FIX to MQ Solution troubleshooting

Overview

- Several FIX sessions are connected to FIXEdge and sending messages to the middleware over IBM MQ WebSphere
- The middleware sends messages and responses back to the FIX Clients via MQ which is converted to FIX by FIXEdge MQ TA.

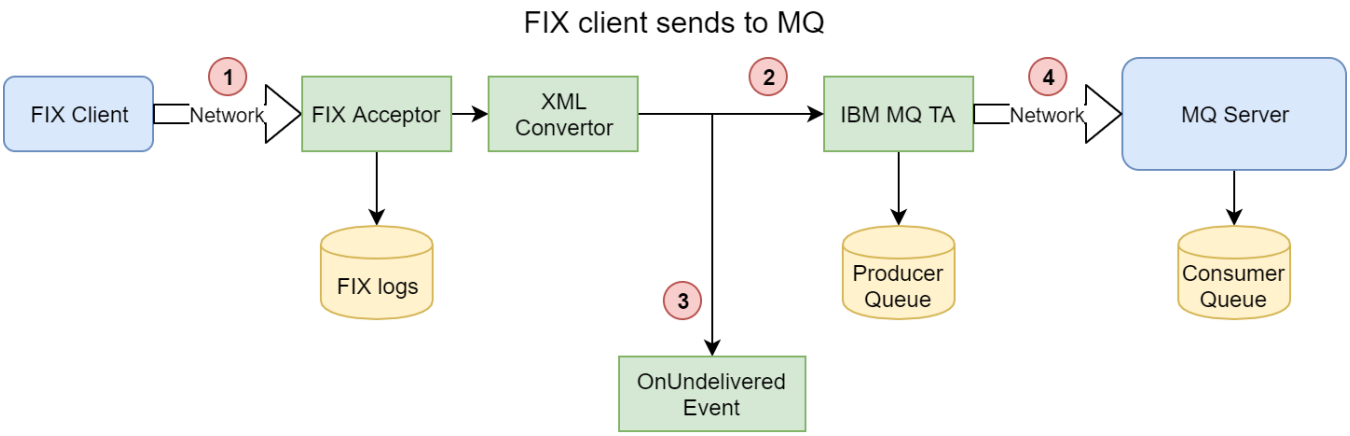
In this document, there is a detailed analysis of the reasons leading to missing/non-processed messages in your solution and troubleshooting recommendations.

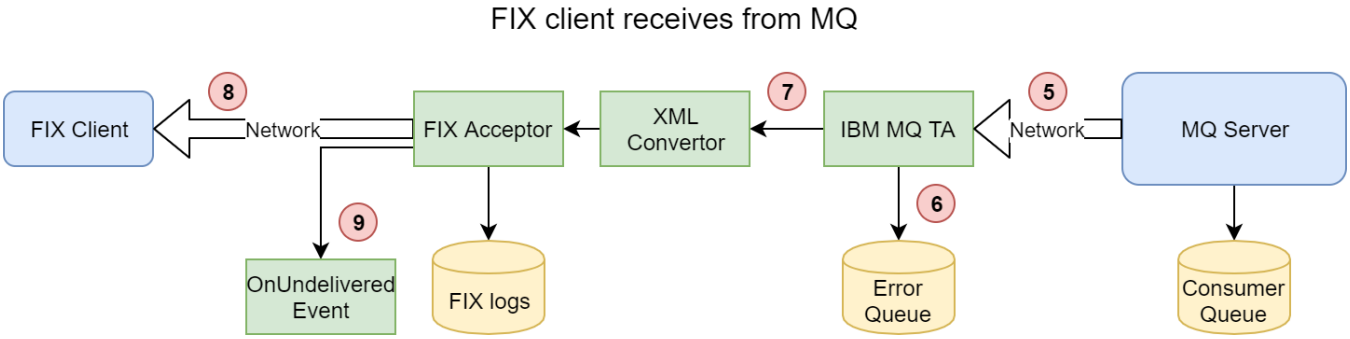
The goal that is taken into account is to exclude messages loss or assist the user with message recovery in case of various failures.

> Assuming that the configuration is working and there have been no recent changes in the configuration that may lead to the failure.

Failure scenarios

A detailed description of the failures are described below





1. The network between FIX Client (Sender) and FIXEdge is down.

The messages from the client are not delivered to FIXEdge. In this case, the messages are stored on the client's side.
After the reconnection, they recover automatically via standard resend request mechanism if the sender party keeps the sequences.

Exceptional case

When counterparty resets or lowers the sequences then the undelivered messages will be lost.

FIXEdge have 2 options to handle these cases:

- Do not accept a session with unexpected sequences (Default behavior).
  The FIXEdge user can detect this issue and should communicate with the sender in order to resolve the situation **manually**.
  In FIXEdge.properties:
  IgnoreSeqNumTooLowAtLogon = false
- Accept session with a low sequence. The missing messages will be not requested.
  The process of recovery becomes **automated**, no manual session recovery is needed by the cost of missing messages.
  If a party lowers or reset Sequence Number, it means that all previous messages are considered as non-important and can be lost.
  In FIXEdge.properties:
  IgnoreSeqNumTooLowAtLogon = true

Users should apply the best strategy for their case.

2. FIXEdge failure during FIX to MQ routing

If FIXEdge failure happens in the middle of routing the sequence number in FIX incoming logs is not updated so messages are considered as not received and will be recovered via FIX Standard resend request mechanisms after the restart. In some conditions, it may lead to duplicates. Missing messages are not expected in these cases.

3. The message doesn't reach MQ due to issues in the Business Rules

The messages can be not delivered to the MQ in some cases. FIXEdge has a mechanism for handling these situations.

FIXEdge detects an event on the Business Logic layer that can be captured by a user. The user can implement the logic for error-handling these kinds of errors.
For example, it is possible to send reject back to the sender or to a special session for manual processing these messages later.

Error events types for the configuration above:

- Business Logic OnUndeliveredMessageEvent event is called when the destination is not available.
- Business Logic OnRuleFailEvent is called when js execution fails or configuration is wrong.

in most cases, these issues are related to the wrong or inefficient configuration.

4. FIXEdge can't push the message to the IBM MQ Server

FIXEdge saves a message to the producer queue with persistent storage before sending it.

In the case of FIXEdge failures or, network failures, or if the connection is dropped all non-delivered messages will remain as undelivered in the storage. O nce the connection will be recovered, FIXEdge tries to finish the delivery and resend messages to IBM MQ Server.

In case if the IBM MQ server (WebSphere) is not available (e.g. stopped) there will be the next error in the logs.

```
ERROR [MQQueueWrite] <thread> MQ PUT Commit on queue '<src queue>' failed
```

5. FIXEdge can't get from a message from IBM MQ

Since the messages are saved on the IBM MQ server-side, they will be automatically recovered when IBM MQ Server becomes available from the FIXEdge server.

The recovery process may lead to duplicates if a transaction mode is enabled or missing messages if the transaction mode is disabled

FIXEdge requests the whole transaction if the network drop happens in the middle of the transaction so the messages that were processed before connectivity issues will be reprocessed/resent to the counterparty again

6. IBM MQ TA errors during message processing on BL

The message can be delivered to an error queue instead of FIX recipient.

MQ TA handles errors when sending messages to BL. If there is an error then the messages are sent to a special error queue so the messages can be processed later, e.g. when the root cause of the problem is resolved.

The only reason for the errors if there is no destination is configured for sending received messages from MQ Server. In most cases, it is related to wrong /invalid Business Logic configuration like there are no specific conditions fit a received message.

Transactions in IBM MQ TA

MQ Transport Adapter has a mechanism of sending messages as batches within a single transaction. This mechanism is used to detect message loss and recover missing messages.

The batch is considered as delivered if MQ Adapter gets confirmation about the delivery of all of the messages from the batch. Failure of sending even a single message makes the transaction unsuccessful and is resulted in all batch rollback.

If the transaction mechanism is enabled it may lead to message duplicates after recovery in some cases. In order to reduce the number of duplicates, it is recommended to use a transaction batch with a small size.

However, disabling the transaction mechanism may lead to missing messages in case of failures during receiving messages from MQ.

It is designed to increase the performance of the MQ TA.

MQ Client version 7 doesn't have the full support of batch sending so sending performance is not increased.

Additional information about persisting state of transactions can be found: Transactions in MQ

Error queue management

The messages for the error queue can be re-processed using dmpmqmsg tool. The tool can move messages across queues and should run on the server with IBM WebSphere.

Example of moving messages from the queue 'FIX.SESSION1.ERROR' to 'FIX.SESSION1.IN' with the content of 'TEST_SENDERID':

```
dmpmqmsg -m TestMQ -IFIX.SESSION1.ERROR -oFIX.SESSION1.IN -sTEST_SENDERID
```

Where,

-m - QueueManagerName, see **TransportLayer.MQAdaptor.MQ_MANAGER_NAME** parameter
-I - Input queue name, e.g. name of Error Queue, see **TransportLayer.MQAdaptor.Session.1.ErrorQueue** parameter
-o - Output queue name, in most cases, should be consumer queue, see **TransportLayer.MQAdaptor.Session.1.FromClientQueue** parameter
-s - Message content, In the example above we use senderCompId for the target FIX session for filtering.

It is recommended to use different Error Queues per session for simplifying troubleshooting of the issues. If messages are related to a session it is easier to create filtering rules.

The queue per MQ session is specified with **TransportLayer.MQAdaptor.Session.<N>.ErrorQueue** parameter

7. FIXEdge problems with routing from MQ to FIX

If there was a FIXEdge failure when messages are in transit from MQ to FIX there is a chance to lose the messages. E.g. when FIXEdge marked the message as received from MQ but the message is not sent to the FIX recipient.

All such cases should be considered independently

The messages that came from transport adaptors must be routed somewhere. If the destination specified in the BL rule is not found then FIXEdge will raise a "Sending message to TL failed" error.

The user can use the Condition element for filtering messages to specifying the destinations. If there are no suitable conditions in the BL rule the message will not be delivered anywhere and the will be the next error.

fixedge.log

```
2021-07-07 17:45:49,565 UTC   DEBUG   [BL_RoutingTable]  24288  No BL rules found for message with ClientID
'MQHub2', executing DefaultRule.
2021-07-07 17:45:49,565 UTC   DEBUG   [CC_Layer]  24288  BL has processed a message. Number of client IDs for
delivery :0. Number or FIX sessions for delivery :0.. Number or sources identifiers for delivery :0.
2021-07-07 17:45:49,565 UTC   ERROR   [MQQueuesReader]  24288  Sending message to TL failed. Reason: TL has
returned false.
2021-07-07 17:45:49,565 UTC   DEBUG   [MQQueuesReader_Debug]  24288  Sending to Error queue.
```

Note

Conditions can work with regex

For Example

If the following BL rule results with the above error, that means that the incoming message does have ".*MATCHTHISTEXT.* in the tag 55

BL_Config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<FIXEdge>
    <BusinessLayer>
        <!-- Routing Rule for MQHub incoming messages -->
        <Rule>
            <Source Name="MQHub2" />
            <Condition>
                <MatchField Field="55" Value=".*MATCHTHISTEXT.*"/>
            </Condition>
            <Action>
                <Send Name="FIXCLIENT" />
            </Action>
        </Rule>
        <DefaultRule>
            <Action>
                <DoNothing/>
            </Action>
        </DefaultRule>
    </BusinessLayer>
</FIXEdge>
```

To troubleshoot the user should check message content if message content fits if is not fitting apply additional conditions covering adding in the rules for another condition

When there it will be a destination for the message, the user will have the next output specifying that the message has a delivery point.

fixedge.log

```
2021-07-22 20:56:45,325 UTC   DEBUG   [BL_RoutingTable]  8848  Found 1 suitable rules for message with ClientID
'MQHub2'
2021-07-22 20:56:45,325 UTC   DEBUG   [CC_Layer]  8848  BL has processed a message. Number of client IDs for
delivery :0. Number or FIX sessions for delivery :0.. Number or sources identifiers for delivery :1.

    Source IDs:
        1. 'FIXCLIENT'

2021-07-22 20:56:45,325 UTC   DEBUG   [FL_MsgTrace]  8848  Sending FIX message to session with ID 'FIXCLIENT'.
Message: '8=FIX.4.49=15835=D49=FIXCLIENT56=FIXEDGE34=252=99990909-17:17:
1711=Order#064=2021021921=3100=155=MATCHTHISTEXT54=160=20210219-04:11:46.76838=2000040=244=34.710=065'.
2021-07-22 20:56:45,437 UTC   DEBUG   [MQQueueRead]  8848  MQ GET Commit took 111.15ms, res: 1
2021-07-22 20:56:45,437 UTC   DEBUG   [MQQueuesReader_Debug]  8848  1 messages are removed form client Queue.
```

8. The network between FIX Client (Recipient) and FIXEdge is down or the session is offline

The messages from the FIXEdge are not delivered to the FIX Client if there are network issues. After the reconnection, the messages are recovered automatically via a standard resend request mechanism.

By default, the messages are stored on the FIXEdge side before sending it.

The messages can be lost when FIXEdge reset sequences by the schedule. All undelivered messages will be lost

If configuring proper schedule is not flexible enough and it is required to deliver missing messages the next day after sequence reset, it is recommended using the Managed Queue feature implementing a store-and-forward workflow.

The Managed Queue feature was introduced in FIXEdge 6.9.0

9. OnUndeliveredMessageEvent on sending to FIX message.

The message can be not delivered to the FIX Client in some cases.

It happens when FIXEdge tries to send a message from MQ but the session is offline and property for rejection fix messages in case of disconnection is set:

> FixLayer.FixEngine.Session.Session_Name.RejectMessageWhileNoConnection = true

- Business Logic OnUndeliveredMessageEvent event is called in these cases

It is not recommended to use logic for sending messages in the same session again if the first attempt has failed and the OnUndeliveredMessageEvent event is called.

MQ TA Configuration

Property	Description	Required	Default value
TransportLayer. TransportAdapters	**The list of user-defined Transport Adapter names. The several instances should be separated by a comma.** The following configuration will show configuration parameters for an adapter with the name **TransportLayer.MQAdaptor**, i.e.: TransportLayer.TransportAdapters = TransportLayer.MQAdaptor		
MQ parameters			
TransportLayer.MQAdaptor. NumAttemptReconnect	(deprecated) Defines a number of reconnection attempts if MQ Server not available. The value of -1 means for unlimited reconnection attempts. Should be always -1		-1
TransportLayer.MQAdaptor. TimeIntervalBeforeReconnect	Time interval in milliseconds between reconnection attempts		1000
Transaction parameters			
TransportLayer.MQAdaptor. UseTransactions	Enables/disables transactions mode: • **false** – disable batch sending in a single transaction • **true** – enables transactions, messages are sent as batches. Unsuccessful delivery results with a new attempt to send the whole batch.	No	false
TransportLayer.MQAdaptor. Session.<N>. ReceiveTransactionSize	The maximum number of messages the batch for receiving from the MQ Server in a single transaction. Should be positive. The bigger value increases performance.	Required if UseTransactions is set to '**true**'	10
TransportLayer.MQAdaptor. Session.<N>.SendTransactionSize	The maximum number of messages the batch for sending to the MQ Server in a single transaction. Should be positive.	Required if UseTransactions is set to '**true**'	10