

# Business Transaction Monitoring in IIB

MattClarke

Published on December 4, 2015 / Updated on March 22, 2016

With the availability of IBM Integration Bus v10 Fixpack 3, we introduced the concepts on Business Transaction Monitoring (BTM) in [Business Transaction Monitoring – Why, what, how.](#)

In this blog post we will detail an end-to-end example scenario involving BTM from definition to viewing of transaction results.

We will create a new Business Transaction Definition (BTD), perform the configuration steps to leverage recording of BTM events in preparation for sending messages that will be monitored by our definition. Then view the results of the recorded business transactions.



What is required

An IIB node associated with a queue manager. (The monitoring feature in IIB requires WebSphere MQ to function)

DDL scripts to generate required table

The IIB resource BTM\_Purchase.bar that contains the application that will be used

create\_queues.mqsc script

Sample messages BTM\_NewPurchase.xml and BTM\_InvalidPurchase.xml files

IBM DB2 (or other supported Database – Oracle)

[IIB Getting Started with BTM Artefacts .zip \(attached below\)](#)

Scenario

In our application there are 3 message flows that form a path for purchase order messages to be processed across.

BTM1\_Validate\_Purchase message flow performs initial validation checks on the message.

BTM2\_Dispatch\_Purchase message flow receives a valid message and sends it out for dispatch.

BTM3\_Purchase\_Complete message flow completes the purchase order.



BTM\_Purchase application overview

The purpose of these message flows is to facilitate the ordering of purchases made by a customer by validating the purchase message, and then forwarding onto dispatch and purchase.

The end to end execution of this can be considered to be one single transaction, and as a business level entity (making a purchase) it is known as a business transaction.

Suppose you would like to know about when such transactions occur, how they are handled, how many there are, when they occurred and so on?

This is where the BTM feature offers a representation of a defined business process, in a multi-flow, event-driven environment that can be defined and viewed as a set of transaction results.

MQ Setup

The messaging technology used for this scenario is MQ, and several queues need to be created in order to facilitate the sending and receiving of messages between flows.

MQ is also required for using BTM, so the integration node must have an associated queue manager using the `-q` parameter when the node was created, or when running the `mqsic changebroker` command. MQ is used for publishing the monitoring events to MQ topics. BTM sets up subscriptions to these MQ topics to process and correlate the business events which contribute to a particular business transaction.

Several queues need to be created under the queue manager. Running the script against the relevant queue manager `create_queues.mqsc` will create the correct queues. The queues that are created are bespoke for the application, but there are 2 further queues needed for BTM to function.

When a monitoring event is emitted it is published to a MQ Topic on the associated Queue Manager. These subscriptions purpose is to handle event emission and write the event to a MQ queue, the `SYSTEM.BROKER.DC.RECORD` queue, which queues up event messages to be processed and written to a database. A further queue `SYSTEM.BROKER.DC.BACKOUT` is also needed to handle events that cannot be at process time be written to the database, and thus are backed out onto this queue. These queues need to be created prior to using BTM, and are generated by running the script located at `{install path}/sample/wmq/iib_createqueues.cmd`.

### Database Setup

To facilitate the recording of transactions, a supported database must be configured before using BTM. For BTM this tutorial uses IBM DB2.

If Record & Replay is currently used, then only a single database table needs creating for BTM. If it isn't then a new database and tables need to be created first. The IIB installation includes an SQL script, located at `{install path}/ddl/db2/DataCaptureSchema.sql` to generate the necessary database tables. You will need to customize the script to suit your recording needs and database naming standards. At the top of the file, you will find the database name `MBRECORD`, and you can optionally use a database schema.

Run this script using the command:

```
db2 -tvf DataCaptureSchema.sql
```

from a db2 command shell. Check that the statements executed correctly before continuing.

To define the BTM table, the IIB installation includes an SQL script, located at `{install path}/ddl/db2/BusinessCaptureSchema.sql`.

Run the script using the command:

```
db2 -tvf BusinessCaptureSchema.sql
```

The next task is to create an ODBC datasource for this database so that the integration node can connect to it. This task is platform-dependent. For more information, see [Enabling ODBC connections to databases](#) in the IBM Integration Bus v10 Knowledge Center.

In this example, we use the name `MBRECORD` for the ODBC definition to match the database name.

The database should now be configured and ready to use, and you can now connect the integration node to the database. For example you would use the following `mqsisetdbparms` command to store the database credentials:

```
mqsisetdbparms IB10NODE -n odbc::MBRECORD -u {DBUSER} -p {DBPASSWORD}
```

After this command completes successfully, restart the integration node to pick up the new credentials. Then use the `mqsicvp` command to verify that the database connection is correct:

```
mqsicvp IB10NODE -n MBRECORD
```

If successful, this command returns a list of supported operations against the database.

### Working with the BTM\_Purchase Application

BTM uses the existing message flow monitoring capability to record and correlate events across a business transaction. To enable this for BTM, there is a requirement to specify such a correlator throughout the intended flow of a transaction to be monitored. For BTM this is the global transaction ID. This must be specified when defining monitoring events within a message flow to be used for recording business transactions. You must define the value of a global transaction ID in each monitoring event. The global transaction ID is used to correlate the individual events which contribute to the business transaction. The global transaction ID is typically some part of the message that is being processed, for example, a customer number or an order ID.

**Default Values for Message Flow Properties - BTM1\_Validate\_Purchase**

Description: Configure monitoring events.

**Monitoring**

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
<input checked="" type="checkbox"/>	Purchase order rec...	Transaction start	Purchase order received.trans
<input checked="" type="checkbox"/>	Validation Failure	In terminal	Validation Failure.terminal.in

Overview of BTM\_Purchase in IBM Integration Toolkit

The application's message flows already have the required monitoring events defined with a global transaction correlator value set. In this scenario we use the purchase order ID in the message.

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.

OK

**Input Node Properties - Purchase order received**

Configure monitoring events.

Events:

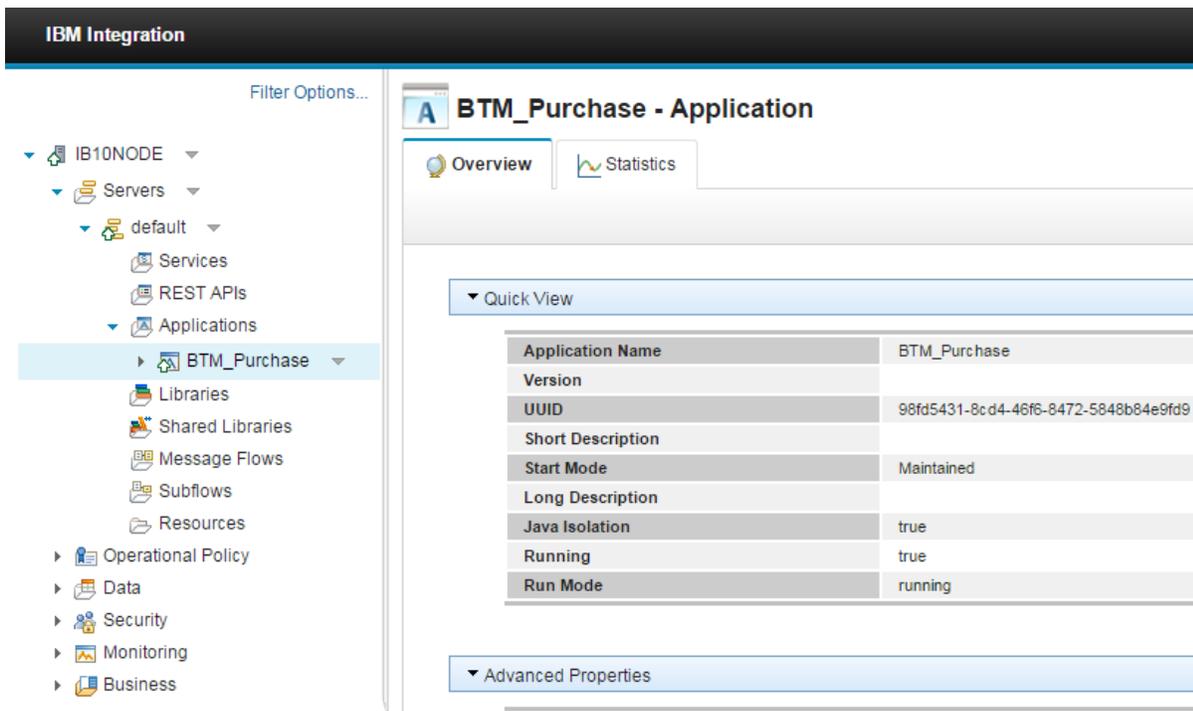
Enabled	Event Source	Event Source Address	Event Name	Event
<input checked="" type="checkbox"/>	Transaction start	Purchase order received.transaction.Start	Purchase order received	tru

Monitoring event global transaction correlator value

All other monitoring events defined on the application's message flows also use the same global transaction correlator value. This means that in order to be able to distinguish between recorded business transactions, a single message's purchase order ID must be unique from other's that have previously been processed.

### Creating the Business Transaction Definition

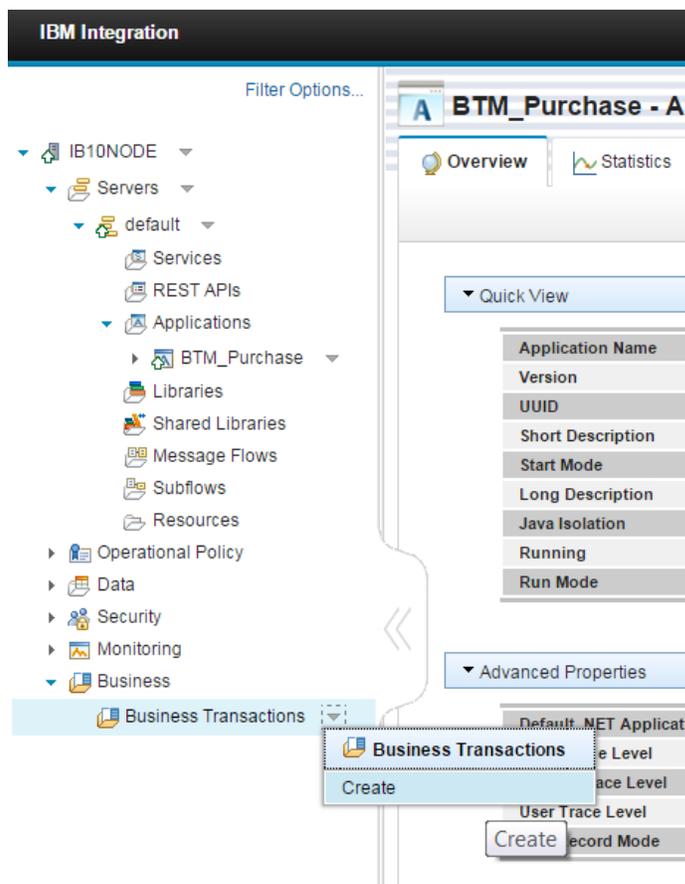
After deploying the BTM\_Purchase application to your integration node, all remaining configuration can be performed using the IBM Integration Bus WebUI for your Integration Node. Once deployed you may typically have a WebUI tree populated in the following manner:



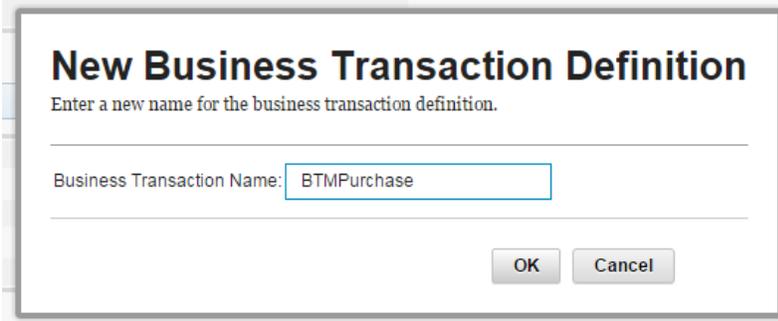
IIB WebUI tree with BTM\_Purchase application deployed

Notice that for IBM Integration v10 Bus Fix pack 3 there is now a new category in the tree named “Business”. This is where all the BTM capability is accessed in the WebUI.

BTM uses the concept of a Business Transaction Definition (BTD) that contains the configuration for the transaction. In order to be able to start monitoring transactions a new definition must be created. Go to the business section of the tree and expand the Business node. Then to create a new definition click on the down-arrow to bring up the business transactions options, and select Create.



When prompted, specify a name for the new BTM and confirm by clicking OK.



The dialog box is titled "New Business Transaction Definition" and contains the instruction "Enter a new name for the business transaction definition." Below this is a text input field labeled "Business Transaction Name:" with the value "BTMPurchase" entered. At the bottom right, there are two buttons: "OK" and "Cancel".

Specify new BTM name

This creates a new BTM instance on your integration node, ready to be defined. There are 3 tabular sections for a BTM:

Define – Where the BTM structure is defined and visualised.

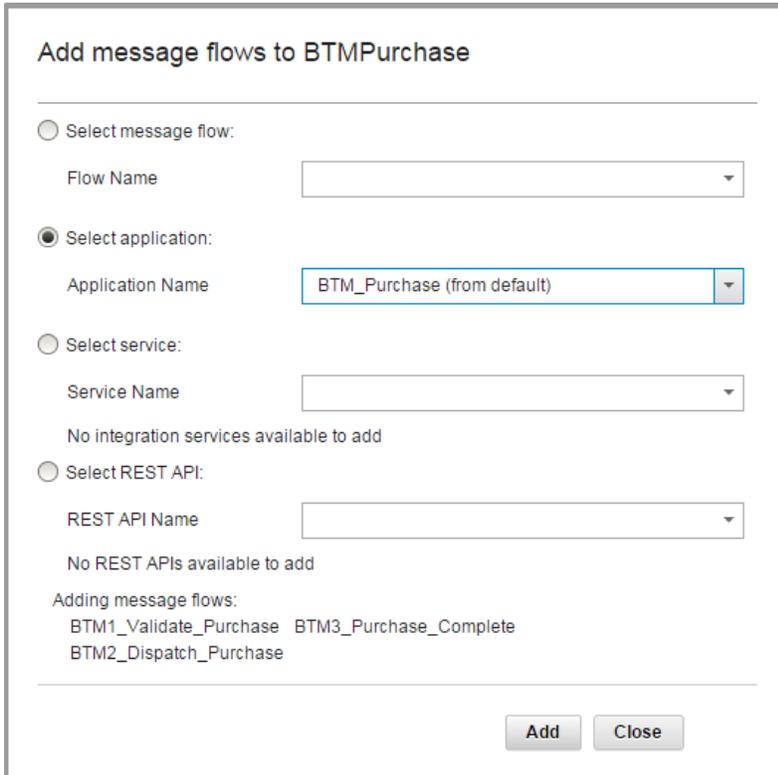
Configure – Where configuration information is given and potentially guided tasks to resolve any problems.

View – Where results can be viewed for recorded business transactions for this definition.

#### Define the definition

A BTM is able to work with artefacts that are currently deployed to the same integration node. In this instance we want to work with our recently deployed BTM\_Purchase application and the message flows that it contains.

You'll notice that there is a blank canvas in the top section. The canvas is a visual tool to illustrate the currently included message flows in the definition, and provides additional information related to the definition that is relevant to BTM. To populate the definition with message flows the mechanism provided is to use the Add Flow button. This will display the Add flow dialog. Here all deployed artefacts are placed into their respective categories as a series of selectable drop-down lists.



The dialog box is titled "Add message flows to BTMPurchase". It has four radio button options: "Select message flow:", "Select application:", "Select service:", and "Select REST API:". The "Select application:" option is selected. Under "Select application:", the "Application Name" dropdown is set to "BTM\_Purchase (from default)". Under "Select service:", the "Service Name" dropdown is empty, with the text "No integration services available to add" below it. Under "Select REST API:", the "REST API Name" dropdown is empty, with the text "No REST APIs available to add" below it. At the bottom, there is a section "Adding message flows:" with a list of three items: "BTM1\_Validate\_Purchase", "BTM3\_Purchase\_Complete", and "BTM2\_Dispatch\_Purchase". At the bottom right, there are two buttons: "Add" and "Close".

Add BTM\_Purchase application message flows to BTM

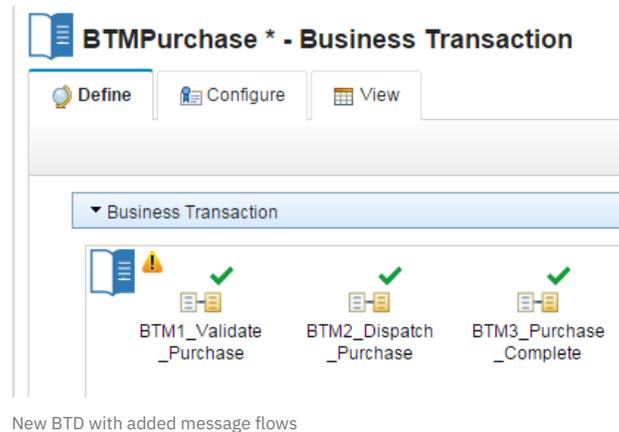
To add a single message flow at a time, the message flow list has all available message flows deployed on the integration node that are not currently included in the definition.

To add all message flows within a single application, the application list has all available applications deployed on the integration node. Any message flows currently included in the definition that are a part of the selected application will not be added.

To add a single Integration Service object, the services list has all available Integration services deployed on the integration node.

To add a single REST API object, the REST APIs list has all available REST APIs deployed on the integration node.

As we want to include monitor events that span across multiple message flows in our business transaction, then it is easier to add all message flows in the BTM\_Purchase application. When an application is selected, a summary is presented of the message flows that will be added upon confirmation. Clicking Add will add the message flows to the definition and each flow is rendered as a message flow artefact on the canvas.

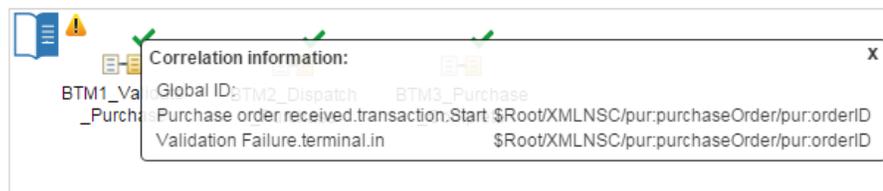


An \* (asterisk) next to the BTD name now denotes that modifications have taken place since the last save of the definition. This is the mechanism used throughout the define task to denote that there are currently unsaved changes against the definition.

If a message flow is not required to be a part of the BTD, then they can be removed using the remove flow button to bring up a similar dialog as is used when adding message flows. A message flow can also be deleted by selecting it on the canvas and pressing the delete key.

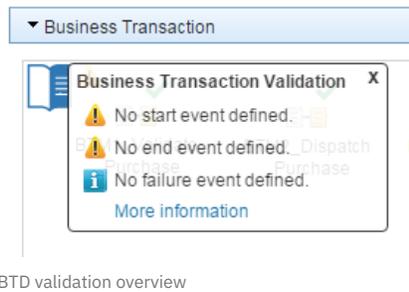
#### Available Monitoring Information

On the define tab there are several methods for inspecting and discovering information about the message flow that you have just added. Validation – Clicking on the green tick brings up information on the validation assessment of the message flow with respect to BTM and the definition. In this instance everything is resolved to ok for the message flow because it contains some monitoring events on its nodes. We can also see the global transaction ID correlator information for all monitoring events that are defined on message flow nodes for the message flow.



Monitoring event correlation information for a message flow

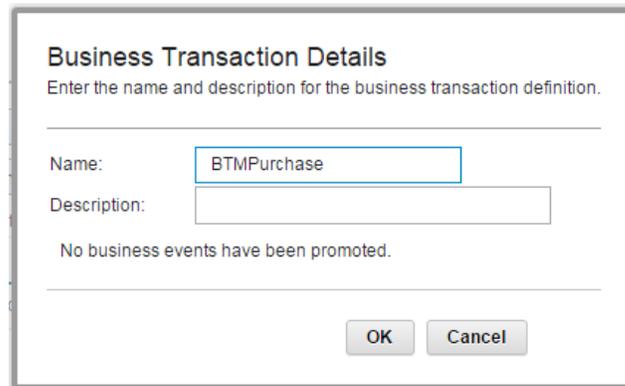
If the message flow is not valid a warning icon will be shown, and clicking on that will give further information as to what is not currently valid. Such a warning icon can be seen next to the BTD icon on the canvas. This object is a representation of the current BTD entity. Clicking on the warning icon will give details as to why the BTD might not currently be defined correctly:



BTD validation overview

No business transaction events have yet to be defined, so the BTD is not complete and not ready to start recording business transactions.

You can also inspect a summary of the BTD by clicking on the BTD icon itself. A new dialog will appear and show information on the BTD:



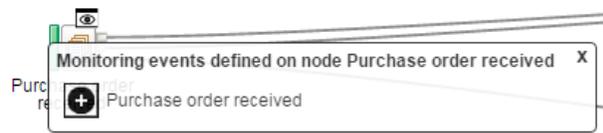
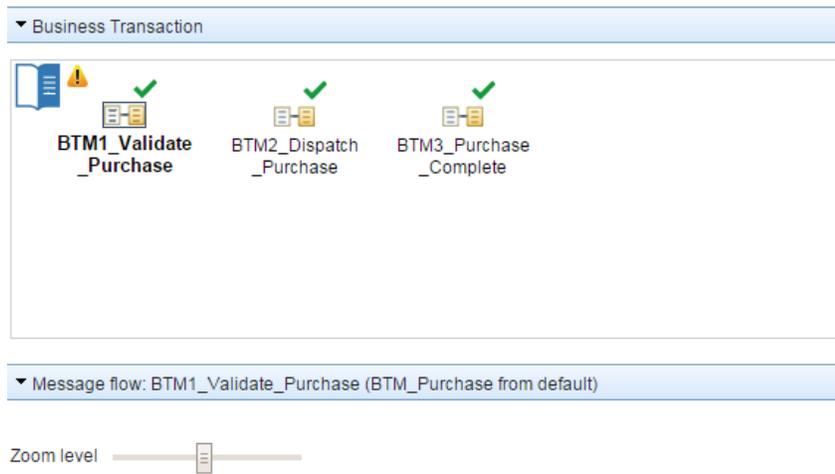
BTD definition overview

As is confirmed by the validation warning, no business events have been defined yet.

### Flagging Business Events

Currently there are added message flows that have valid monitoring events associated with them but by default they are not included in the business transaction. The definition needs to know which events it needs to include in its monitoring of a transaction. This is achieved through the “Flagging” of monitoring events to become business events.

To assist in making such a decision, the middle section provides a flow profile view of a message flow when it is selected on the canvas. Inspection of event information (denoted by the event icon) on individual nodes provides a visual message flow perspective of what monitoring events are available to be flagged to business events.



Inspection of monitoring events on flow profile node

The bottom section on the define tab provides a grid view of the current definition. When no message flow is selected it provides a tree grid of all included flows, and their respective monitor events details. When a message flow is selected on the canvas, the scope of the grid narrows to that message flow monitoring event information.

Business Transaction Event Definitions		
This section shows the monitoring events that are defined for the selected flow or for all flows in the business transaction. Select the monitoring events that signify the start, en		
Message Flow - Monitor Event	Event Source Address	Flag as
[-] BTM1_Validate_Purchase (BTM_Purchase from default)	2 Monitoring Events	
Purchase order received	Purchase order received.transaction.Start	Do not flag ▾
Data validation failure	Validation Failure.terminal.in	Do not flag ▾
[+] BTM2_Dispatch_Purchase (BTM_Purchase from default)	1 Monitoring Event	
[-] BTM3_Purchase_Complete (BTM_Purchase from default)	2 Monitoring Events	
Purchase complete: Notify customer	Completion.transaction.Start	Do not flag ▾
Purchase Complete.InTerminal	Purchase Complete.terminal.in	Do not flag ▾

BTD Message Flow Event Grid

It is within this grid that the flagging of monitoring events to business events is performed. Changing the value of the drop-down in the Flag as column will change the role of that monitoring event with respect to the current BTM. By default every monitoring event is set to Do not flag, so it will not participate in the business transaction. Any other value will mean it has some meaning to the business transaction and will be recorded when an event is emitted.

For example changing the Purchase order received monitor event in BTM1\_Validate\_Purchase message flow to have a Flag as value of Start will cause a new business transaction to be recorded each time an event is emitted at this location. The inclusion of a business event within a message flow now decorates the flow object with a business event icon on the canvas that when clicked details the current business event(s) for that message flow.



▶ No flow selected

▼ Business Transaction Event Definitions

this section shows the monitoring events that are defined for the selected flow or for all flows in the business transaction. Select the monitoring events 1

Message Flow - Monitor Event	Event Source Address	Flag as
BTM1_Validate_Purchase (BTM_Purchase from default)	2 Monitoring Events	
Purchase order received	Purchase order received.transaction.Start	Start
Data validation failure	Validation Failure.terminal.in	Do not flag

Monitoring event flagged as Business event for BTD

To end the transaction we want to ensure that the purchase has complete. This means setting a business event on a monitoring event in a different message flow, the Purchase Complete.InTerminal event in the BTM3\_Purchase\_Complete message flow.

We now have the minimum required number of business events set to be able to record a business transaction, a start and an end.

For this scenario we want to also know when a received message has failed validation, and complete the business transaction as failed. Setting the monitoring event on that validation failure node to be a Failure business event will ensure that a business transaction can be distinguished between failed and completed.

A value of Progress can also be set on a monitoring event as the business event type. These events are recorded as part of the business transaction, but don't pertain a particular business meaning. They may still be important for example in order to track the path of a message through the message flow, or they have some valuable message payload associated at this point that might want to be recorded. After setting a further monitoring event to Progress the final definition has the following structure:

Message Flow - Monitor Event	Event Source Address	Flag as
BTM1_Validate_Purchase (BTM_Purchase from default)	2 Monitoring Events	
Purchase order received	Purchase order received.transaction.Start	Start
Data validation failure	Validation Failure.terminal.in	Failure
BTM2_Dispatch_Purchase (BTM_Purchase from default)	1 Monitoring Event	
Dispatch order	Dispatch_Purchase.transaction.Start	Progress
BTM3_Purchase_Complete (BTM_Purchase from default)	2 Monitoring Events	
Purchase complete: Notify customer	Completion.transaction.Start	Do not flag
Purchase Complete.InTerminal	Purchase Complete.terminal.in	End

Final BTD flagged events

## Business Transaction Details

Here are the business transaction details.

Name:

Description:

### Business Events:

Type	Flow Name	Event Source
Start	BTM1_Validate_Purchase	Purchase order received.transaction.Start
End	BTM3_Purchase_Complete	Purchase Complete.terminal.in
Failure	BTM1_Validate_Purchase	Validation Failure.terminal.in
Progress	BTM2_Dispatch_Purchase	Dispatch_Purchase.transaction.Start

Final BTM overview

Upon the completing of any changes to a BTM these must be saved by clicking the save button in the toolbar in order to be persisted. Any changes that do not want to be changed can be undone by clicking the cancel button and BTM will revert to the last saved state.

### Completing BTM Configuration

Now that the definition of the BTM is complete, there is one or potentially two more setup steps that need to be complete in order to prepare the integration node for the recording of business transactions. This is performed on the Configure tab for the BTM.

The first is that the data source to which business transactions will be recorded to needs to be defined. The default value for first time use is set to empty (or Not set), so this needs to be set for BTM to function. Note that the data source name that is chosen will apply for all BTMs.

**BTMPurchase - Business Transaction**

Define | **Configure** | View

▼ Data source configuration

Data source name: Not set

[Click here to specify a data source name](#)

Default configuration data source name

There is a click-able link “Click here to specify a data source name” that takes you to the configuration page for BTM. This is also the Business landing page in the tree location. We will set this to the data source name we specified in the `mqsisetdbparms` command, or MBRECORD as this should be available from the drop-down list.

**Configuration**

▼ Data source configuration

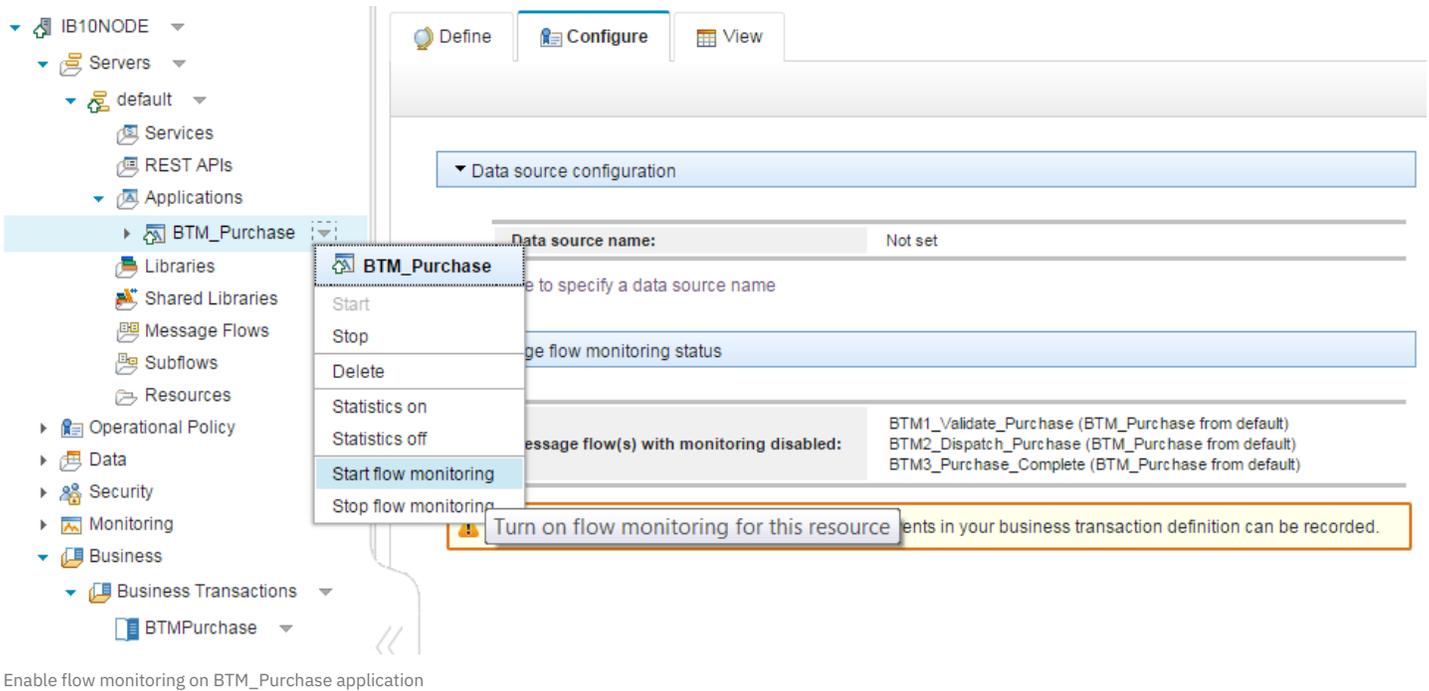
Data source name: MBRECORD

[Resume editing of business transaction BTMPurchase](#)

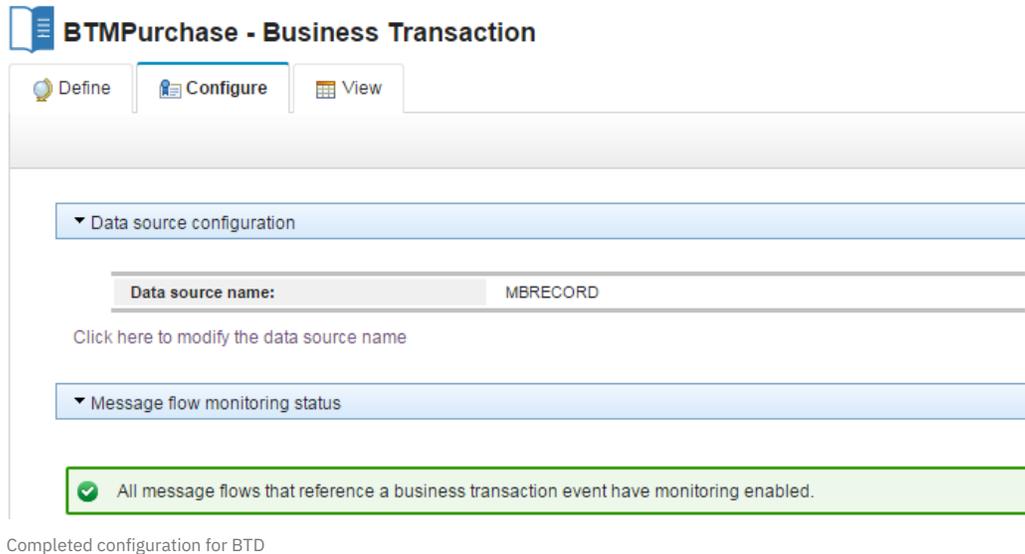
Set the data source name for BTM configuration

Once set, clicking save will store this value, and begin using the associated database for BTM. To resume editing of the BTM that we came from, there is a click-able link that will take a user back to the configure tab for that BTM.

There is just one more configuration step that needs to be performed if not done so already. That is to turn on flow monitoring for those flows that will be part of the business transaction. Under the flow monitoring status section, the current state of the BTM message flows have been assessed and any warnings shown. Turning flow monitoring on and off can be a new feature available in IBM Integration Bus v10 Fix Pack 3. Either at the message flow level, or the parent container, selecting the options panel on that node in the tree will present an option to change the flow monitoring state. So turning it on for the BTM\_Purchase application will mean all message flows in the BTM now have monitoring enabled:



Once turned on subsequent visits (switching back and forth between tabs will perform a reassessment of message flow monitoring status) to the configure tab for the BTM will now show a good message for flow monitoring.



Now everything is set to start recording transactions. To test that we can record, and then view we must send a test message. Placing a message on the earlier created BTM.VALIDATE.IN queue will then begin processing of the message across the message flows until it reaches the final out queue. Whilst being processed specified monitoring events should emit and be recorded in the database, for then later viewing as a single correlated business transaction.

Using rfHUtil (or other queue writing method) place the BTM\_NewPurchase.xml message on the BTM.VALIDATE.IN queue. If you want to

send further test messages, you can use the same test message file, but ensure the orderId value is changed from Purchase001 to another unique purchase order ID. We also want to test the ability to record failed business transactions, to do this we need to emit the monitoring event defined to be failed business event. The BTM\_InvalidPurchase.xml is not a fully valid message, and should cause a transaction to be failed.

### Viewing BTM Results

Now that some transactions should have occurred, we can view the results for that BTM by going to the View tab. There are two grid showing results, the top is the view for business transaction results. The bottom is event results for a single business transaction when one is selected from the top grid.

The querying of business transaction results requires either requesting all results, or a subset based on filtering upon some of the columns. As prompted, clicking the refresh button will query the database for viewing results:

The screenshot shows the 'BTMPurchase - Business Transaction' interface. It has three tabs: 'Define', 'Configure', and 'View'. Below the tabs is a dropdown menu set to 'All instances of BTMPurchase'. Below that, it says 'Total : 3'. A table displays the following data:

Transaction ID	Start Time	Last Update	Status
Purchase999	2015-11-23 16:25:53.929	2015-11-23 16:25:53.929	Failed
Purchase002	2015-11-23 16:25:14.758	2015-11-23 16:25:14.761	Ended
Purchase001	2015-11-23 16:24:26.566	2015-11-23 16:24:26.570	Ended

Results grid for BTMPurchase

To apply filtering conditions on results, use the Filter button to launch the filter dialog and set filtering parameters to be applied to the BTM results query.

The screenshot shows a 'Filter' dialog box with the title 'Filter' and the subtitle 'Specify some filtering criteria'. It contains two filter criteria:

- Transaction ID equals [ ]
- Status equals [Failed] (dropdown menu is open showing options: In progress, Ended, Failed, Inconsistent)

A 'Cancel' button is visible on the right side of the dialog.

Filter dialog for BTM results

Clicking on a single business transaction result in the grid will populate the events grid with the events that were emitted and contribute to that business transaction instance.

▼ All instances of BTMPurchase

Total : 3

Transaction ID	Start Time	Last Update	Status
Purchase999	2015-11-23 16:25:53.929	2015-11-23 16:25:53.929	Failed
Purchase002	2015-11-23 16:25:14.758	2015-11-23 16:25:14.761	Ended
Purchase001	2015-11-23 16:24:26.566	2015-11-23 16:24:26.570	Ended

▼ Events of BTMPurchase instance Purchase001

Total : 3

Event Name	Local Correlation ID	Flow Name	Timestamp
Purchase order received	03b84221-ed76-4f7c-aa08-261ef6dfcf4b-1	BTM1_Validate_Purchase	2015-11-23 16:24:26.566
Dispatch order	8a4ea1d0-31b8-43f6-bb75-433d8211a8cc-1	BTM2_Dispatch_Purchase	2015-11-23 16:24:26.568
Purchase Complete.InTerminal	808985a3-0f9d-44f3-bf29-8e2c0c77e13c-1	BTM3_Purchase_Complete	2015-11-23 16:24:26.570

Event results for a single business transaction

Where applicable, bitstream and exception data can be downloaded and viewed for a single monitoring event that was emitted.

### Summary

In this tutorial, we have taken you through the new Business Transaction Monitoring capability that has been added in IBM Integration Bus v10.0.0.3. Using this capability, you are able to track the life cycle of your business events across your lines of business. You can view which business transactions end successfully or which have failed. You can track specific business transactions using a global transaction identifier which can help you find what happened to a lost order or why a particular business transaction has taken longer to complete than normal.

Further to this tutorial there is also an informative video of the using BTM at [Business Transaction Monitoring Demo](#)

---

TAGS BUSINESS-TRANSACTION-MONITORING, BTM, IIBV10.0.0.3, FESTIVE2015

---

MattClarke

---

## 23 comments on "Business Transaction Monitoring in IIB"

Hitesh Agrawal October 11, 2019

BTM , can we monitor the number of message if it is not flowing to the application by setting some threshold

[Reply \(Edit\)](#)

MattClarke October 14, 2019

Hi Hitesh,

Can I ask for more details on the task you are trying to achieve?

I don't fully understand where you ask "can we monitor the number of message if it is not flowing to the application by setting some threshold".

Regards,

Matthew

[Reply \(Edit\)](#)

YvettePalmer October 12, 2018

You talk about MQ queues have to be created from your script that is provided but you dont talk about the permissions needed for the queues. II would think read and write? Is that correct? .

```
define qlocal(BTM.VALIDATE.IN) replace
define qlocal(BTM.VALIDATION.FAILURE.IN) replace
define qlocal(BTM.DISPATCH.IN) replace
define qlocal(BTM.COMPLETE.IN) replace
define qlocal(BTM.COMPLETE.OUT) replace
```

for these two queues  
SYSTEM.BROKER.DC.RECORD- read

SYSTEM.BROKER.DC.BACKOUT – execute

Let me know if this is correct?

Thanks, Yvette

[Reply \(Edit\)](#)

MattClarke November 27, 2018

Hi Yvette,

So yes the MQ queues that you create for your BTM scenario would need READ and WRITE permissions.

The SYSTEM.BROKER.DC.RECORD and SYSTEM.BROKER.DC.BACKOUT queues should be created for you by running the iib\_createqueues command in order to create the default system queues (see [https://www.ibm.com/support/knowledgecenter/en/SSMKHH\\_10.0.0/com.ibm.etools.mft.doc/ah09090\\_.htm](https://www.ibm.com/support/knowledgecenter/en/SSMKHH_10.0.0/com.ibm.etools.mft.doc/ah09090_.htm)). Those would again need to be READ and WRITE permissions.

Regards,  
Matthew

[Reply \(Edit\)](#)

Kiren Pillay July 27, 2017

Hi Matt

Any idea when this feature will be made available for SQL Server. I've configured my broker such that I actually see the events in the WMB\_BUSTRANS table, however these transactions don't appear in the web console.

Regards  
Kiren

[Reply \(Edit\)](#)

MattClarke August 22, 2017

Hi Kiren,

At this time the Business Transaction Monitoring feature as per when it was introduced in 10.0.0.3, is not supported for SQL Server database at the latest IIB 10.0.0.9 level.

The only supported databases are DB2 and Oracle.

If this is a feature that you'd like to see added, you can submit an enhancement request here: <https://developer.ibm.com/integration/help/submit-rfe/>

Regards,  
Matthew

[Reply \(Edit\)](#)

kirenpillay August 23, 2017

Thank you Matthew,

I see its already been logged [https://www.ibm.com/developerworks/rfe/execute?use\\_case=viewRfe&CR\\_ID=81163](https://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&CR_ID=81163).

Hopefully more people will vote for this feature and we can have it soon!

Regards  
Kiren

[Reply \(Edit\)](#)

tektreesuman April 13, 2017

Hi,

I'm trying to working on record and replay but i faced issue like  
Capture Store could not be accessed  
error  
Data Capture Store could not be accessed  
BIP2111

[Reply \(Edit\)](#)

Jonatan Maya March 24, 2017

Hi Matt

A customer will implement BT in production environment. Is there a maximum number of BT that could be declared per IB Node? What would be the best cleanup and maintenance recommendation if they expect to store 24 Millions of events in a very busy day.

[Reply \(Edit\)](#)

MattClarke April 04, 2017

Hi Jonatan,

We have provided some details on Performance with BTM on this article <https://developer.ibm.com/integration/blog/2016/03/04/business-transaction-monitoring-performance-considerations/> There is no upper limit but performance could potentially be impacted once a certain number of events are being processed per second. Translating to the number of Business Transactions (BT) depends on the number of events that contribute to a BT.

There are potentially further considerations stated in the related articles that may affect performance:  
Planning – <https://developer.ibm.com/integration/blog/2015/12/07/advanced-usages-of-business-transaction-monitoring/#planbtm>  
When used with Record and Replay – <https://developer.ibm.com/integration/blog/2016/01/14/business-transaction-monitoring-vs-record-and-replay/>

For cleanup then you can archive your business transaction data as detailed in this article <https://developer.ibm.com/integration/blog/2016/01/13/archiving-business-transaction-monitoring-data/>

Hope this helps.

Thanks.

Matt

[Reply \(Edit\)](#)

Ajay November 29, 2016

Hi,

Please let me know if sql server is compatible to implement BTM in IIB v10.0.0.6.  
I have implemented Record and Replay with sql server DB set up in Iib v10, working as expected.

Getting the below sql exception in web UI when trying to view BTM transactions(but data is available in WMB\_BUSTRANS) table:

) Database error: SQL State "07009"; Native Error Code '0'; Error Text "[Microsoft][ODBC SQL Server Driver]Invalid parameter number".

The error has the following diagnostic information: SQL State "07009" SQL Native Error Code '0' SQL Error Text "[Microsoft][ODBC SQL Server Driver]Invalid parameter number"

This message may be accompanied by other messages describing the effect on the integration node itself. Use the reason identified in this message with the accompanying messages to determine the cause of the error. Use the mqsicvp command to test connectivity to this database.

[Reply \(Edit\)](#)

MattClarke December 06, 2016

Hi Ajay,

The Business Transaction Monitoring feature as per when it was introduced in FP3, is still not supported for FP6.

The only supported databases are DB2 and Oracle databases at this time.

Regards,

Matthew

[Reply \(Edit\)](#)

Paul November 23, 2016

With v10.0.0.4, all events are recorded in the event tables with times (varchar2) in UTC. Can the timezone of recorded events be changed?

[Reply \(Edit\)](#)

MattClarke December 06, 2016

Hi Paul,

When you mentioned the 'recorded events', do you mean the already recorded events in the WMB\_MSGS table?

The changing of the timezone of already recorded events, or changing the field type for the timestamp would be unsupported as it may impact the ability to record new or query existing Business Transaction results correctly.

Regards,

Matthew

[Reply \(Edit\)](#)

Paul December 07, 2016

Hi Matt,

I agree that changes the tables would likely break things. I was more asking if there was a setting to ensure what was placed into the tables was a local timezone value.

I've since been told via other channels that there is no such setting.

So... I created a Chrome Extension to fix the problem and add some hotkey function to the BTM interface.

Feel free to make use of it everybody.

'Improved IIB Business Transaction Monitor': <https://chrome.google.com/webstore/detail/improved-iib-business-tra/halnhmfjmdagcpcpjdbknhaagimpek>

[Reply \(Edit\)](#)

Islam\_Azaz September 19, 2016

Hi Matt,

In case You have a custom event definition; in xsd file,

1. Can you extend the table structure to include custom transaction attributes?
2. Can you modify the built-in Business Monitoring capabilities to persist and query user-defined fields?

[Reply \(Edit\)](#)

MattClarke September 26, 2016

Hi,

In response to your questions regarding Business Transaction Monitoring (BTM):

1. For BTM it is not possible to extend the main table (WMB\_BUSTRANS) to include custom transaction data.
2. Explicitly for BTM this is not possible. But you mention Business Monitoring. Are you referring to BTM (which this article is about) in this context or WebSphere Business Monitor?

Regards

[Reply \(Edit\)](#)

Islam\_Azaz October 04, 2016

Thank you for your response. The context of my question is "Service Audit", nothing complex, Our Audit table is keeping some extra

basic data about each transaction/job run. Nothing Needs WBM.

[Reply \(Edit\)](#)

Blee August 31, 2016

Where can I get a table schema for Oracle?

[Reply \(Edit\)](#)

Ian\_Larner September 12, 2016

Hi Blee,

To configure the database and tables on Oracle to record transactions, you can use the Oracle scripts provided with IIB. For more information about using the scripts for Oracle, see [Creating a business transaction definition](#) (which also links to) [Creating and configuring an Oracle database for recording data](#).

Regards,

Ian

[Reply \(Edit\)](#)

Blee September 12, 2016

Hi Ian:

Thanks for the response. I am able to configure and run the BTM now.

[Reply \(Edit\)](#)

Celia July 26, 2016

Hi Matt

It is possible that in the WMB\_BINARY\_DATA table instead of having DATA, have DATA1 and DATA2 ?

Thank you.

[Reply \(Edit\)](#)

MattClarke July 27, 2016

Hi Celia,

The WMB\_BINARY\_DATA table schema is fixed for field names. So you have to use the DDL structure provided, thus DATA for the binary field. It is not possible to change the name of the binary field / add more fields with different names.

Regards,

Matthew

[Reply \(Edit\)](#)