name	Width (px)
globalTransactionId	☑	Stock / Trade amount	125
hasBitstream	☑	Data	70
hasException	☑	Errors	70
messageFlowName	□	Flow name	100
messageFlowUuid	□	Flow UUID	100

The Errors column will show a red cross for all monitoring events that contain the $ExceptionList data.



□	Event time ▲	Customer number	Trade number	Stock / Trade amount	Data	Errors	Trade processing stage
□	2013-01-14 15:38:37.394	CG123456	BNY590012			-	Trade instruction received
□	2013-01-14 15:38:37.394	CG123456	BNY590012			⊗	Data validation failure

6. This concludes the Record and Replay – Trades Example Lab Guide