C:\Users\vrush\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\image.jpeg

Broadridge Trade Transmission Process Flow

* SQL function specifications.

Version:2.0

1- Trades From Gresham and FTP

- Step 1: Trade data from Gresham or FTP feeds into the tradeloadstaging.dbo.messageinbound table.

The sample data of table [Tradeloadstaging.dbo.Messageinbound]



Database: Tradeloadstaging

- Step 2: The trigger kicks off in the tradeloadstaging.dbo.messageinbound table and loads the trade data into the db.trade table in the Trade Transformation/IREP Staging DB Server 10.1.10.15.

The trigger [Name:ProcessTrade]:

1. Loop for each inserted trades
2. Revoke a stored procedure [ServiceBroker].[sp\_serviceBroker\_add\_trade].
3. Call Service Broker to load trades into tradeloadstaging.dbo.trade table

---------------sp------

ProcessTrade

->ServiceBroker.sp\_serviceBroker\_add\_trade

->[ServiceBroker].[sp\_serviceBroker\_Receive\_trade]

->[sp\_Load\_OutrightTrades\_realtime\_oneby\_one]

-> FROM dbo.f\_AllTranslated\_outright\_trades(@recvedmessagekey,@processtype,@system) a

->V\_FIX\_Final

->V\_FIX

->InsertInto Trade

- Step 3: The trigger kicks off in the db.trade table to transform trades into FIX message, puts them in the MSMQ and sends them to GEMS FIX Server 10.1.10.6.

Trigger name: [SendTrade] on Server tradeloadstaging

Call sp: upload\_trd\_to\_gems\_realtime\_fixedincome\_tradefeed\_onebyone

1. Update column source\_BRID
2. Call get\_TradeOutboundMQConfig
3. Function Fn\_send\_single\_msmq\_mesage
4. call Assemblies MSMQ.dll.

- Step 4: During the transformation process, the system generates status logs for all FIX trade messages and saves them in the messageinbound\_audit table. The logs show if the trade successfully transformed into an FIX message or failed.

Note: messageinbound\_audit has the trades from Gresham/FTP and portal, and it does not works with the table tradeloadstaging.dbo.gems\_acknack.

- Step 5: Once the trades are in the GEMS FIX Server 10.1.10.6, Raisul’s FIX Consumer process picks the messages and sends them to Broadridge GEMS and APPIA. This process runs in every 1 minute.

SQL Agent Job load\_gems acknackmessage usingSSISpackage in server 10.1.10.15 loads the data into the tradeloadstaging.dbo.gemsacknack table.

2- Trades From Client Portal

- Step 1: Trade data from the Client Portal feeds into the tradeloadstaging.dbo.trade\_gui table.

- Step 2: The trigger kicks off in the tradeloadstaging.dbo.trade\_gui table and loads the trade data into the db.trade table in the Trade Transformation/IREP Staging DB Server 10.1.10.15.

Tragger [sendTrade\_gui]

Stored Proc : GUIFInanceTradeUpload

[upload\_trd\_to\_gems\_realtime\_fixedincome\_oneByOne]

-> fn\_return\_fi\_fixMessage\_gui = fn\_return\_fi\_fixMessage

-> select [dbo].[fn\_return\_fi\_fixMessage\_gui](1586688)

- Step 3: Same steps 3 through 5 in section 1

Add a Trade from Client Portal, and display



Step1: add a trade

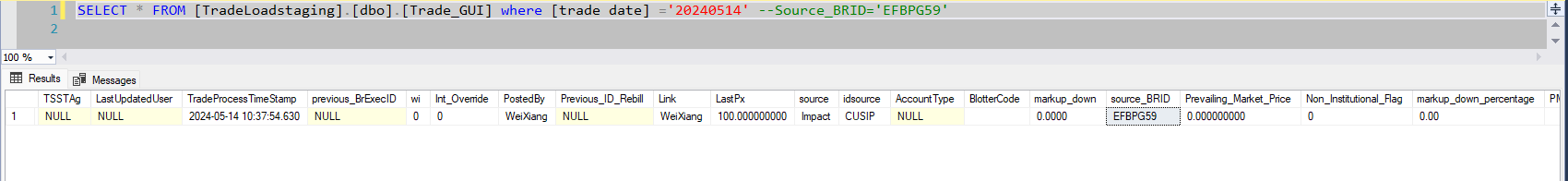
A screenshot of a computer

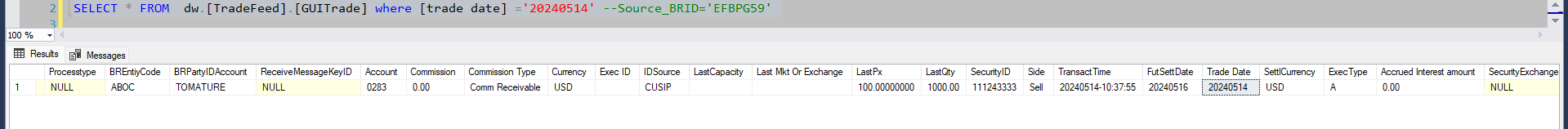
Description automatically generated

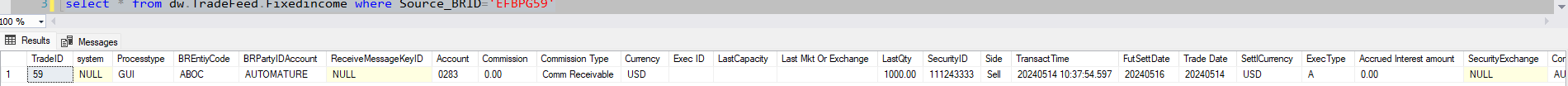
Step2: search the trade just added.

A screenshot of a computer

Description automatically generated







Troubleshooting

Portal: 1.Security not found: Use 9128% like as security ID

2.Status = Reject means there is an issue.

3. ExecID is empty, how to test on full cycle?

Repo the trade.

3- GEMS ACK/NACK Process

- Step 1: Once GEMS receives trade message from GEMS FIX Server 10.1.10.6, it returns ACK or NACK message as a confirmation. The ACK message indicates that the trade is booked successfully and the NACK message indicates that an error occurred when booking the trade.

- Step 2: Raisul’s FIX session puts the ACK/NACK message in the MSMQ and the SQL Agent Job load\_gems acknackmessage usingSSISpackage in server 10.1.10.15 loads the data into the tradeloadstaging.dbo.gems\_acknack table.

Stored procedure: GetFixackNackMessage\_from\_stage

select \* from [dbo].GemsAckNACK

select distinct messageprocessstatus from dbo.vw\_gems\_acknack\_from\_stage

Note:

1. To cross check with the trades, link gems ack nack column Exec ID with table column Source\_BRID to join check the process status column.
2. Column OrderStatus categorlize the status of trade.
3. New
4. Rejected

4- Data Load into DW Database and Display on Client Portal

The SQL Job TradeFeed\_Load in DW Database (server 10.1.10.13) runs every 1 minute to load trade and ACK/NACK data from the Staging Database (server 10.1.10.15) into the DW Database as described below.

Stored proc ICBCFSAPPDB.[Screen].[CRUD\_FI\_trade\_FRM\_GUI\_update]:

Loop each transaction from dbo.trade\_gui

Update table dw.irep.TRANFILP\_DW

table dw.Derived.outrightdescription

Delete Trade from dw.TradeFeed.gemsacknack

Update trade from dw.TradeFeed.Fixedincome

Trade from dbo.trade\_gui

Stored procedure [TradeFeed].[LoadTradedata]:

from irep.TradeLoadstaging.dbo.dweqtrade

select \* from dw.TradeFeed.Equity

select \* from irep.TradeLoadstaging.dbo.[DWgems\_ack]

select \* from dw.TradeFeed.gemsacknack

select \* from irep.TradeLoadstaging.dbo.dwFItrade

from dw.TradeFeed.Fixedincome

select TETID from irep.TRANFILP\_DW where TCRVI=''

select TETID from irep.TRANFILP\_DW where TCRVI='CXL'

For the trades, the job takes centralized data from the dbo.trade and dbo.trade\_gui tables and loads it into the DW.tradefeed.fixedincome table.

For the ACK/NACK, the job takes the data from the tradeloadstaging.dbo.gemsacknack table and loads it into the DW.tradefeed.gemsacknack

The Trade Processing Queue function in the portal takes the data from the DW.tradefeed.fixedincome and DW.tradefeed.gemsacknack tables and displays it on the portal GUI.

All trades in the portal Trade Processing Queue can be cancel or rebill.

Rebill stands for cancel and new trade, for cancel trade it will same Exec ID for new trade new exec ID,

Currently, from business perspective, Trader can make same account of identical trades.

Appendix



Reference Mapping

Sumary Steps:

USE [TradeLoadConfig]

GO

1.Account and entityID: [TradeLoadConfig].[dbo].[TraderConfig\_manual\_entry]

K.foespat4 the pre-fix "K" , need to match pre-fix 'BFIX"

i,e change to "B.foespat4".  I tested it, it works.

2. BRPartyIDAccount: ].[FCCSystemDetails].

I added below missing mapping, then the fields is populated, see below result.

image

3. BRPartyIDAccount -CounterPartyID match: INSERT INTO [dbo].[BR\_ICBC\_AcountMapping]

([corr\_sys\_office\_acct\_subacc]

,[br\_acct]

,[MS\_Access\_ID]

,[CORR\_CD]

,[SYS\_CD]

,[OFFICE\_CD]

,[ACCOUNT\_CD]

,[SUB\_ACCOUNT]

,[BPS\_BRANCH]

,[BPS\_ACCOUNT]

,[BPS\_TYPE]

,[BPS\_CHK\_DIG]

,[TSS\_SYS]

,[FED\_FLAG]

,[Aux\_file]

,[Updated\_from\_John\_Lee]

,[Old\_bps]

,[Old\_bps1]

,[Shortcode]

,[BREntityCode])

select NULL,NULL,NULL ,NULL ,'FOS' ,NULL, '65041018' ,NULL ,'126', '80008', '9','6', 'FOS', NULL, NULL, NULL, NULL, NULL, 'FEN PRE', '4170'

GO

--------------------------------------------------------------------------------------------------

------Account and BREntityCode

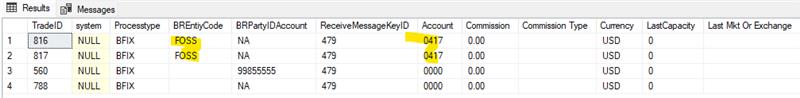
they are added in SELECT [Trader]  
      ,[Account]  
  FROM [TradeLoadConfig].[dbo].[TraderConfig\_manual\_entry]

I found the root cause of the issue.

K.foespat4 the pre-fix "K" , need to match pre-fix 'BFIX"

i,e change to "B.foespat4".  I tested it, it works.

Select \* from DBo.Trade where processtype='BFIX' and receivemessagekeyid in ( 479,128 )



---------------BRPartyIDAccount

hi Ravi,

The reason BRPartyIDAccount is not populated is because mapping is missing in [TradeLoadConfig].[dbo].[FCCSystemDetails].

I added below missing mapping, then the fields is populated, see below result.

Note: I don't know what value should be forNewSystem.  so I copied from another 4170 row.

image

 -------------------------



MessageInbound -> Trade [ProcessTrade]

sp\_serviceBroker\_add\_trade

sp\_serviceBroker\_Receive\_trade

sp\_Load\_OutrightTrades\_realtime\_oneby\_one

Trade -> MQ FIX-> Broadridge