

A cluster of colorful gears in various sizes and colors (red, green, blue, yellow, purple) is located in the top-left corner of the page.

**betaWorks**

**IBM Integration Bus**

## Business Transaction Monitoring

Featuring:

- Flow Monitoring Events
- Configuring a BTM
- Viewing Business Transaction instances and events
- Securing access to BTM instances using file-based access controls

**June 2016**

Hands-on lab built at product  
Version 10.0.0.5

<b>1. INTRODUCTION</b> .....	<b>3</b>
1.1 SCENARIO .....	3
<b>2. PREPARE THE IIB NODE</b> .....	<b>4</b>
2.1 OPEN THE WINDOWS LOG MONITOR FOR IIB .....	4
2.2 CONFIGURE THE IIB NODE FOR BUSINESS TRANSACTION MONITORING .....	5
2.2.1 <i>Define an associated queue manager for the IIB node</i> .....	5
2.2.2 <i>Configure the BTM components</i> .....	7
2.2.3 <i>Define the MQ resources</i> .....	7
<b>3. PREPARE THE APPLICATION</b> .....	<b>8</b>
3.1 IMPORT THE APPLICATION .....	8
3.2 INVESTIGATE THE APPLICATION .....	9
<b>4. CONFIGURE THE BUSINESS TRANSACTION DEFINITION</b> .....	<b>14</b>
4.1 DEFINE THE BTD .....	14
4.2 CONFIGURE THE BTM DATABASE .....	18
<b>5. TEST THE BUSINESS TRANSACTION DEFINITION</b> .....	<b>20</b>
5.1 ACTIVATE FLOW MONITORING FOR THE APPLICATION .....	20
5.2 EXECUTE THE BUSINESS TRANSACTION .....	21
<b>6. ADD SECURITY ROLES FOR WEB USERS</b> .....	<b>28</b>
<b>END OF LAB GUIDE</b> .....	<b>32</b>

# 1. Introduction

Business Transaction Monitoring (BTM) enables the monitoring of a transaction through various IIB flows deployed on an integration node, to track and report the lifecycle of a message as it is processed by IIB.

A business transaction is a unit of function that can be seen as a unit from a business point of view, like a purchase, a booking, an auction. It is not a transaction in technical sense, neither a one nor a two-phase transaction, but more like a till transaction. A business transaction instance is a specific order or booking, usually identified by a reference number or ID, such as "order123456".

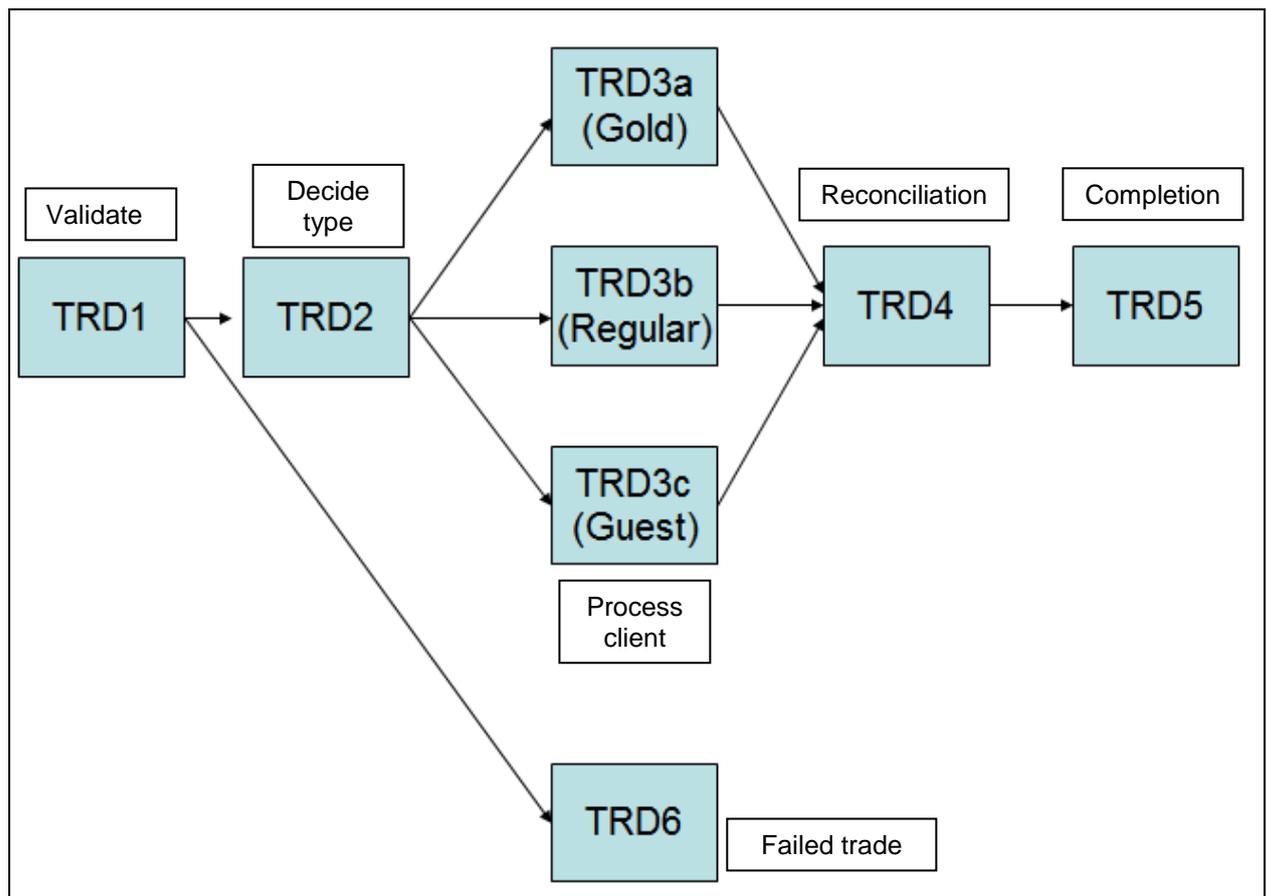
## 1.1 Scenario

This lab will create and use a business transaction based on the message flows contained within a single application. Schematically, the business transaction can be illustrated as shown below, and is based on a Trade Order scenario.

Each business transaction consists primarily of 5 message flows. The client type is either Gold, Regular or Guest. TRD2 examines the message, determines the client type, and routes the message accordingly. MQ queues are used to pass messages between each message flow.

If TRD1 detects a problem with the incoming message, it is routed directly to TRD6.

The message flows are very simple, and do not contain any significant logic. They do however contain a number of event monitoring points, so that each stage of the overall business transaction can be identified and tracked.

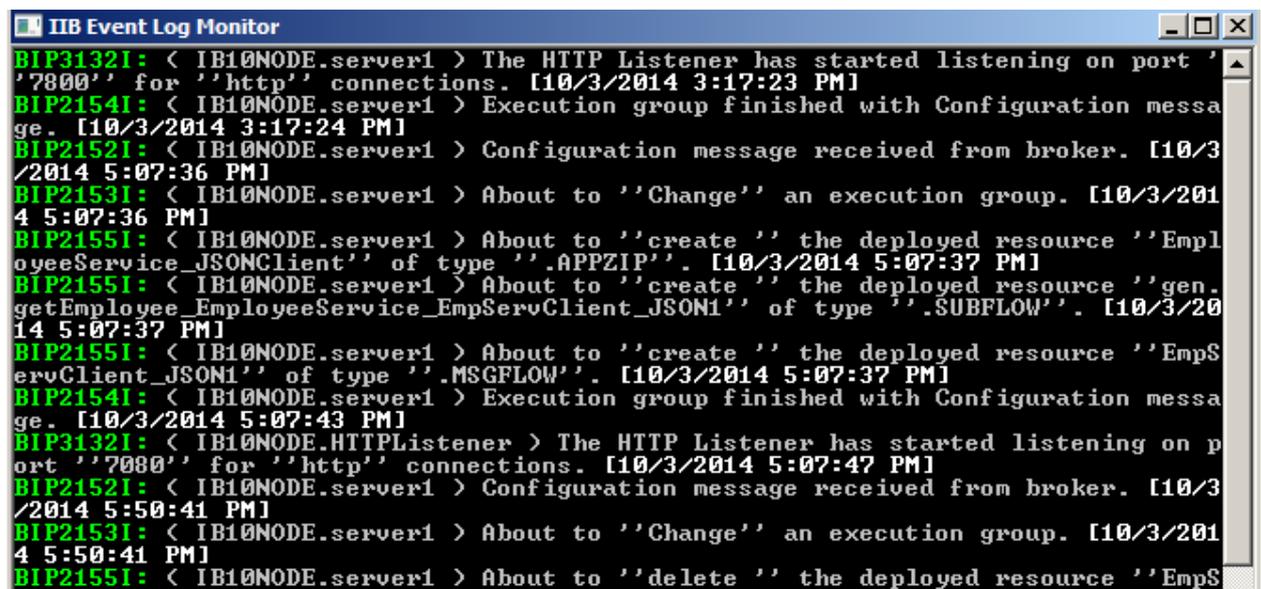


## 2. Prepare the IIB Node

### 2.1 Open the Windows Log Monitor for IIB

A useful tool for IIB development on Windows is the IIB Log Viewer. This tool continuously monitors the Windows Event Log, and all messages from the log are displayed immediately.

From the Start menu, click IIB Event Log Monitor. The Monitor will open; it is useful to have this always open in the background.



```
IIB Event Log Monitor
BIP31321: < IB10NODE.server1 > The HTTP Listener has started listening on port '
'7800' for 'http' connections. [10/3/2014 3:17:23 PM]
BIP21541: < IB10NODE.server1 > Execution group finished with Configuration messa
ge. [10/3/2014 3:17:24 PM]
BIP21521: < IB10NODE.server1 > Configuration message received from broker. [10/3
/2014 5:07:36 PM]
BIP21531: < IB10NODE.server1 > About to 'Change' an execution group. [10/3/201
4 5:07:36 PM]
BIP21551: < IB10NODE.server1 > About to 'create' the deployed resource 'Empl
oyeeService_JSONClient' of type '.APPZIP'. [10/3/2014 5:07:37 PM]
BIP21551: < IB10NODE.server1 > About to 'create' the deployed resource 'gen.
getEmployee_EmployeeService_EmpServClient_JSON1' of type '.SUBFLOW'. [10/3/20
14 5:07:37 PM]
BIP21551: < IB10NODE.server1 > About to 'create' the deployed resource 'EmpS
ervClient_JSON1' of type '.MSGFLOW'. [10/3/2014 5:07:37 PM]
BIP21541: < IB10NODE.server1 > Execution group finished with Configuration messa
ge. [10/3/2014 5:07:43 PM]
BIP31321: < IB10NODE.HTTPListener > The HTTP Listener has started listening on p
ort '7800' for 'http' connections. [10/3/2014 5:07:47 PM]
BIP21521: < IB10NODE.server1 > Configuration message received from broker. [10/3
/2014 5:50:41 PM]
BIP21531: < IB10NODE.server1 > About to 'Change' an execution group. [10/3/201
4 5:50:41 PM]
BIP21551: < IB10NODE.server1 > About to 'delete' the deployed resource 'EmpS
```

This tool is not shipped as part of the IIB product; please contact us directly if you would like a copy.

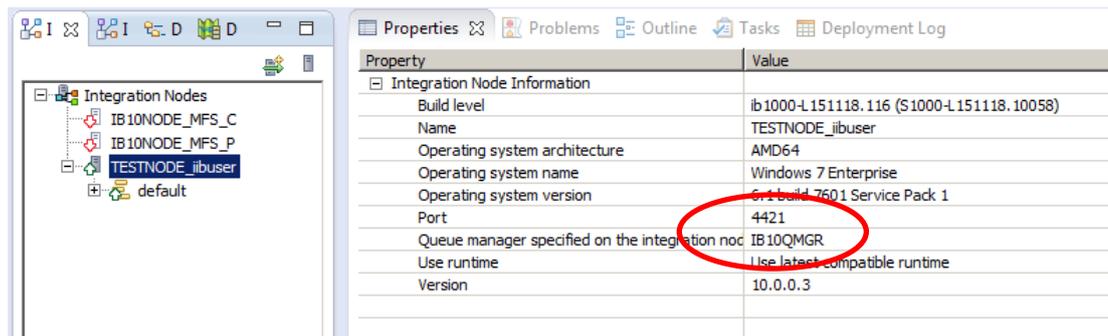
## 2.2 Configure the IIB node for Business Transaction Monitoring

### 2.2.1 Define an associated queue manager for the IIB node

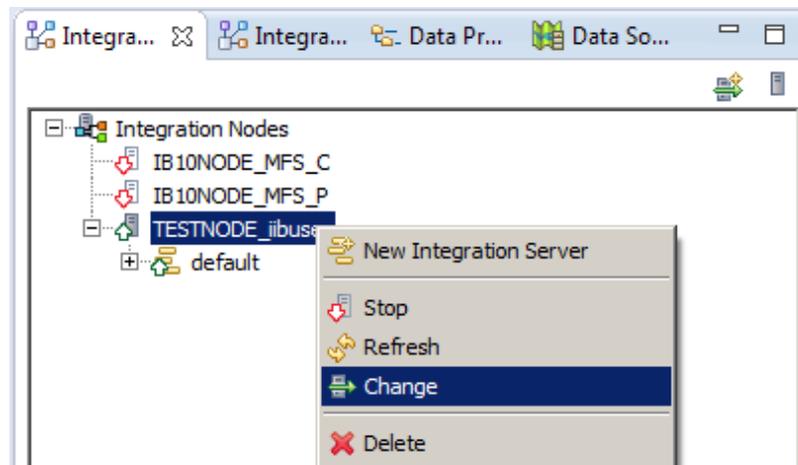
The Business Transaction Monitoring function requires the IIB node to have an associated queue manager. Make sure that the IIB node that you are using has a queue manager.

If you are using the TESTNODE\_iibuser in the workshop environment, perform the following steps.

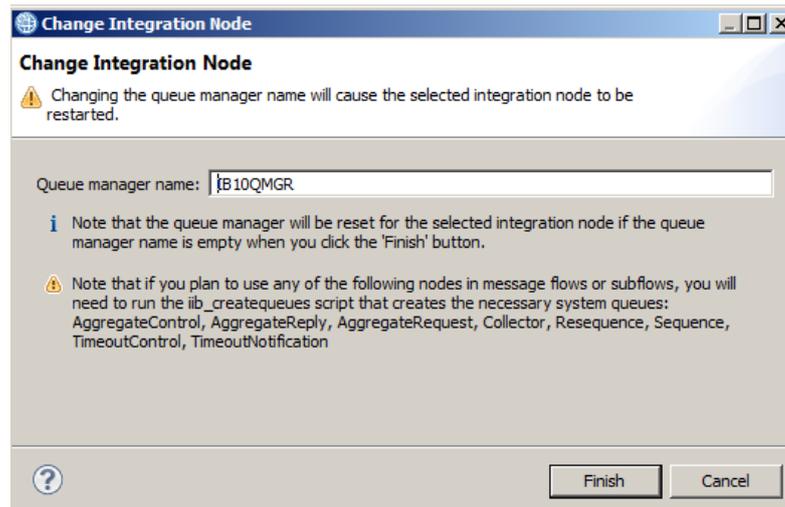
1. In the Toolkit IIB node view, in the Integration Nodes pane, highlight TESTNODE\_iibuser. In the Properties of the node, you will see the property "Queue manager specified on the integration node". If this is already set to a queue manager (eg. IB10QMGR), then you are all set.



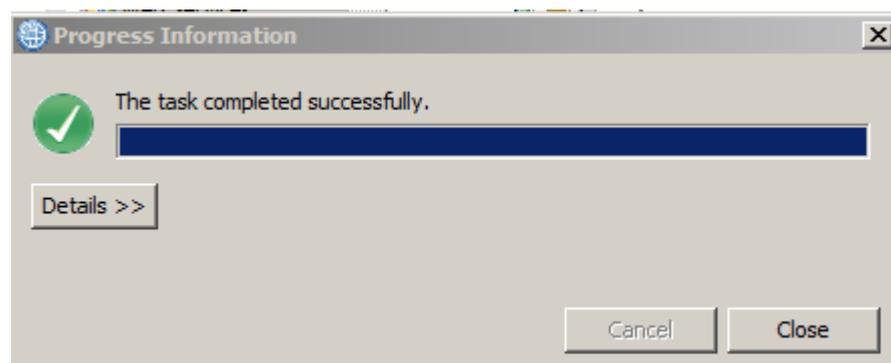
2. If a queue manager is not shown, right-click TESTNODE\_iibuser and select Change.



- Specify IB10QMGR as the queue manager name, and click Finish.



- This will automatically stop and restart the IIB node. When complete, you will see the following message.



## 2.2.2 Configure the BTM components

Two further items of configuration need to be done to enable BTM.

1. Activate BTM for the IIB node.

Execute the following command in an IIB Console:

```
mqsichangeproperties TESTNODE_iibuser
-o BrokerRegistry
-n productFunctionality
-v BUSINESS_TRANSACTION_MONITORING
```

2. Authorise access the BTM database, BTMDB, for the IIB node.

Execute the following command:

```
mqsisetdbparms TESTNODE_iibuser
-n BTMDB
-u iibadmin
-p passw0rd
```

Stop and restart the IIB node:

```
mqsistop TESTNODE_iibuser

mqsistart TESTNODE_iibuser
```

## 2.2.3 Define the MQ resources

Finally, the provided application requires several MQ queues, as well as some system queues for IIB. In a Windows DOS window (or IIB Console window), run the command:

```
c:\student10\BTM_Trades\commands\defineBTMqueues.cmd
```

## 3. Prepare the Application

### 3.1 Import the application

1. To avoid naming clashes with earlier labs, this lab will be developed using a new workspace.

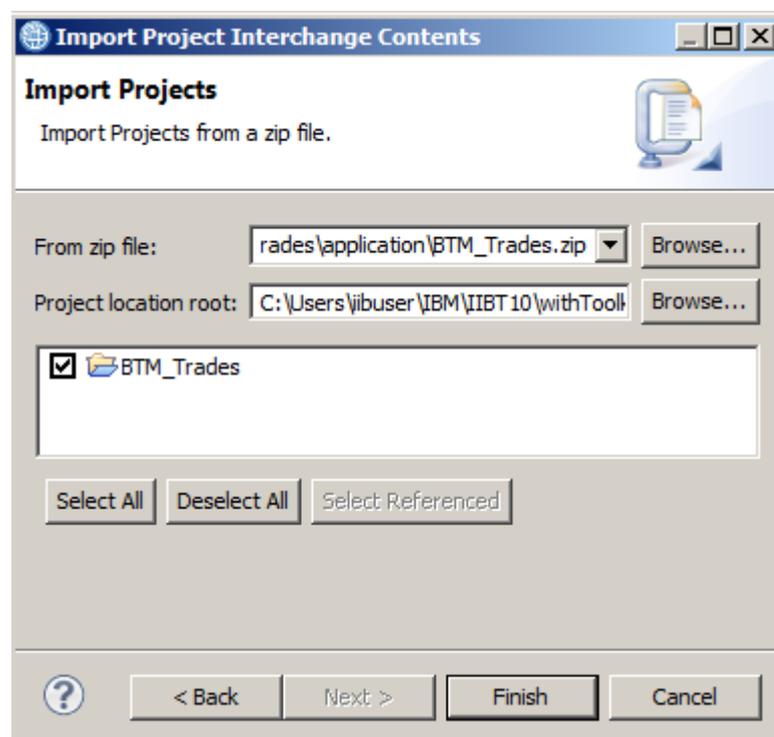
If you already have a workspace open, click File, Switch Workspace. Give the new workspace the name

```
c:\users\iibuser\IBM\IIB 10\workspace_BTM
```

2. Import the Project Interchange file

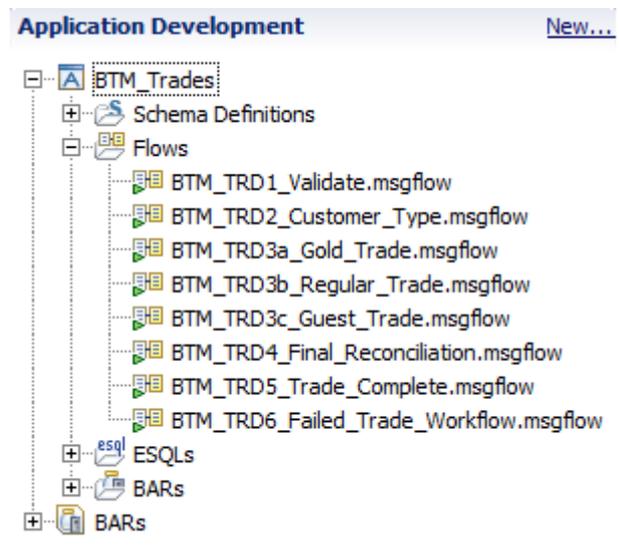
```
c:\student10 \ BTM_Trades \ application \ BTM_Trades.zip
```

This PI file contains one project. Ensure it is selected, and click Finish to import.



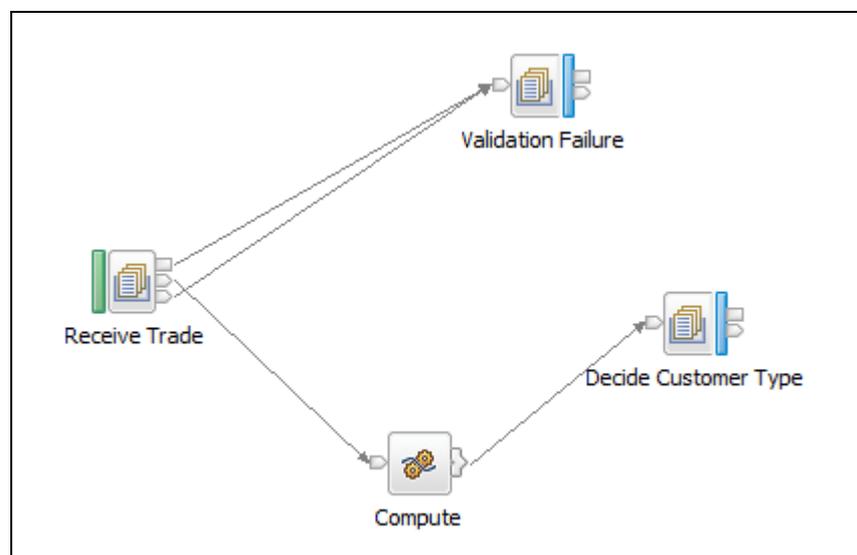
## 3.2 Investigate the application

1. The application contains a number of message flows. Each flow is quite simple, and is designed to pass the incoming message on to the next flow in the sequence of the business transaction.

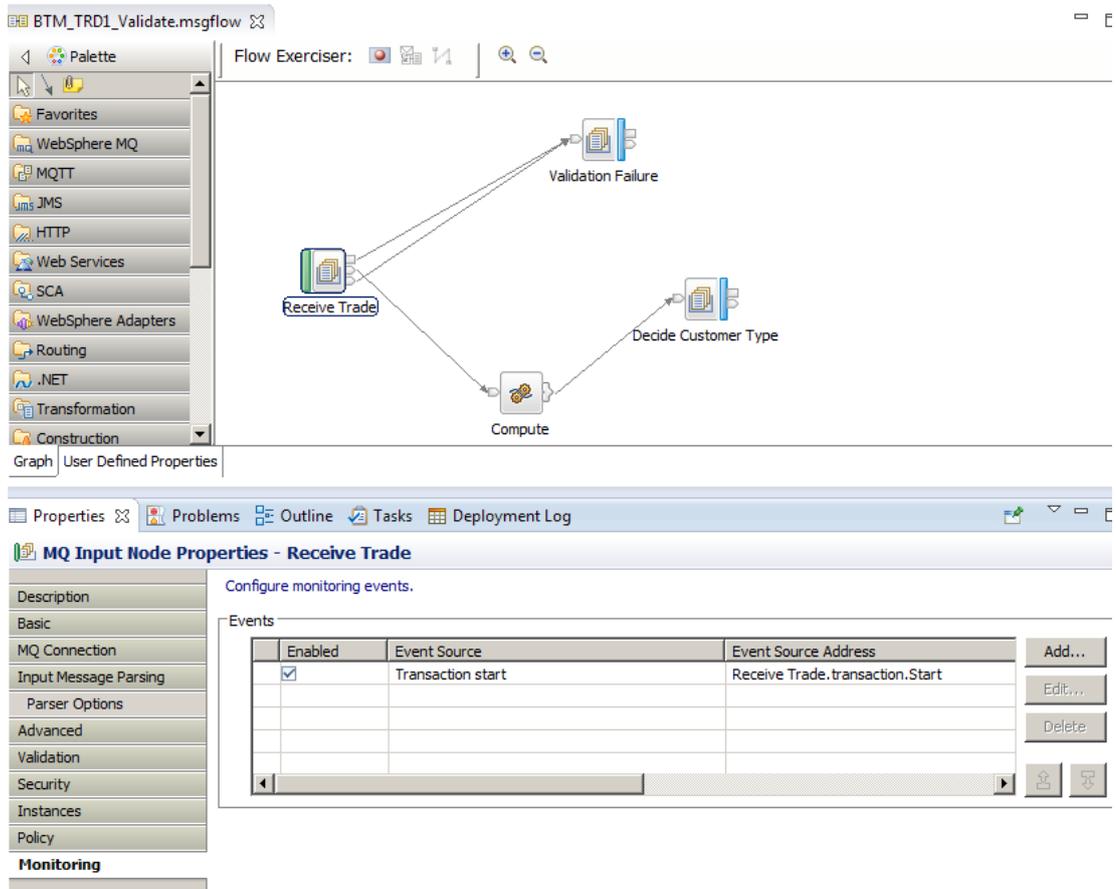


2. Most of the message flows have one or more monitoring events points associated with the nodes in the flow.

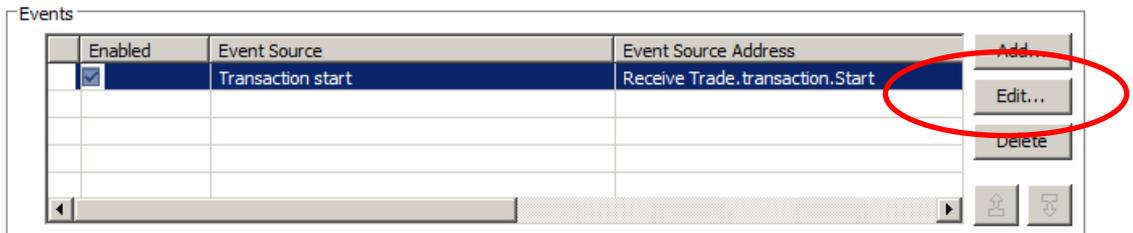
Open the BTM\_TRD1\_Validate message flow.



- Highlight the "Receive Trade" node, and look at the Properties of the node. Select the Monitoring tab. You will see that a monitoring event point has been defined on this node.



- Highlight the event, and then click the Edit button.



- The first tab, Basic, will be displayed. This is where the basic details of the event are configured. This event has been set on the start of the transaction, and has had the name of the event set as a literal.

**Edit event**

Basic | Correlation | Transaction

**Event Source**  
Select the source of the event.  
Transaction start

**Event Source Address**  
The integration node 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 instruction received  
 Data location Edit...

**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() Edit...

**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	

Add... Edit... Delete

Include bitstream data in payload  
 Content All Encoding base64Binary

? OK Cancel

- Click the Correlation tab. The most important correlator for Business Transaction Monitoring is the Global Transaction Correlator. This value enables the BTM component in IIB to connect the events from different message flows, in order to provide a status of the business transaction.

In this case, the global transaction correlator has been set to an element contained within the message payload, the TradeOrderID element.

You can examine other monitoring options. When complete, click Cancel.

The screenshot shows the 'Edit event' dialog box with the 'Correlation' tab selected. The dialog has three tabs: 'Basic', 'Correlation', and 'Transaction'. The 'Event Correlation' section contains a description of event correlators and three sections for configuring correlators: Local, Parent, and Global. Each section has radio buttons for 'Automatic' and 'Specify location of correlator', a description box, and a text input field with an 'Edit...' button. The Global transaction correlator is selected.

**Edit event**

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 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.

Parent transaction correlator:

Automatic  Specify location of correlator

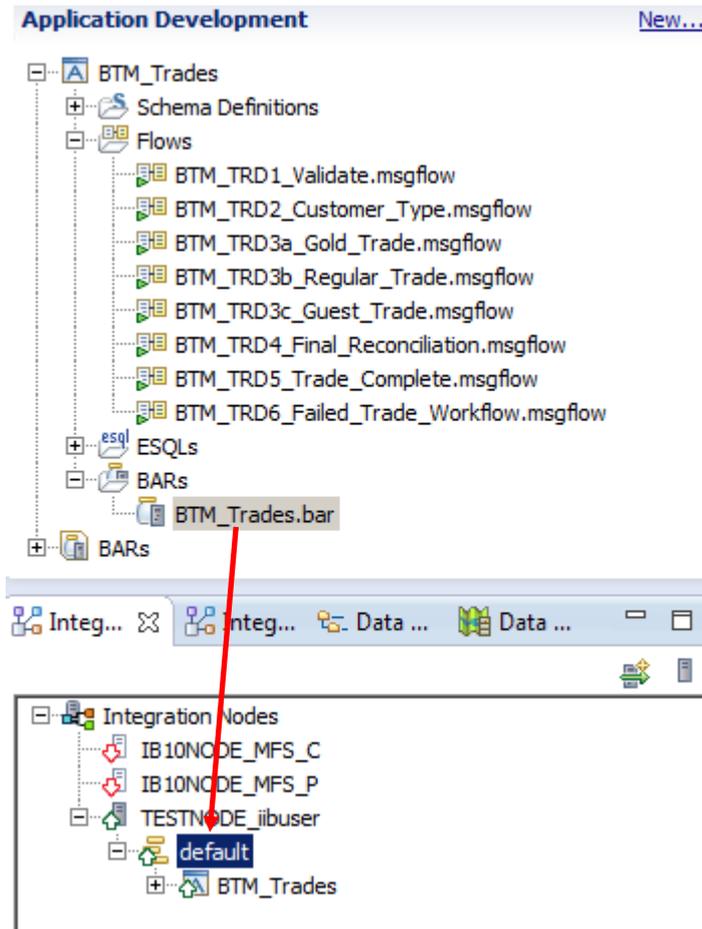
Description  
The parent correlator will be read from the specified location in the message tree. Ensure the specified location contains a suitable parent correlator value.

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.

- 7. Deploy the application to the IIB node (deploy the supplied barfile).



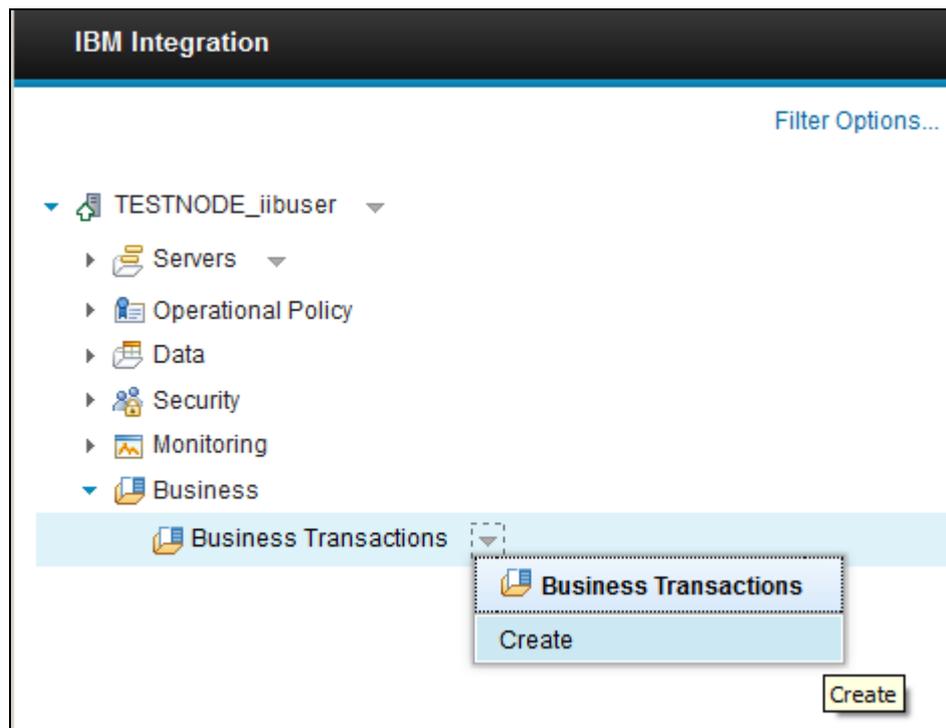
## 4. Configure the Business Transaction Definition

### 4.1 Define the BTD

1. Open a web browser and connect to the IIB node. (In the workshop VM, connect to betaworks-esb10 using the supplied shortcuts in Firefox. Depending on whether you have performed the Web Admin lab, you may need to use http(4414) or https(4421).

(If the IIB node has a different webadmin port to these, right-click the node in the Integration Toolkit, and allow the Toolkit to open a browser tab with the correct port).

Expand the Business folder, and click the context menu dropdown on the Business Transactions item. Click Create.



2. Provide a name, BTD\_Trades, and click OK.

### New Business Transaction Definition

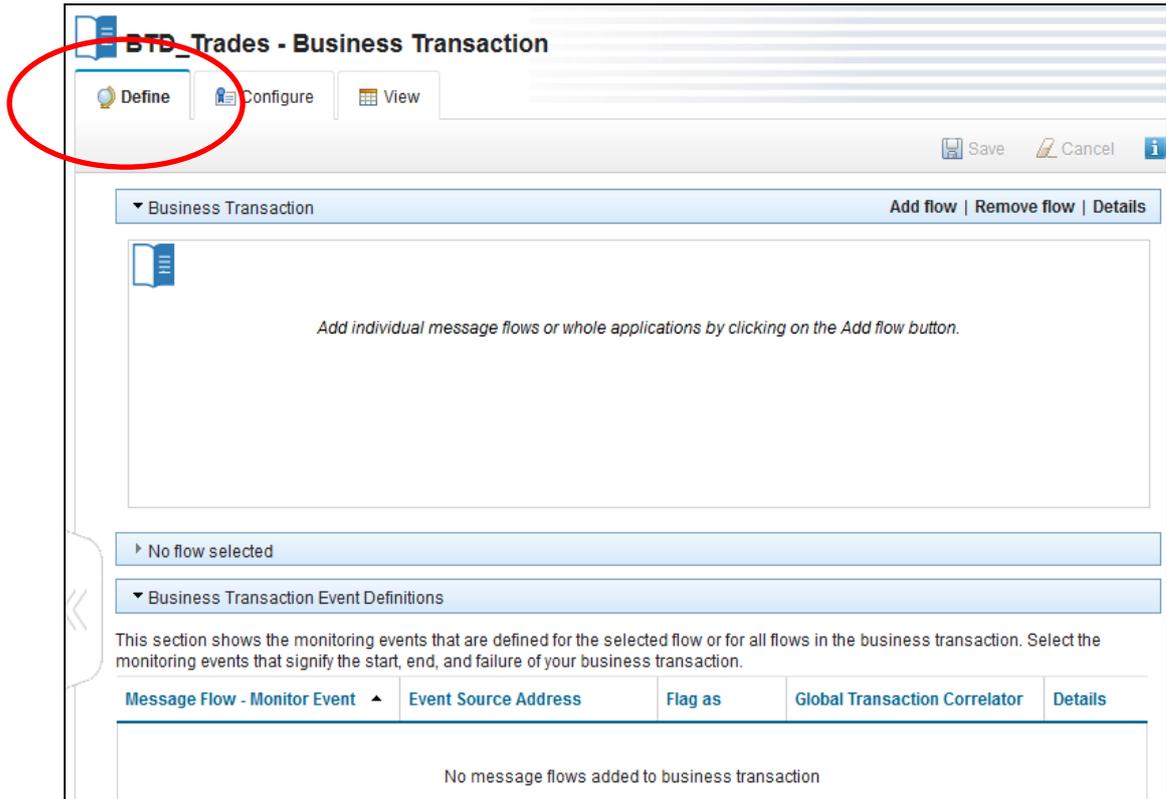
Enter a new name for the business transaction definition.

---

Business Transaction Name:

---

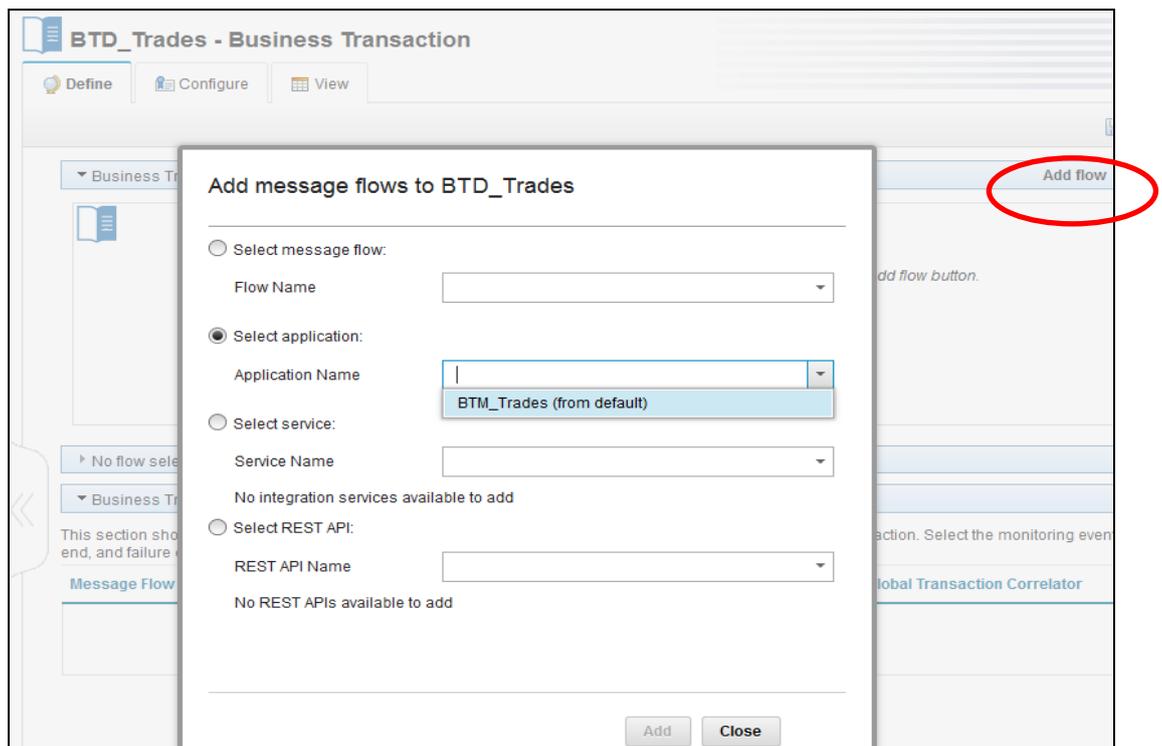
- The BTD will be shown, but no message flows have been added yet. You will be on the Define tab.



- Click "Add flow".

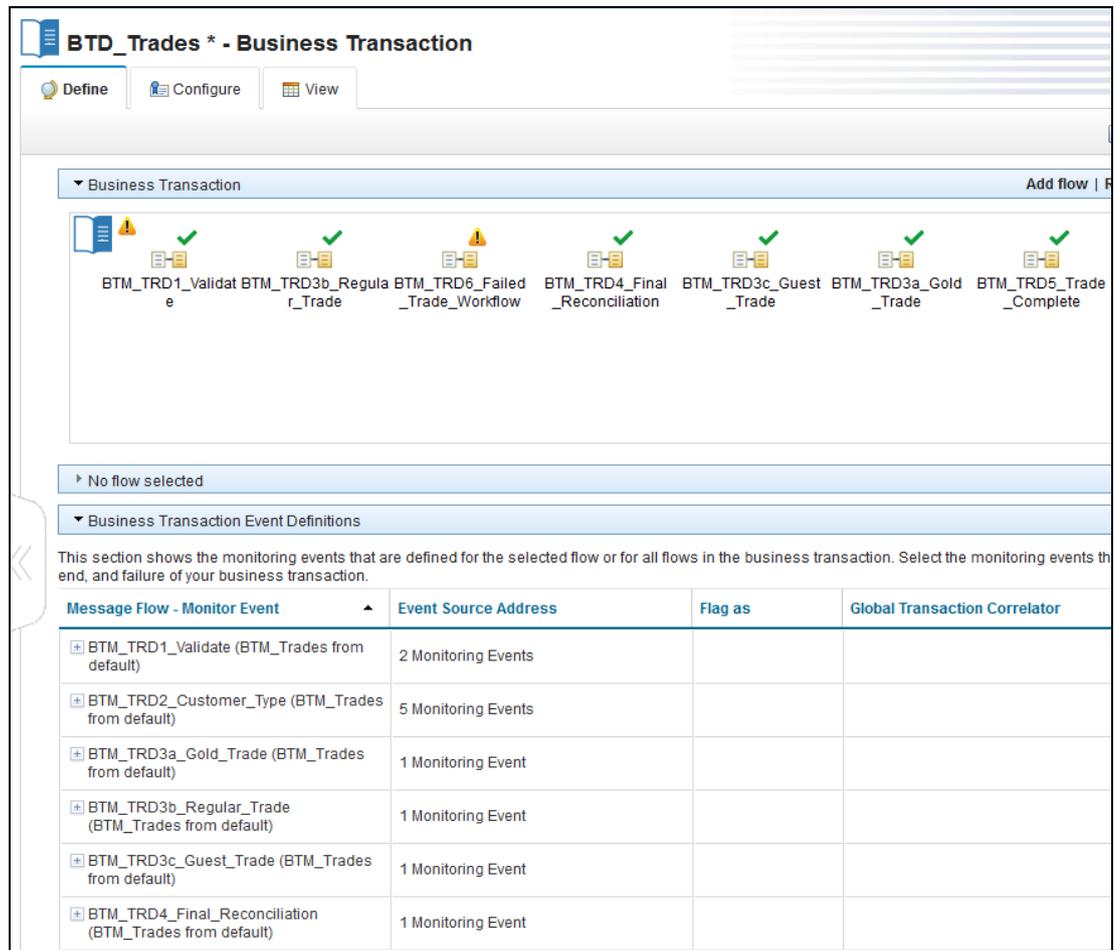
Select an application, and from the drop-down, select BTM\_Trades.

Click Add.



- The message flows will be added to the BTD, and the monitoring points that have been defined on the message flows will be listed on the bottom of the window.

The BTD define process has examined the deployed message flows, and has extracted details of the monitoring event points.



- Expand the first message flow, BTM\_TRD1\_Validate. You will see that two monitoring events have been defined on this flow.

Message Flow - Monitor Event	Event Source Address	Flag as	Global Transaction Correlator	Details
BTM_TRD1_Validate (BTM_Trades from default)	2 Monitoring Events			
Trade instruction received	Receive Trade.transaction.Start	Do not flag	\$Root/XMLNSC/tra:tradeOrder /tra:tradeOrderID	More details
Data validation failure	Validation Failure.terminal.in	Do not flag	\$Root/XMLNSC/tra:tradeOrder /tra:tradeOrderID	More details

- The first event for TRD1 will signify the start of the Business Transaction. In the "Flag As" column, click the drop-down, and select "Start" as the flag for this event.



- For the "Data validation failure" event, select Failure.

<input type="checkbox"/> BTM_TRD1_Validate (BTM_Trades from default)	2 Monitoring Events	
Trade instruction received	Receive Trade.transaction.Start	Start ▾
Data validation failure	Validation Failure.terminal.in	Failure ▾

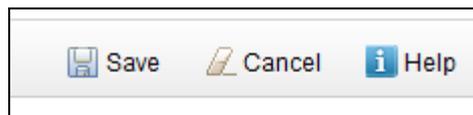
- For all the events up to and including TRD4, set the flag to Progress.

<input type="checkbox"/> BTM_TRD2_Customer_Type (BTM_Trades from default)	5 Monitoring Events	
Decision: Gold customer	GoldCustomer.terminal.in	Progress ▾
Decision: Guest customer	GuestCustomer.terminal.in	Progress ▾

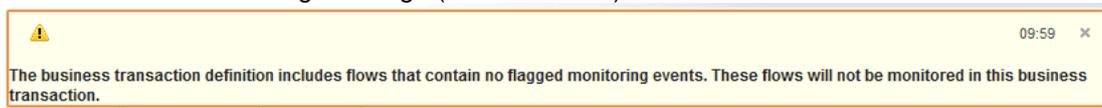
- For the TRD5 message flow, set the event flag to End.

<input type="checkbox"/> BTM_TRD5_Trade_Complete (BTM_Trades from default)	1 Monitoring Event	
Trade processing: Complete	Completion.transaction.Start	End ▾

- At the top of the window, click Save.

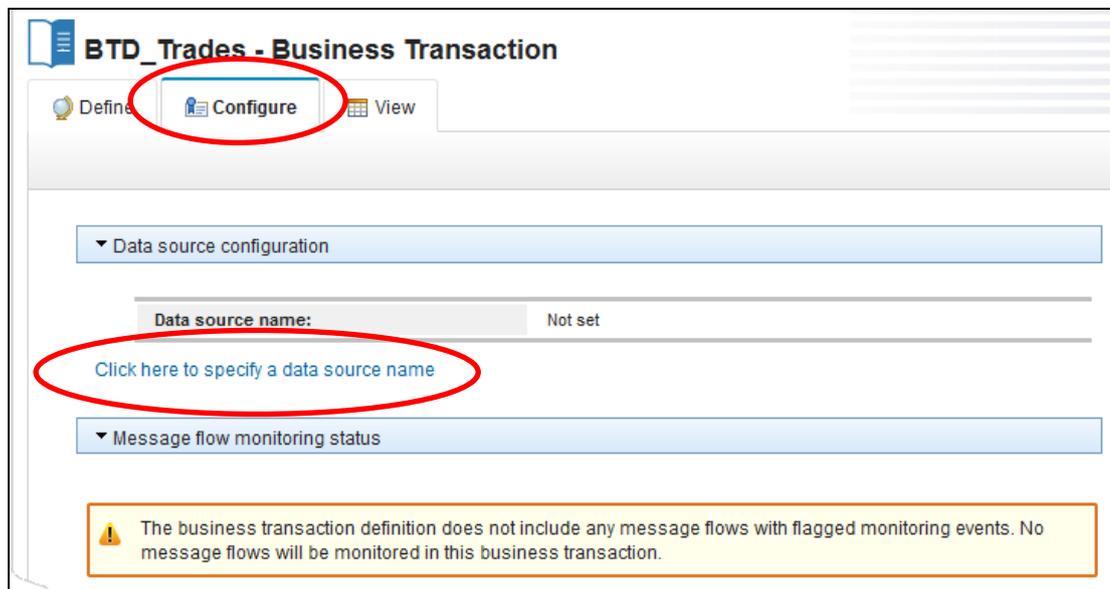


- You will see a warning message indicating that not all flows contain flagged monitoring events. Close this warning message (click the cross).

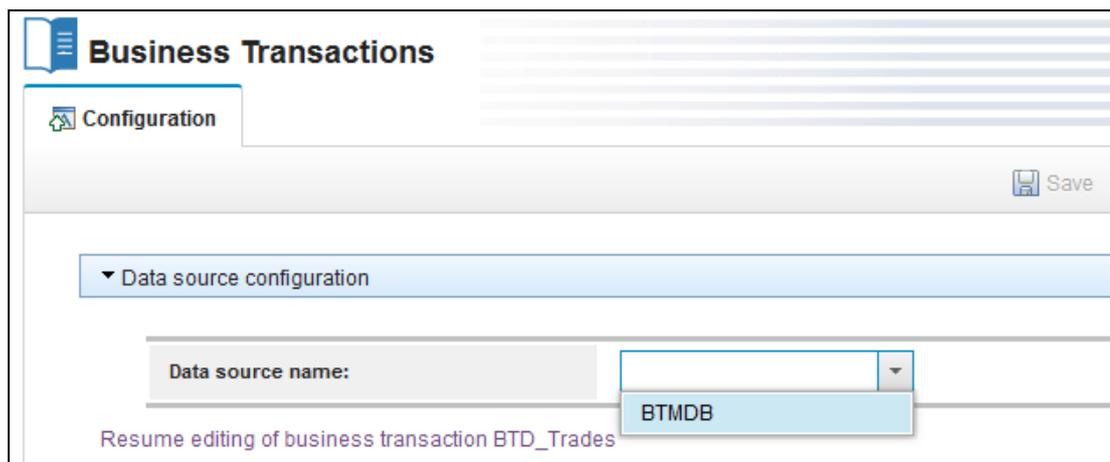


## 4.2 Configure the BTM database

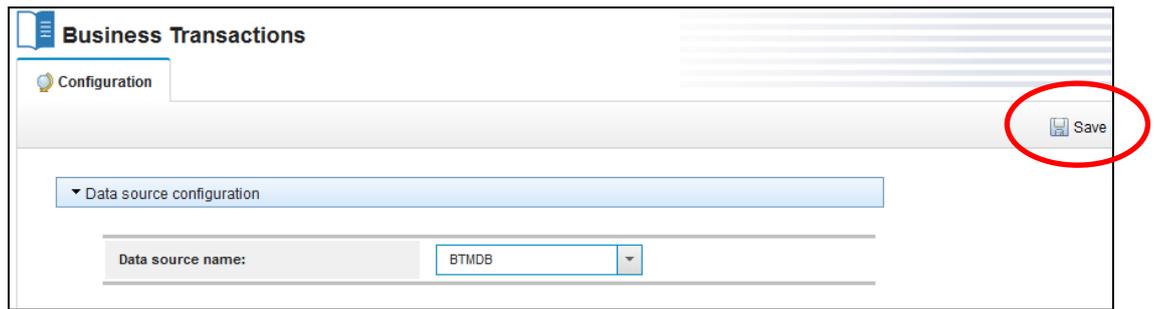
1. The BTM database needs to be specified to the IIB node. Click the Configure tab.  
You will see that the Data Source Name for the BTM database has not been set.  
Click to specify the data source name.



2. In the drop-down, select BTMDB. If BTMDB does not appear in the drop-down list, then something has gone wrong in the BTM enablement steps earlier in the lab.



3. Select BTMDB, and click Save.



The screenshot displays the 'Business Transactions' configuration interface. At the top, there is a 'Configuration' tab. Below it, a section titled 'Data source configuration' contains a dropdown menu. The dropdown is currently set to 'BTMDB'. To the right of the configuration area, a 'Save' button is highlighted with a red circle. The 'Save' button is represented by a floppy disk icon followed by the text 'Save'.

## 5. Test the Business Transaction Definition

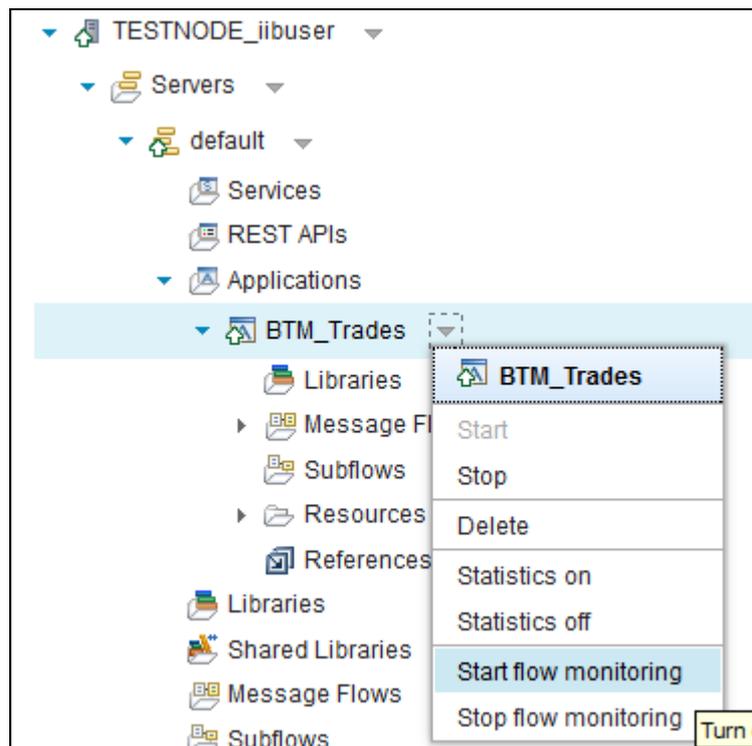
### 5.1 Activate Flow Monitoring for the Application

In order for any business event to emit monitoring data, flow monitoring must be enabled for each message flow that will form part of the business transaction.

In this lab, this can be achieved by activating flow monitoring at the application level; this will activate monitoring for all message flows contained within the application.

In the Web UI navigator, expand the Applications folder, and on the BTM\_Trades application, click the drop-down arrow.

Click "Start flow monitoring".



Flow monitoring will be activated.



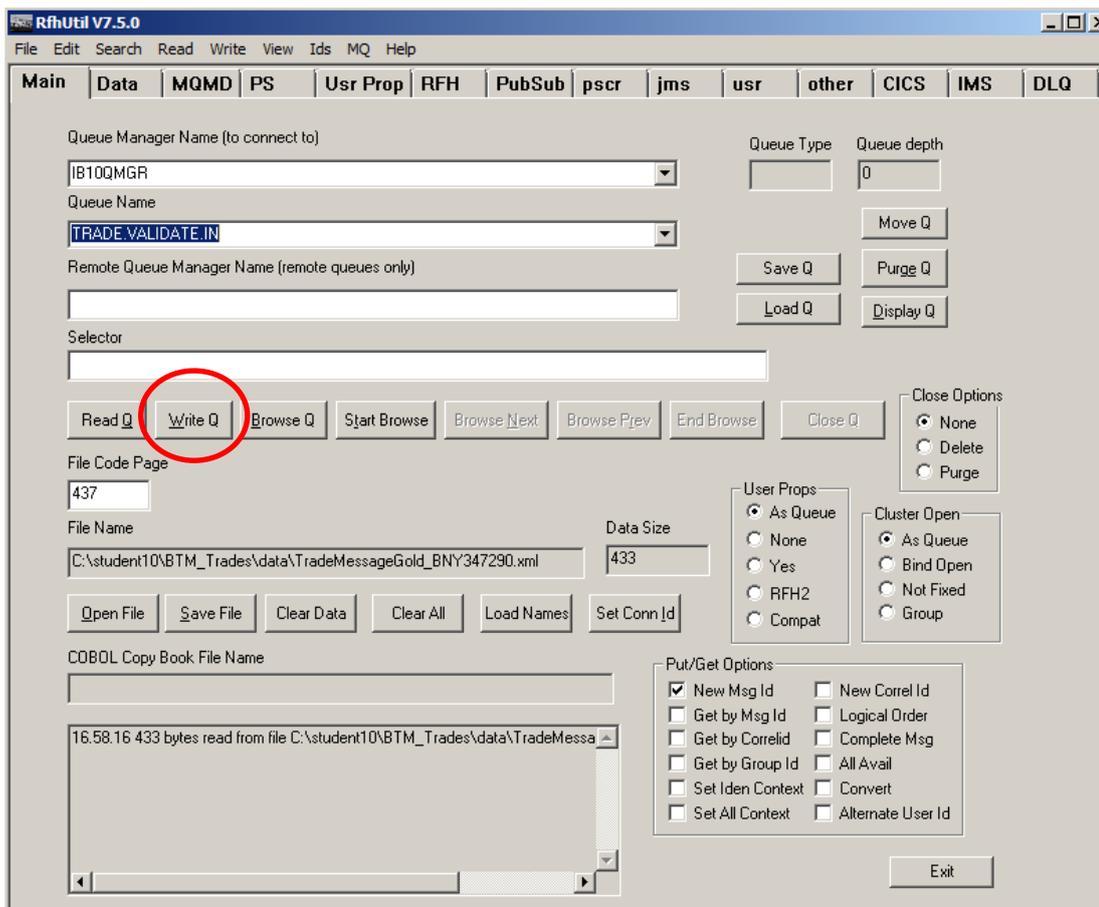
## 5.2 Execute the Business Transaction

Note - please follow the tests described in this guide. If you send messages with the same TradeOrderID more than once, you will see the status of the associated business event become "Inconsistent". This is not an error, but an indication that there is a problem with a particular business transaction. For example, events for a particular business transaction have been received after the business transaction status has been set to "End".

You will now send a number of messages to the application. These will result in different outcomes in the BTM which you will examine.

1. Open an instance of RFHUtil from the Start menu, and set the following values:
  - Queue Manager Name = IB10QMGR
  - Queue Name = TRADE.VALIDATE.IN
  - File Name = c:\student10\BTM\_Trades\data\TradeMessageGold\_BNY347290.xml

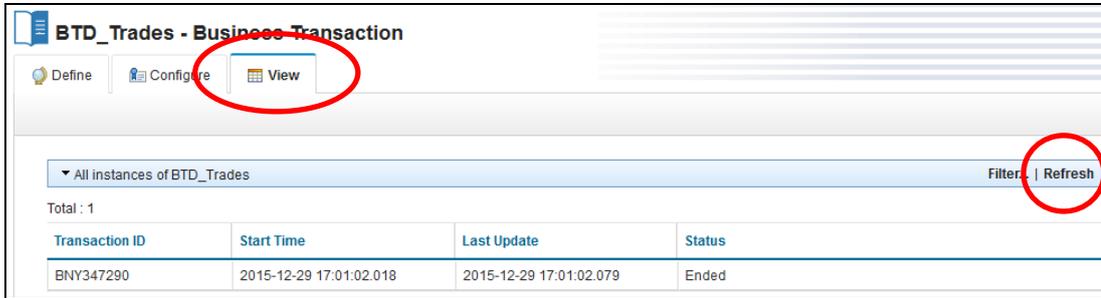
Click the Write Q button to send this message to the first message flow.



- In the IIB web browser, switch back to Business Transaction / BTM\_Trades. On the View tab, click Refresh.

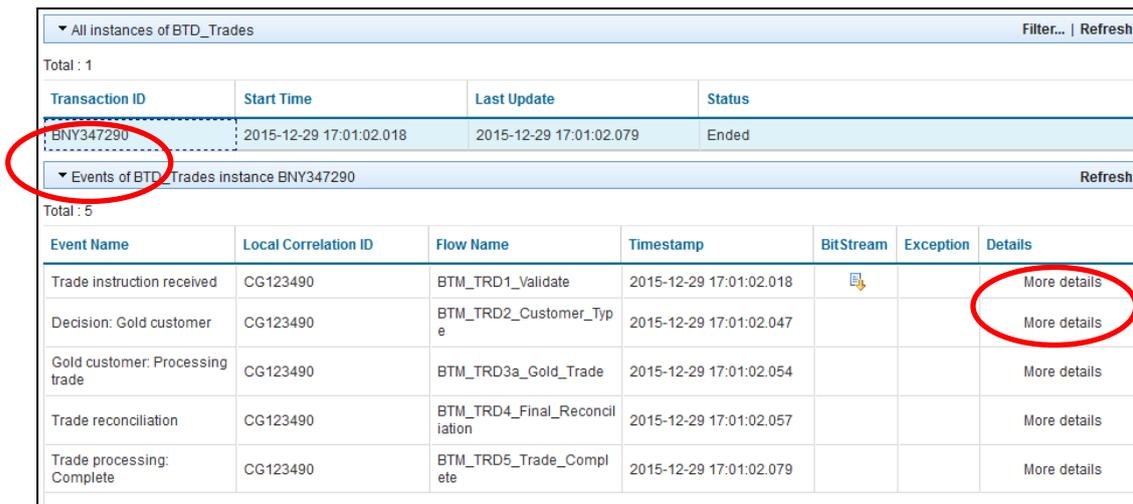
You will see that an instance of the BTM\_Trades business transaction has appeared, denoted by the ID BNY347290.

Note that the Status is **Ended**.

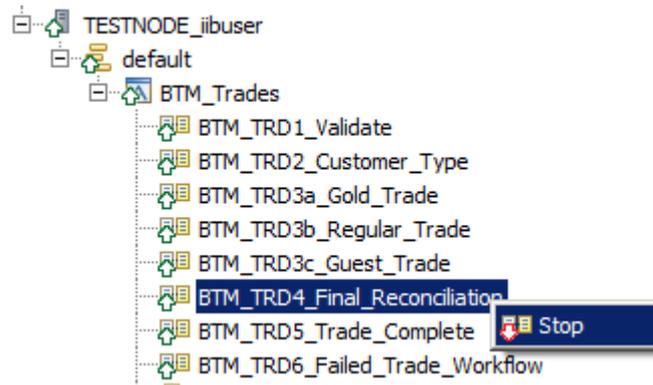


- Click on the line showing the business transaction. The line will highlighted in blue, and all the individual events that comprise the business transaction will be shown.

Observe that the business transaction ID is represented by the Trade ID (the Global Transaction Correlator). For each event, you can click "More details" to see the details of the individual event.



4. Stop the message flow BTM\_TRD4\_Final\_Reconciliation. In the Toolkit, right-click TRD4 and select Stop.



5. Now send several messages through to the message flow.

In a DOS window (or IIB Console), change directory to c:\student10\BTM\_Trades\test.

Run the command :

```
testBTM.cmd
```

This will send a further 10 messages to the application, each of which has a different Trade ID.

```
10 messages to be written to queue TRADE.VALIDATE.IN on queue
manager IB10QMGR
minimum queue depth 3 max 15 batchsize 2
initial sleep time 20 tune = 0
connecting to queue manager IB10QMGR
opening queue TRADE.VALIDATE.IN for output
First message written at Tue Dec 29 17:10:26 2015
MQ Timestamp of first message written at 17:10:26
number on queue after sleep - min 0, max 0
Last message written at Tue Dec 29 17:10:26 2015
MQ timestamp of last message written at 17:10:26

Total messages written 10 out of 10
Total elapsed time in seconds 0.006891
Total bytes written 4330
Total memory used 8180

closing the queue
closing the inquiry queue
disconnecting from the queue manager
MQPUT2 program ended
```

- Back in the Web UI, click Refresh.

You will see a number of instances of BTM\_Trades. They will have a status of "In progress". This means that they have all been started (have executed TRD1), but have not yet reached the final message flow (TRD5) where the end of the business transaction is defined. This is because TRD4 is stopped.

▼ All instances of BTM\_Trades

Total : 11

Transaction ID	Start Time	Last Update	Status
BNY347300	2015-12-30 08:37:47.833	2015-12-30 08:37:47.949	In progress
BNY347299	2015-12-30 08:37:47.832	2015-12-30 08:37:47.912	In progress
BNY347298	2015-12-30 08:37:47.831	2015-12-30 08:37:47.908	In progress
BNY347297	2015-12-30 08:37:47.828	2015-12-30 08:37:47.896	In progress
BNY347296	2015-12-30 08:37:47.827	2015-12-30 08:37:47.894	In progress
BNY347295	2015-12-30 08:37:47.826	2015-12-30 08:37:47.889	In progress
BNY347294	2015-12-30 08:37:47.821	2015-12-30 08:37:47.886	In progress
BNY347293	2015-12-30 08:37:47.813	2015-12-30 08:37:47.873	In progress
BNY347292	2015-12-30 08:37:47.804	2015-12-30 08:37:47.860	In progress
BNY347291	2015-12-30 08:37:47.799	2015-12-30 08:37:47.837	In progress

▼ Events

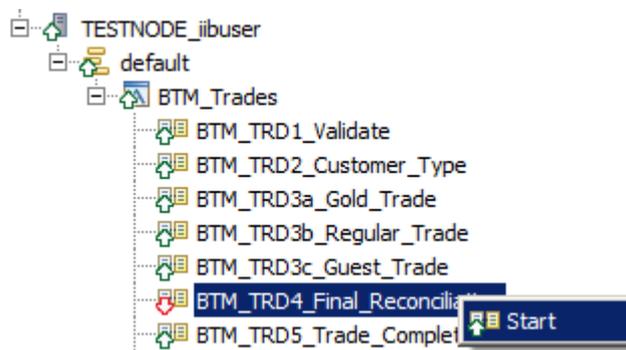
- You can click on any of the transaction instances to see the individual events that have been received for that instance.

▼ Events of BTM\_Trades instance BNY347300 Refresh

Total : 3

Event Name	Local Correlation ID	Flow Name	Timestamp	BitStream	Exception	Details
Trade instruction received	CG123500	BTM_TRD1_Validate	2015-12-30 08:37:47.833			More details
Decision: Gold customer	CG123500	BTM_TRD2_Customer_Type	2015-12-30 08:37:47.944			More details
Gold customer: Processing trade	CG123500	BTM_TRD3a_Gold_Trade	2015-12-30 08:37:47.949			More details

- Now restart the TRD4 message flow.



- Back in the Web UI `BTD_Trades` view, click Refresh again.

You will see the transaction instances have been updated. All should have a status of Ended.

**BTD\_Trades - Business Transaction**

Define | Configure | View

▼ All instances of BTD\_Trades Filter... | Refresh

Total : 11

Transaction ID	Start Time	Last Update	Status
BNY347300	2015-12-29 17:10:27.047	2015-12-29 17:10:27.070	Ended
BNY347299	2015-12-29 17:10:27.031	2015-12-29 17:10:27.066	Ended
BNY347298	2015-12-29 17:10:26.980	2015-12-29 17:10:27.063	Ended
BNY347297	2015-12-29 17:10:26.976	2015-12-29 17:10:27.056	Ended
BNY347296	2015-12-29 17:10:26.920	2015-12-29 17:10:27.049	Ended
BNY347295	2015-12-29 17:10:26.901	2015-12-29 17:10:27.034	Ended
BNY347294	2015-12-29 17:10:26.874	2015-12-29 17:10:27.002	Ended
BNY347293	2015-12-29 17:10:26.857	2015-12-29 17:10:26.976	Ended
BNY347292	2015-12-29 17:10:26.850	2015-12-29 17:10:26.957	Ended
BNY347291	2015-12-29 17:10:26.844	2015-12-29 17:10:26.900	Ended

▼ Events Refresh

- Experiment with the Filter options. For example, set the Filter to Transaction ID contains 296, and click OK.

**Filter**

Specify some filtering criteria

Transaction ID contains

Status equals In progress

OK Cancel

The display will be reduced to the matching transaction.

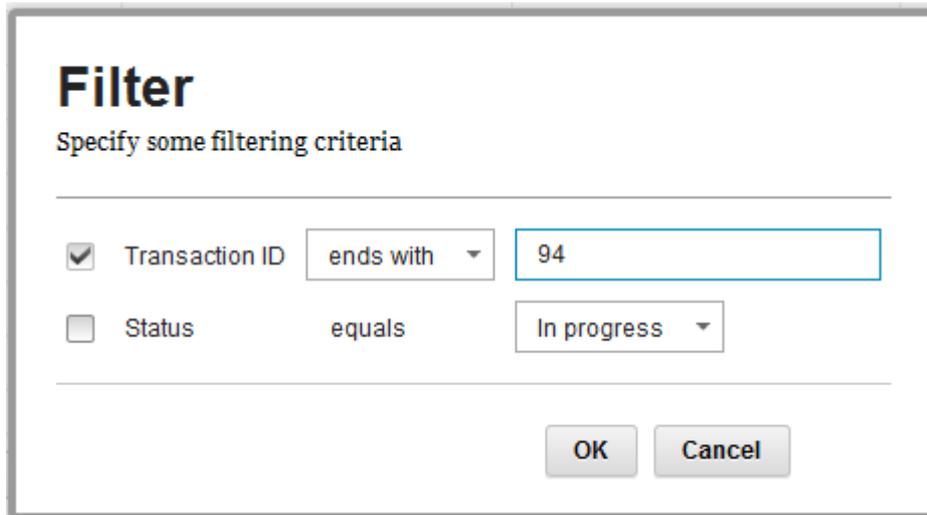
▼ Filtered instances of BTD\_Trades

Total : 1

Transaction ID	Start Time	Last Update	Status
BNY347296	2015-12-29 17:10:26.920	2015-12-29 17:10:27.049	Ended

11. Clear the filter, then set the filter to "Transaction ID ends with 94", click OK.

When finished, clear the Filter.



12. Send a message that is malformed. In RFHUtil, open the file TradeMessage\_Malformed.xml.

This message has a missing closing XML tag, and will fail validation on the input node in the TRD1\_Validate message flow.

In RFHUtil, click Write Q to send this message.

13. In the Web UI, click Refresh. You will see a new transaction has appeared at the top of the list, with the status Failed.

Transaction ID	Start Time	Last Update	Status
BNY590012	2015-12-29 17:19:21.537	2015-12-29 17:19:21.538	Failed
BNY809092	2015-12-29 17:18:52.435	2015-12-29 17:18:52.481	Ended
BNY347300	2015-12-29 17:10:27.047	2015-12-29 17:10:27.070	Ended

- Highlight the business transaction, and you will see the individual events.

You will see that the second event name is "Data validation failure".

Events of BTM_Trades instance BNY590012							Refresh
Total : 2							
Event Name	Local Correlation ID	Flow Name	Timestamp	BitStream	Exception	Details	
Trade instruction received	CG123456	BTM_TRD1_Validate	2015-12-29 17:19:21.537			More details	
Data validation failure	CG123456	BTM_TRD1_Validate	2015-12-29 17:19:21.538			More details	

- Click "More details". You will see the full details of the event. Note that the Correlation IDs are shown correctly. This is because IIB was able to parse the incoming message at least as far as the field that is used for the Global Correlation ID (TradeOrderID). The XML error that caused the validation failure was after this particular element. In cases where IIB is not able to fully parse the message and obtain the value for the Global Correlation ID, it will not be possible to record this data, and a more generic failure will be recorded.

## View Event Details

View the details of the 'Data validation failure' event of 'BTM\_Trades' instance 'BNY590012'

---

Event Name:	Data validation failure
Event Source Address:	Validation Failure.terminal.in
Integration Node Name:	TESTNODE_iibuser
Integration Server Name:	default
Message Flow Name:	BTM_TRD1_Validate
Node Name:	Validation Failure
Node Type:	ComIbmMQOutputNode
Node Terminal Name:	in
Local Correlation ID:	CG123456
Parent Correlation ID:	BNY590012
Global Correlation ID:	BNY590012

---

Cancel

## 6. Add Security Roles for Web Users

So far in this lab, the IIB node has permitted all users to login. This means that users who are only interested in looking at BTM events have also had access to the IIB node, server and applications. You will now introduce administration security, which will provide more granular control of access to these functions.

This is discussed in more detail in the Admin Security lab.

1. First, activate administration security for the IIB node. In an IIB Console, execute the following commands:

```
mqsistop TESTNODE_iibuser  
  
mqsichangeauthmode TESTNODE_iibuser -s active -m file  
  
mqsistart TESTNODE_iibuser
```

2. Define role permissions for various classes of users. In the IIB Console, run the following commands:

```
cd c:\student10\BTM_Trades\webadmin  
  
set_BTM_users_ACL_using_FileAuth.cmd
```

Accept the defaults in the cmd file.

The cmd file will issue a number of mqsi commands similar to these shown here:

```
mqsichangefileauth IIBNODE -r iibAdmin2 -p read+  
mqsichangefileauth IIBNODE -r iibAdmin2 -e default  
-p read+,write+,execute+  
mqsichangefileauth IIBNODE -r iibAdmin2 -o DataCapture -p read-
```

For example, these commands define a profile (role) called iibAdmin2 which permits full read/write/execute access to the IIB node, full access to the server (default), but no access to the BTM functions (the DataCapture object).

3. Define the Web UI users who will use these permissions.

From the same directory, run the following command:

```
Create_BTM_users.cmd
```

This cmd file will run a number of mqsi commands similar to this:

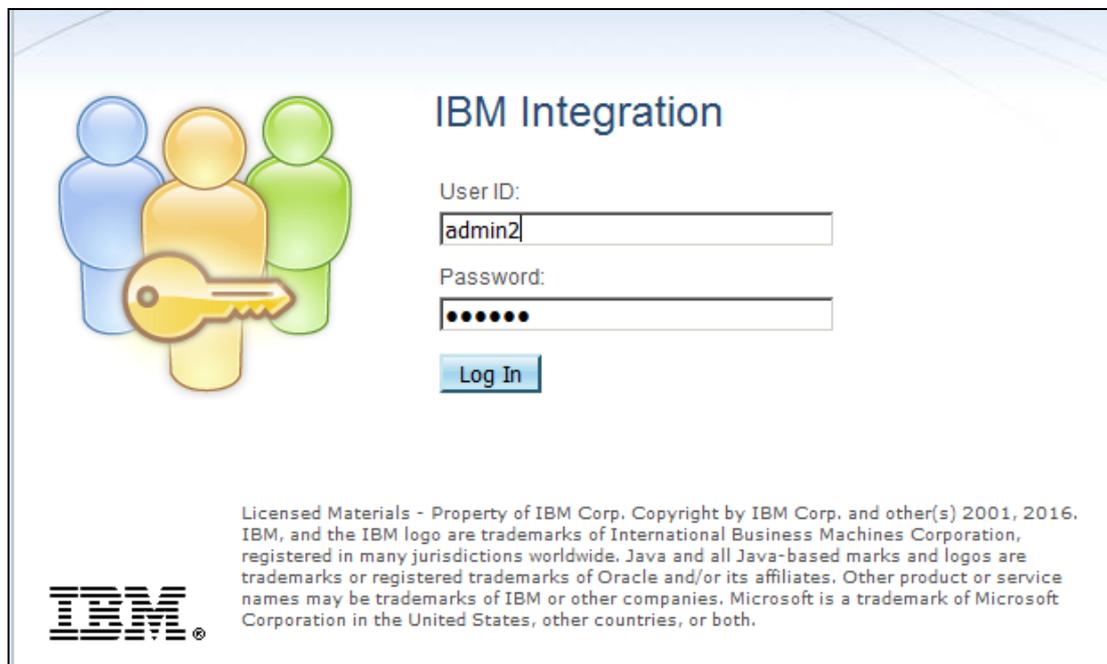
```
mqswebuseradmin IIBNODE -c -u admin2 -a passw0rd -r iibAdmin2
```

This example command will define a new user, admin2, with the associated role of iibAdmin2.

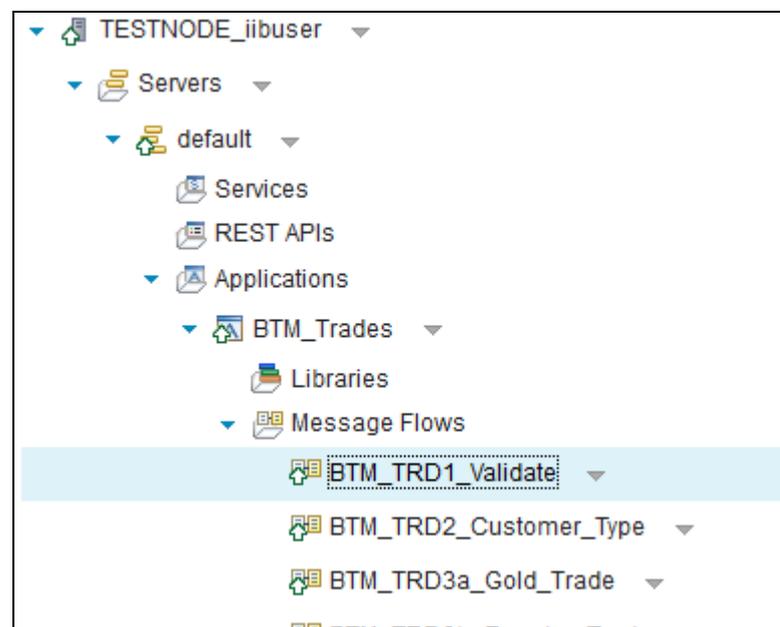
4. In the Web browser, first make sure you are fully logged out of the node.

Login with the user **admin2**, password = admin2.

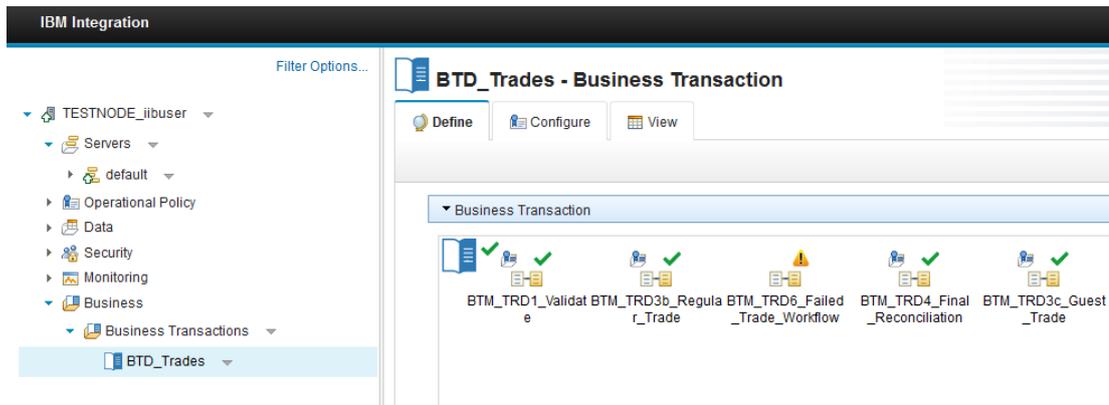
This user has full (update) access to the IIB node, server and applications, but no access the any BTM events.



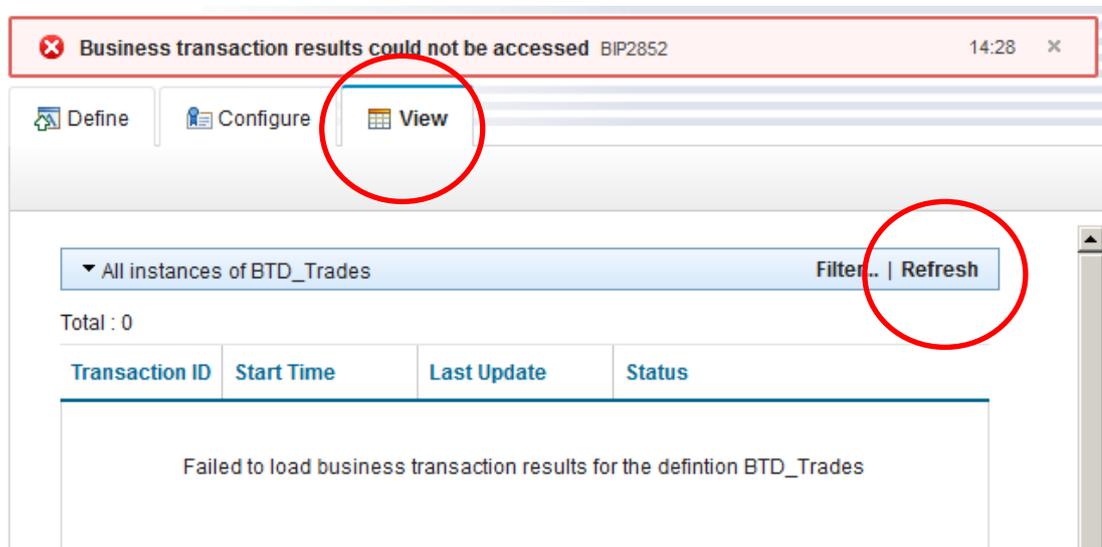
5. In the navigator, you can expand all items, and have full access to control resources.



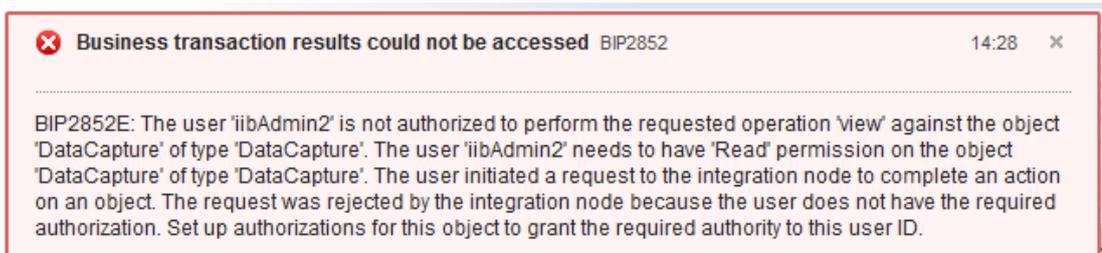
- Expand the Business folder, and select the `BTD_Trades` definition. You will see that you are able to see the definition of the `BTD`, and indeed make changes to the `BTD`. This is because `admin2 (iibAdmin2)` has update access to the `IIB` node and the server, which controls the definition and saving of `BTDs`.



- Click the `View` tab, and then click `Refresh`. You will see the message "Business transaction results could not be accessed".

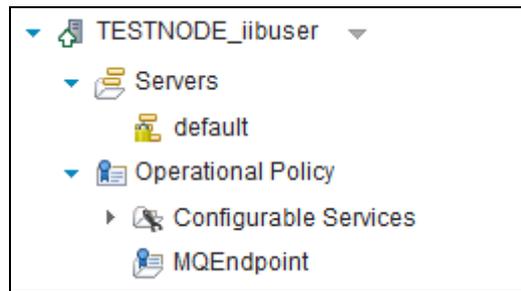


- Clicking on the message will provide more details. Insufficient authority is available for users with role "iibAdmin2". `iibAdmin2` does not have "read access" to the `DataCapture` object.



- Logout admin2, and login with **btm1** (password = btm1).

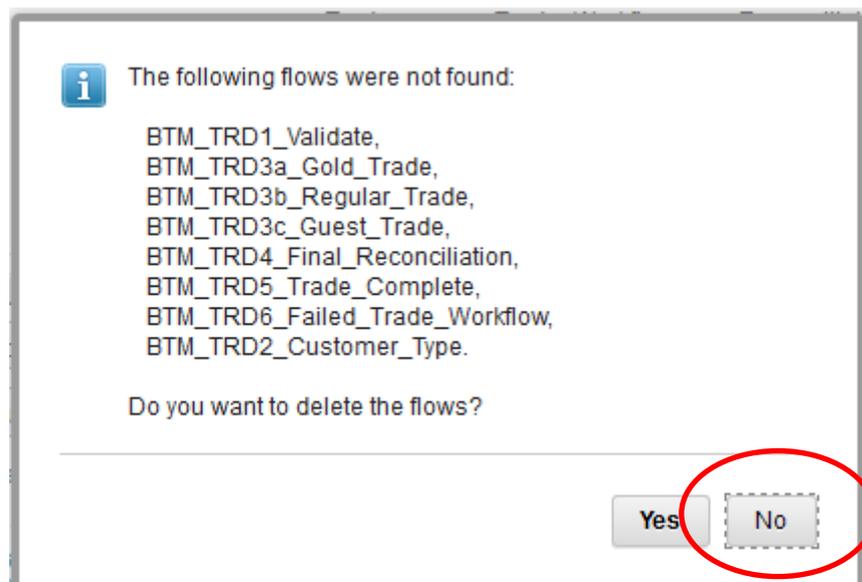
In the navigator, this user cannot see any IIB resources.



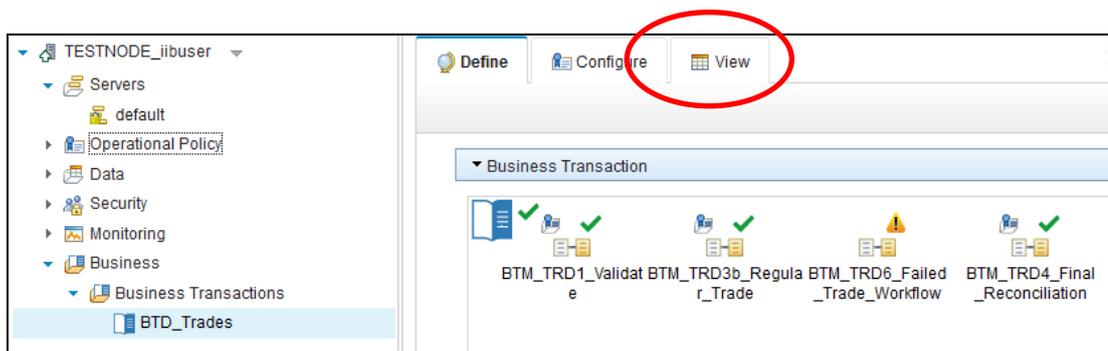
- Expand the Business folder and select **BTD\_Trades**.

This user will not be able to create a new BTD, or make changes to the existing BTD.

Note, you may see a message indicating that the flows that comprise the BTD are not available. Answer No when asked whether to delete these flows. This message appears because this user does not have access to view deployed resources.

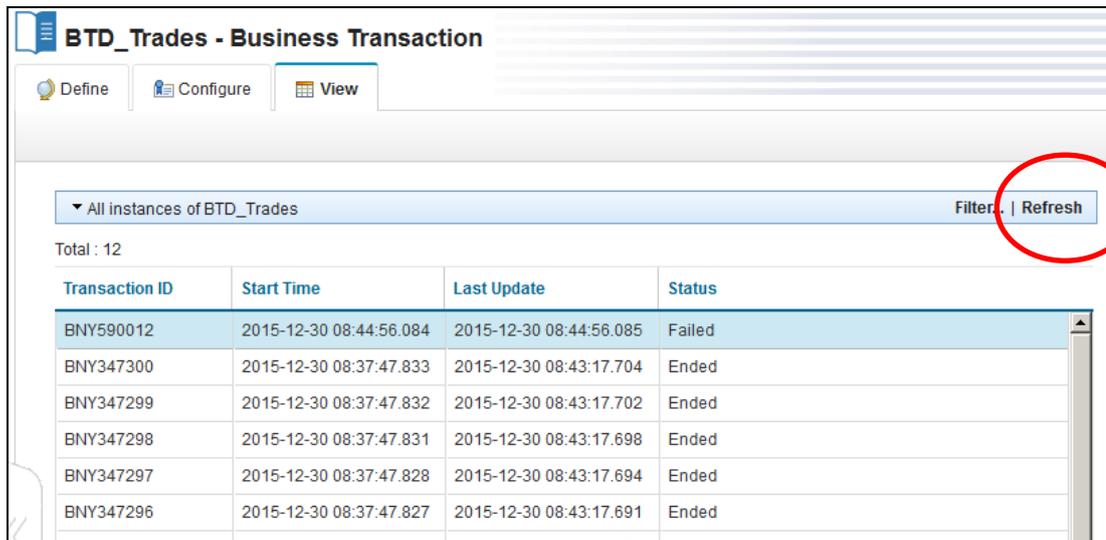


- Click the **View** tab.



- Click the Refresh button. You will see all the transaction instances that were created earlier.

You will be able to perform all the actions that were described earlier, such as filtering, and viewing the individual events.



**BTD\_Trades - Business Transaction**

Define | Configure | View

▼ All instances of BTD\_Trades Filter... Refresh

Total : 12

Transaction ID	Start Time	Last Update	Status
BNY590012	2015-12-30 08:44:56.084	2015-12-30 08:44:56.085	Failed
BNY347300	2015-12-30 08:37:47.833	2015-12-30 08:43:17.704	Ended
BNY347299	2015-12-30 08:37:47.832	2015-12-30 08:43:17.702	Ended
BNY347298	2015-12-30 08:37:47.831	2015-12-30 08:43:17.698	Ended
BNY347297	2015-12-30 08:37:47.828	2015-12-30 08:43:17.694	Ended
BNY347296	2015-12-30 08:37:47.827	2015-12-30 08:43:17.691	Ended

- To ensure other labs run successfully, please deactivate administration security:

```
mqsistop TESTNODE_iibuser
```

```
mqsichangeauthmode TESTNODE_iibuser -s inactive
```

```
mqsistart TESTNODE_iibuser
```

## END OF LAB GUIDE