Nasdaq Calypso

Refinitiv Trade Notification Integration Guide

Version 3.13.0

Revision 12.0
January 2025
Approved

Document History

Revision	Published	Summary of Changes
1.0	May 2020	First edition for version 1.0.0
2.0	August 2020	Second edition for version 1.1.0
3.0	October 2020	Third edition for version 1.1.1
4.0	February 2022	Fourth edition for version 2.0.0, 2.0.1, 2.0.2 – Technical release only – Version 17.0 compatibility
5.0	September 2023	Fifth edition for version 2.5.0 – Generation of acknowledgements
6.0	January 2024	Sixth edition for version 3.3.0 – Technical release only – Compatibility with version 18
7.0	February 2024	Seventh edition for version 3.4.0 - Enhanced logic to get the book
8.0	May 2024	Updates for version 3.6.0 – Added TOF to Refinitiv Trade Notification Migration
9.0	June 2024	Updates for version 3.7.0 – Added Conversation Capture Report
10.0	July 2024	Updates for version 3.8.0 – Added support for FX options
11.0	November 2024	Updated for version 3.12.0 – Added non deliverable currency mapping.
12.0	January 2025	Updated for version 3.13.0 – Added Structured Flows direction type mapping, engine parameter OPTIONAL_FEATURE.

This document provides setup information for the Refinitiv Trade Notification interface.

NOTE: The Calypso License to use this Calypso Integration Module does not include a license for any third-party data services to which this module can interface. Clients are responsible for contracting with the appropriate third-party data service(s) prior to using this Calypso Integration Module

Table of Contents

1 Introduction

This document describes the Refinitiv Trade Notification interface setup.

The Refinitiv Trade Notification interface allows an End User Firm (Buy-Side) to import trades which are delivered by Trade Notification. Trade Notification is a secure messaging hub that captures trades electronically and delivers notifications instantly for both voice and electronically executed trades.

The Refinitiv Trade Notification integration connects to Trade Notification service using a FIX interface. Once the connectivity is setup, a trader can book a trade through the FXall Trading Terminal. The Refinitiv Trade Notification is an electronic publishing venue that uses the Refinitiv Trade Notification service to publish trade confirmations to its receiver clients who execute trades on the Refinitiv Trade Notification platform. Once the trades are booked, Trade Notification will send notifications as a FIX message to the Refinitiv Trade Notification interface. The message will then flow through the configured Calypso workflows which route the message through the appropriate stages to create a Calypso trade.

This document describes the configuration required to setup the workflows, etc. for the Refinitiv Trade Notification interface to run successfully.

1.1 Supported

The Refinitiv Trade Notification interface supports the following FX trades:

Product Type:

- FX Spot
- FX Forward
- FX NDF
- FX Swap
- FX NDF Swap
- FX Options (Vanilla strategy, Drop copy messages, ACK)
- Loan/Deposit
- Structured Flows

Trade lifecycle:

- NEW

Fix Messages:

The interface supports the following FIX messages:

- Trade Capture Report (AE): Once the trade is executed within the FXall Trading Terminal, Trade Notification will send notifications over the FIX interface. A Bilateral trade will be created in Calypso to represent the trade between the two parties (Dealer vs. Buy-Side)
- Trade Capture Report ACK (AR)
- Conversation Capture Report (CCR)

1.2 Not Supported

The Refinitiv Trade Notification interface does NOT support the following:

Trade lifecycle:

- AMEND
- CANCEL

Fix Messages:

- Allocation Report (AS)
- Allocation Report ACK (AT)

2 Installation

2.1 Software Requirements

2.1.1 Supported JRE Versions

Please use the appropriate JRE version depending on the supported version for the base Calypso release you are running.

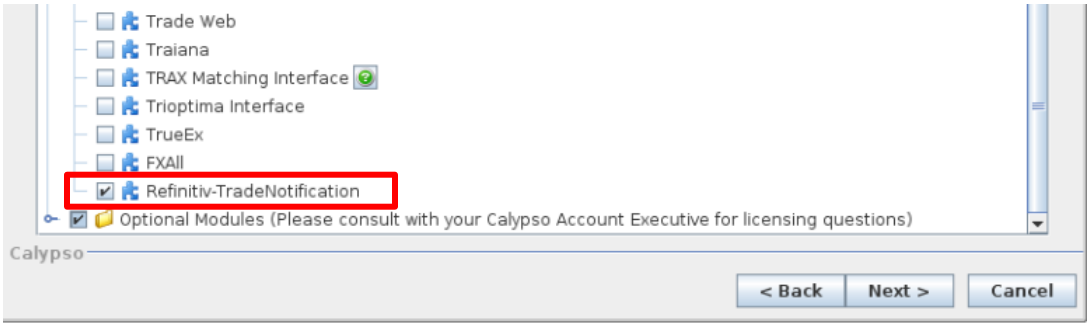
2.1.2 Supported Calypso Versions

Please refer to the Calypso Refinitiv Trade Notification release notes.

2.2 Installation Instructions

2.2.1 Calypso Components

Follow the Calypso System Guide “Installation and Upgrade” to install Calypso. Check the component Refinitiv-TradeNotification during the installation.



The use of the Refinitiv Trade Notification interface requires the Data Uploader. All subsequent instructions assume that all Data Uploader installation steps have been completed successfully.

This includes:

- Choosing Data Uploader from within the Calypso Installer (or using the patch tool to add it to an existing installation)
- Applying the Gateway SchemaBase/SchemaData to your database
- Setting up the GATEWAYMSG and UPLOADESOURCEMSG workflows
- Setting up Task Station tabs for the workflows above

Please refer to the Calypso Data Uploader Integration Guide for specific installation and configuration information. You **must** install and configure Calypso Data Uploader prior to configuring the Refinitiv Trade Notification interface.

2.2.2 Setup Config Data using Execute SQL

Add the following files to Execute SQL from $CALYPSO_HOME/bin/dbscripts, if not already present:

- SchemaBase.xml
- GatewaySchemaBase.xml
- FIXSchemaData.xml
- TRTNSchemaData.xml

2.2.3 Message Workflow Setup

The Refinitiv Trade Notification interface uses the UPLOADSOURCEMSG and GATEWAYMSG workflows when importing messages. These should have been setup as part of the Data Uploader setup.

Messages from the UPLOADSOURCEMSG workflow are translated from the external message format into Calypso's internal format and placed in the GATEWAYMSG workflow. The GATEWAYMSG workflow then translates the internal format, performs verifications, and saves the trade to the database.

2.2.4 Task Station Setup

The Refinitiv Trade Notification interface uses the Data Uploader framework to create task station entries for all the messages and exceptions that are encountered. The user can view / reprocess the messages that are failed in validation from the task station.

Please refer to the Calypso Data Uploader Integration Guide for how to add the appropriate messages and exceptions to the Task Station.

3 Setup Requirements

3.1 Legal Entity Mapping

The incoming FIX messages contain Legal Entity identifiers for Party and Counterparty involved in the trade.

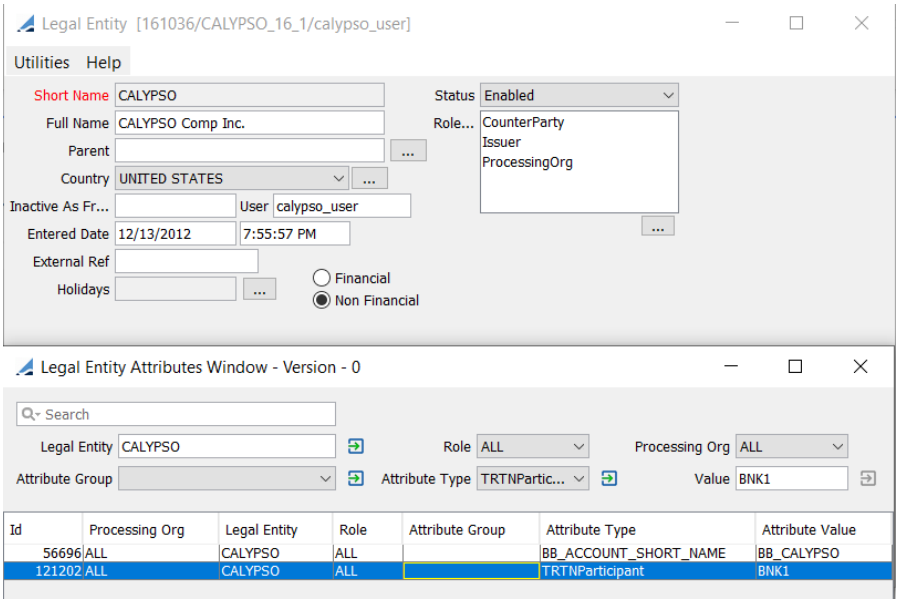
The Legal Entity for the Party and Counterparty identifiers populated in the FIX message need to be configured in Calypso using the Legal Entity attribute **TRTNParticipant**, and the value would match the bank provider value provided in FIX message.

The logic will look for the attribute value **TRTNParticipant** configured on the Legal Entity involved in trade booking (CP-LE) which will be used to generate the counterparty for legal entity. If not found, it will search for a Legal Entity having a matching Short Name (case-sensitive or all uppercase).

This lookup logic will be applied to Party and Counterparty. If no Calypso Legal Entity is found using the rules above, an error will be raised.

3.2 Counterparty Mapping

For the Calypso legal entity which is intended to be used as counterparty in trade booking must be mapped to attribute **TRTNParticipant** with the value configured in the Refinitiv Trade Notification platform. See illustration below.



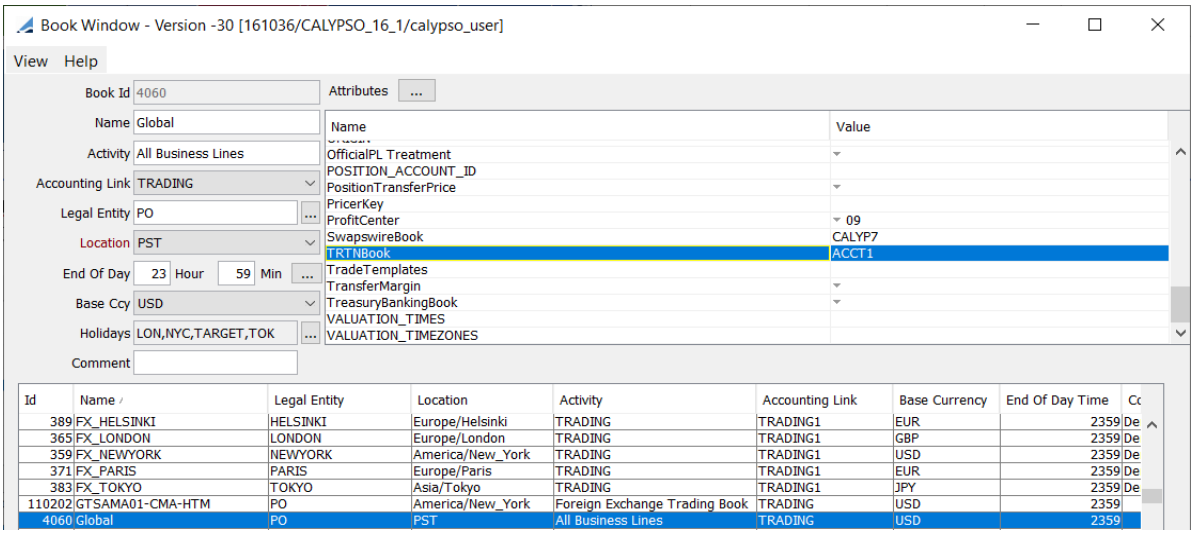
Mapping for TOF Migration

See TOF to Refinitiv Trades Notification Mapping for Migration.

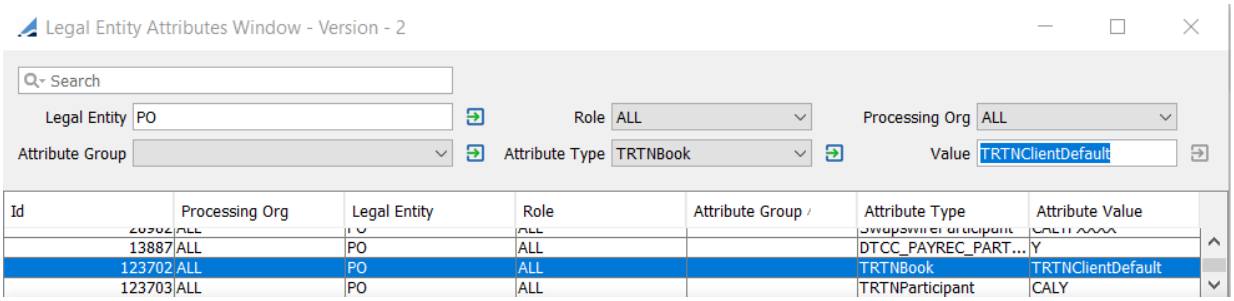
3.3 Book Mapping

Check if tag 79 is present from 78 group. If yes, then check if there is a book with book attribute TRTNBook = value of Tag 79, or book name = value of Tag 79.

If no book is found, get the value of Tag 448 having 447=D and 452=3, and check if there is a book with book attribute TRTNBook = value of Tag 448.



If no book is found, get the PO with legal entity attribute TRTNParticipant = value of Tag 448 and the book from legal entity attribute TRTNBook.



Mapping for TOF Migration

See TOF to Refinitiv Trades Notification Mapping for Migration.

3.4 Trade Keywords

Trade keyword InstrumentUPI and product code UPI are populated with incoming UPI code from Refinitiv Trade Notification for FXForward, FXNDF, FXSwap, FXNDFSwap.

Trade keyword ReportTrackingNumber with value from tag 1903 when tag 1906 = 6.

Mapping for TOF Migration

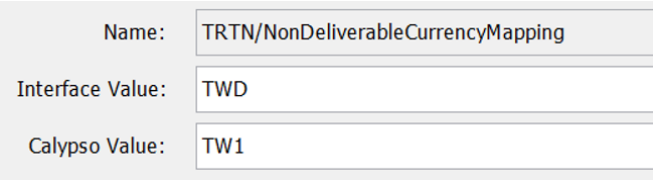
See TOF to Refinitiv Trades Notification Mapping for Migration.

3.5 Product Mapping

When importing product-based trade types into Calypso, mappings must be setup so that the Refinitiv Trade Notification interface can match the incoming product details to objects contained within the client's instance of Calypso.

3.5.1 FX Trades

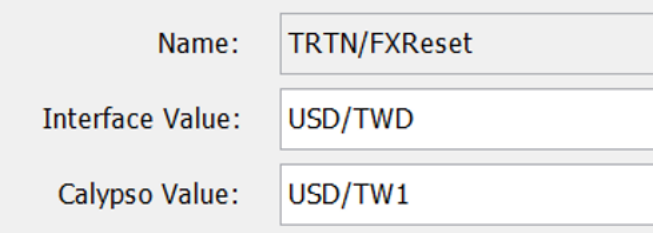
For saving the non-deliverable FX trades in Calypso, FX-Reset is required. User can map the given reset with the Calypso FX-Reset using the Calypso Mapping window.



Name = TRTN/NonDeliverableCurrencyMapping

Interface Value = <TRTN currency from Tag 55>

Calypso Value = <Calypso currency>



Name = TRTN/FXReset

Interface Value = <TRTN currency pair from Tag 55>

Calypso Value = <Calypso currency pair>

When the mapping is not specified, Tag 55 is used.

3.5.2 Loan/Deposit and Structured Flows

Product Type

Name:	TRTN/ProductType
Interface Value:	MMProduct
Calypso Value:	Cash

When Interface Value = MMProduct and Calypso Value = Cash, a Loan/Deposit trade is booked. Otherwise, a Structured Flows trade is booked.

Direction Type

Name = TRTN/Transfer

Interface Value = MoneyMarketTradeDirectionType

Calypso Value = Interest or Principal

If not set, it defaults to Interest.

Trade Direction

Name:	TRTN/MoneyMarketTradePayReceive
Interface Value:	1
Calypso Value:	Pay

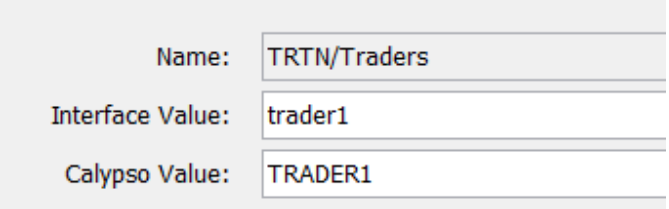
Name:	TRTN/MoneyMarketTradePayReceive
Interface Value:	2
Calypso Value:	REC

Coupon Daycount

Name:	TRTN/CouponDayCount
Interface Value:	123
Calypso Value:	ACT/365

3.6 Trader Mapping

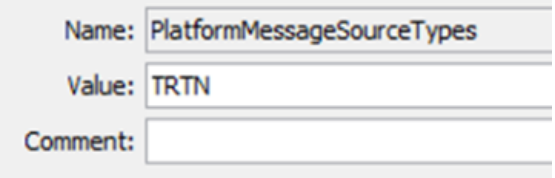
The Processing-Org trader present in fix message can be mapped to the TRADER available in Calypso. In Calypso Window mapping, the incoming trader name can be mapped to the existing TRADER in Calypso as shown below.



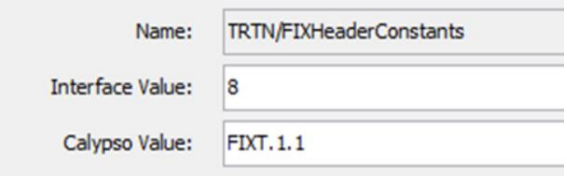
But if no mapping is found, then by default Trader is set with what is present in the incoming message.

3.7 Acknowledgements

You need to add TRTN to the “PlatformMessageSourceTypes”:



You need to map the FixHeaderConstant as shown below:



The following mapping is used to control the generation of acknowledgements.

Name = TRTN/Translator

Interface Value = PublishAcknowledgement

Calypso Value = true/false

When Calypso Value = true, acknowledgements of type PLATFORMMSG are sent to the Refinitiv TRTN platform from Refinitiv TRTN import, as soon as a message is received from TRTN, whether the trade is created in Calypso or not.

If the trade creation failed due to missing mapping or static data issue, then the user can reprocess the same message after fixing the mapping or static data.

Also make sure that the TRTNFixEngine subscribes to PSEventPlatformPublish events.

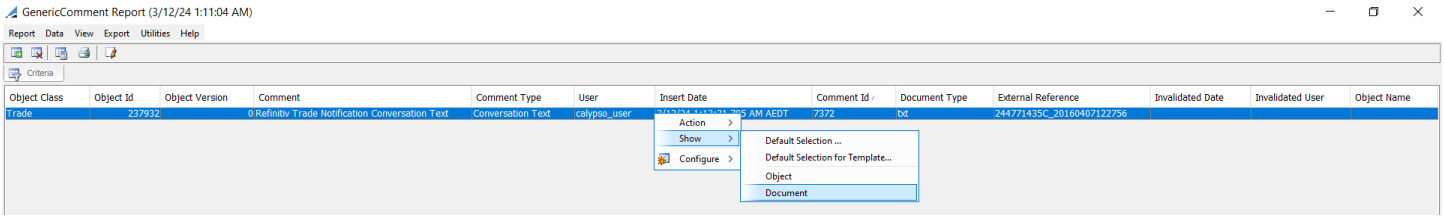
3.8 Conversation Capture Report (35=CCR)

In Refinitiv Trade Notification, Calypso supports the Conversation Capture Report (CCR). This CCR is sent by Refinitiv platform to Calypso which is saved as Generic Comment in Calypso.

Calypso will then send an acknowledgement of CCR depending on the type of FIX session configured by Refinitiv (All Ack or No Ack).

After mapping the config for - PublishCCRAcknowledgement in Calypso Mapping Window, when true = CCR acknowledgement it is sent back to Refinitiv platform. This flag is set as false by default.

For example, if the trade is created in Calypso, then Generic Comment will be linked to the trade. Value of Tag 11092 will be saved as a Document.



Generic Comment can also be viewed from trade via Pricing sheet.

3.9 TOF to Refinitiv TRTN Mapping for Migration

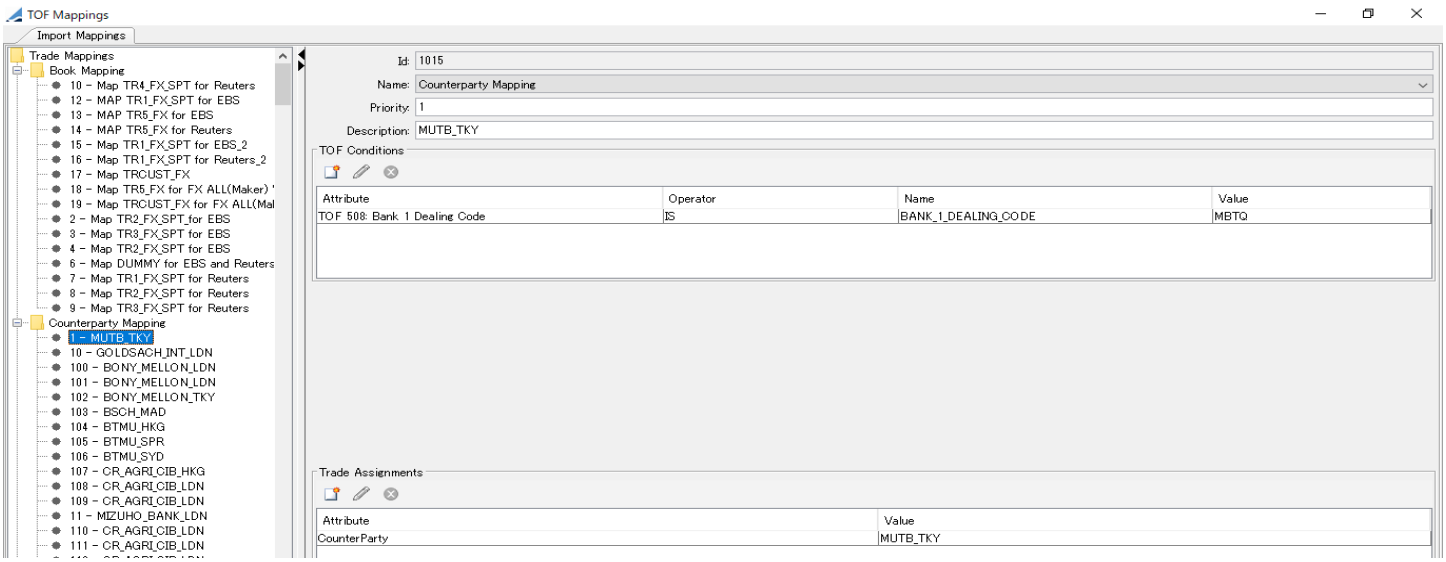
3.9.1 Counterparty Mapping

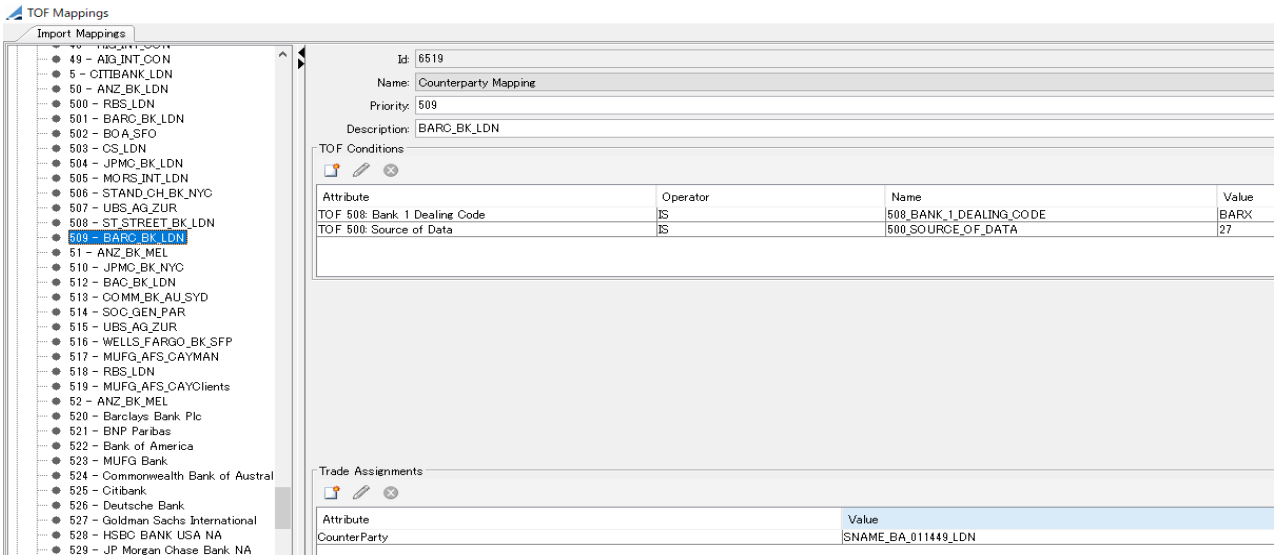
If (TOF 500: Source of Data is In (1,2) and TOF 504:Dealer ID is JS) map this TOF message to Calypso Book TR4_FX_SPT. If (Tag 11001 is IN (1,2) and (Tag 448 448 when 452=36 (entering trader) Or 452=11 (executing trader)) is JS) map this FIX message to Calypso Book TR4_FX_SPT.

Here, let's say for the following cases of counterparty mapping, relationship between TOF field and FIX tag are as follows:

TOF field	FIX tag
508: Bank 1 Dealing Code	448 when 452=17 (OBO trade) Or 452=1 (normal trade)

Refer to the following screenshots:





TOF Interpretation

When the value of TOF tag 508 = MBTQ, then counterparty in Calypso should be MUTB_TKY

Refinitiv Trade Notification Mapping

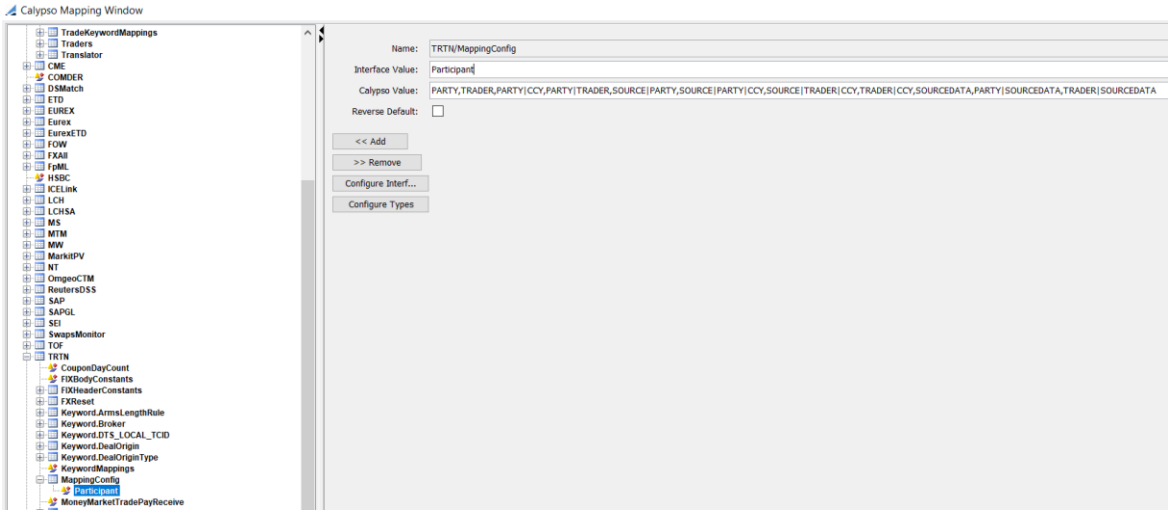
Get the value of 448 tag and map to Counterparty when 452=17 (OBO trade) or 452=1 (normal trade)

This can be done in 2 ways: All the below approaches are sequential. If the mapping via MappingCriteria is not found, then fallback to 2nd approach.

Calypso provides mapping in the following way:

User needs to define a MappingConfig

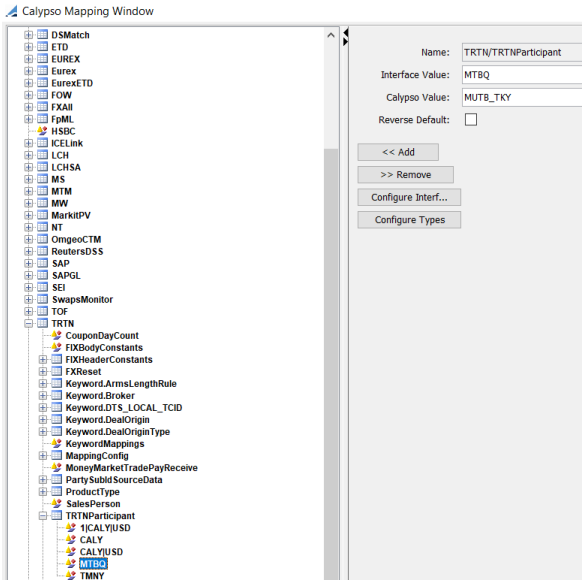
Interface Value = Participant
Calypso Value =
PARTY,TRADER,PARTY|CCY,PARTY|TRADER,SOURCE|PARTY,SOURCE|PARTY|CCY,SOURCE|TRADER|CCY,TRADER|CCY,SOURCEDATA,PARTY|SOURCEDATA,TRADER|SOURCEDATA,SOURCEDATA|PARTY|CCY,SOURCEDATA|TRADER|CCY



Calypsos provides a mechanism to map Counterparty using Mapping Window. It also provides multiple mapping of Calypso Entities to TRTN Parties using the Mapping window with the following 6 criteria. These criteria is used in combination to identify the counterparty in Calypso.

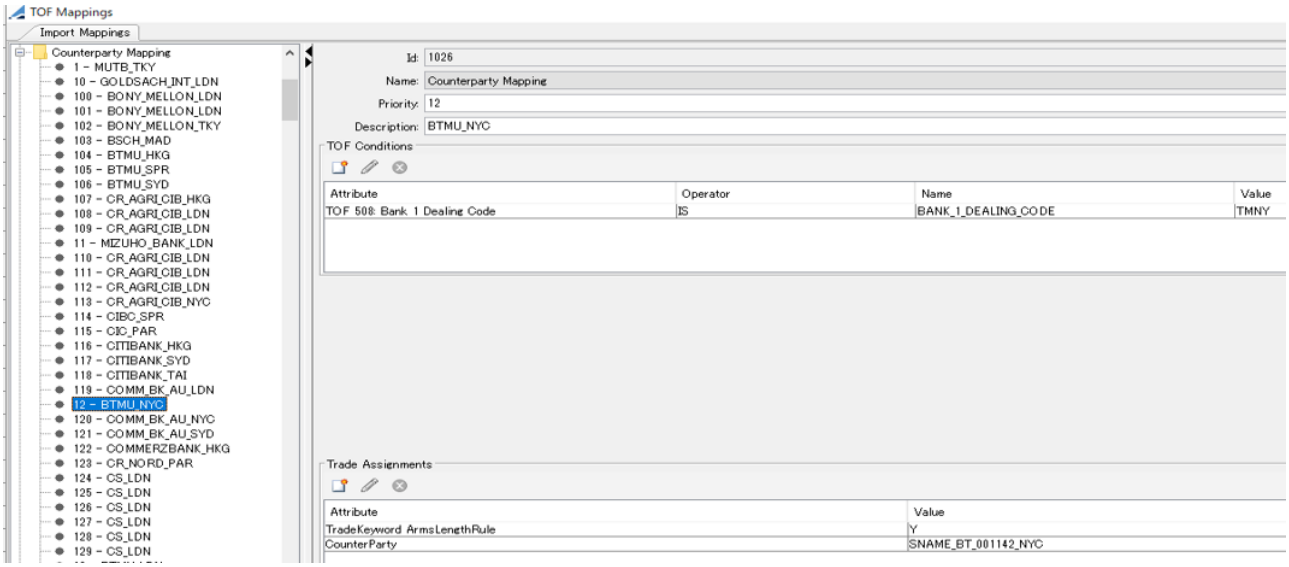
Criteria	Description
PARTY	It is the party information coming from the Platform in the fix file. Fix value: 448 when 452=17 or 452=1
CCY	It is the currency information coming in the fix file. Fix value: Tag 15
CCYPAIR	It is the currency pair information present in the incoming fix file. Fix value : Tag 55
TRADER	It is the trader information present in the fix file. Fix value : 448 when 452 = 36 or 452 = 11
SOURCE	It is the source is the platform from which we are receiving the message. Example: FIX
SOURCEDATA	This is the value of tag 11001

The multiple combinations should be separated by coma and defined using any separator. The same separator that we use in mapping config needs to be provided in TRTNParticipant for incoming. In case of multiple combinations of criteria, then priority is given to the order in which the combinations are defined in the MappingConfig.



If the value of tag 448 having 452=17 or 452=1 is MTBQ and since PARTY mapping is present in the MappingConfig | Participant, then the Counterparty would be MUTB_TKY.

In the below case of counterparty mapping,



TOF Interpretation

When 508 = TMNY, then

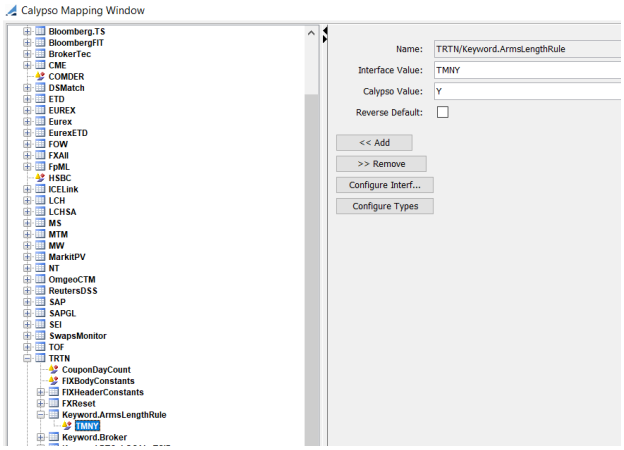
1. Counterparty = SNAME_BT_081142_NYC
2. Map TradeKeyword ArmsLengthRule = Y

TRTN Mapping

1. Counterparty mapping can be done as described earlier.
2. If the user wants to map ArmsLengthRule keyword for a specific Counterparty, then below mapping is required.

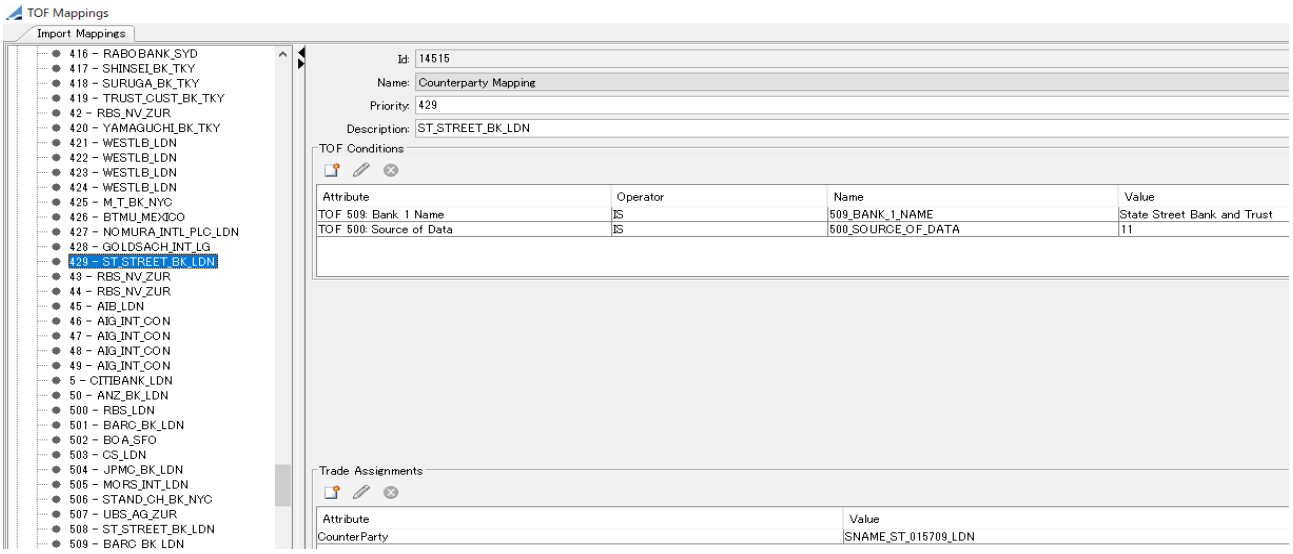
   Interface Value = TMNY (PARTY criteria in this case)

   Calypso Value = Y



Here, let's say for ST_STREET_BK_LDN, relationship between TOF field and FIX tag are as follows:

TOF field	FIX tag
509: Bank 1 Name	523 when 803=5 && 452=17 (OBO trade) Or 803=5 && 452=1 (normal trade)



TOF Interpretation

When 509 = State Street Bank and Trust and 500=11, then Counterparty = SNAME_ST_015709_LDN

TRTN Mapping

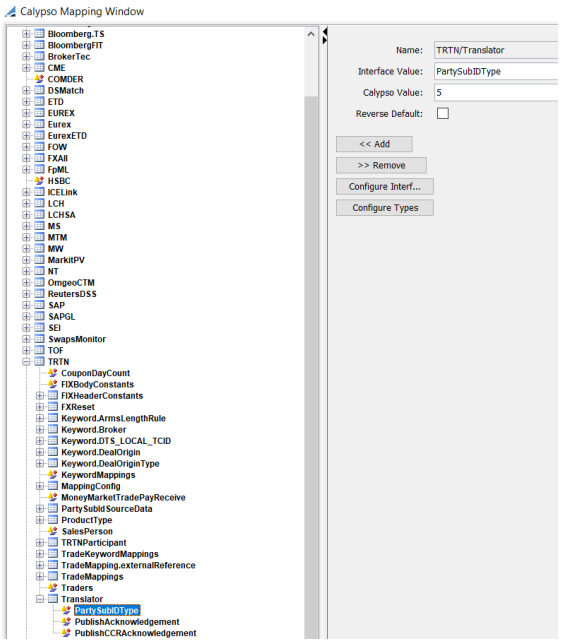
When 11001=11, then get the value of tag 523 when 803=5 and (452=17 or 452=1)

2 levels of mappings need to be done to handle this scenario.

1st Mapping:

Interface Value = PartySulDType

Calypso Value = 5 (This is the value of FIX tag 803)



2nd Mapping:

User needs to have the combination of SOURCEDATA,PARTY|SOURCEDATA,TRADER|SOURCEDATA,SOURCEDATA|PARTY|CCY,SOURCEDATA|TRADER|CCY in MappingConfig as in the 1st Approach Mapping Criteria.

Interface Value : State Street Bank and Trust|11

Calypso Value : SNAME_ST_015709_LDN

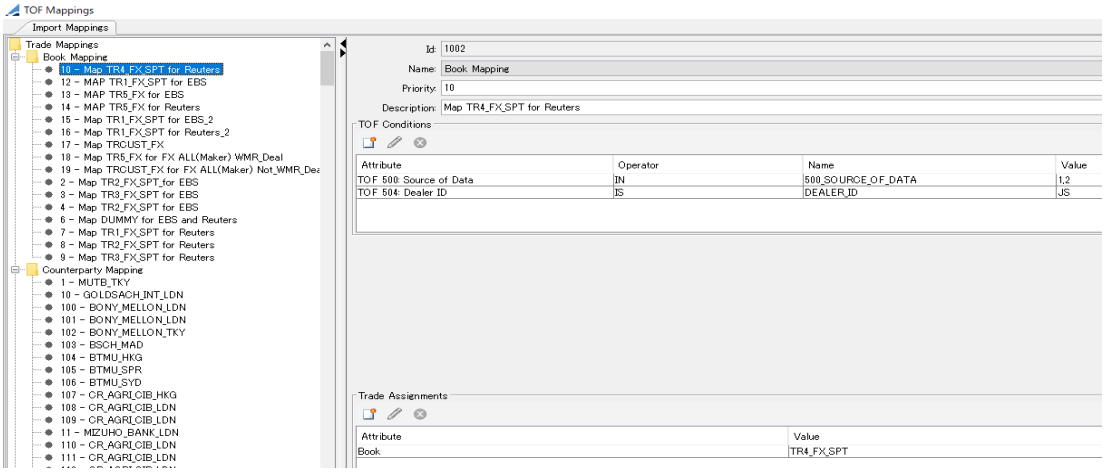
3.9.2 Book Mapping

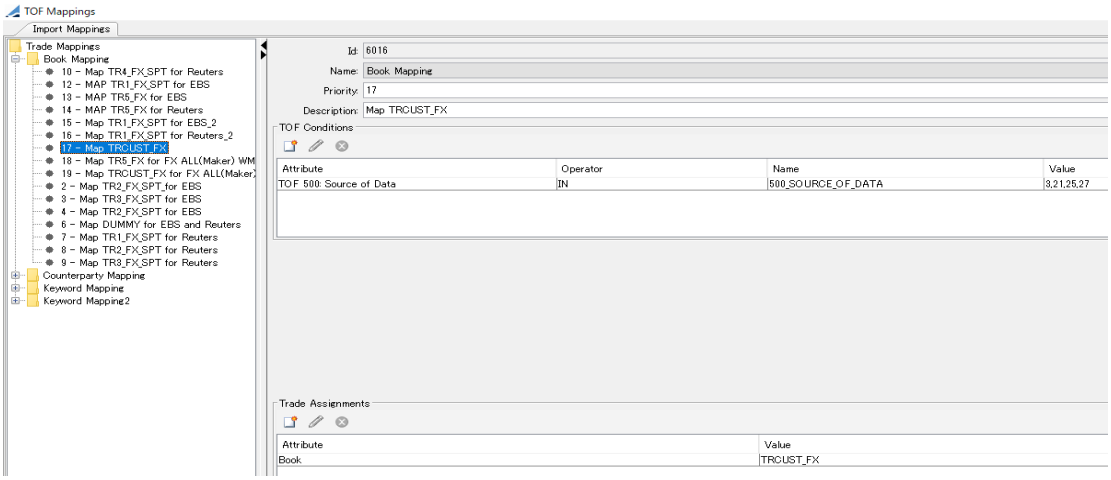
If (TOF 500: Source of Data is In (1,2) and TOF 504:Dealer ID is JS) map this TOF message to Calypso Book TR4_FX_SPT. If (Tag 11001 is IN (1,2) and (Tag 448 448 when 452=36 (entering trader) Or 452=11 (executing trader)) is JS) map this FIX message to Calypso Book TR4_FX_SPT.

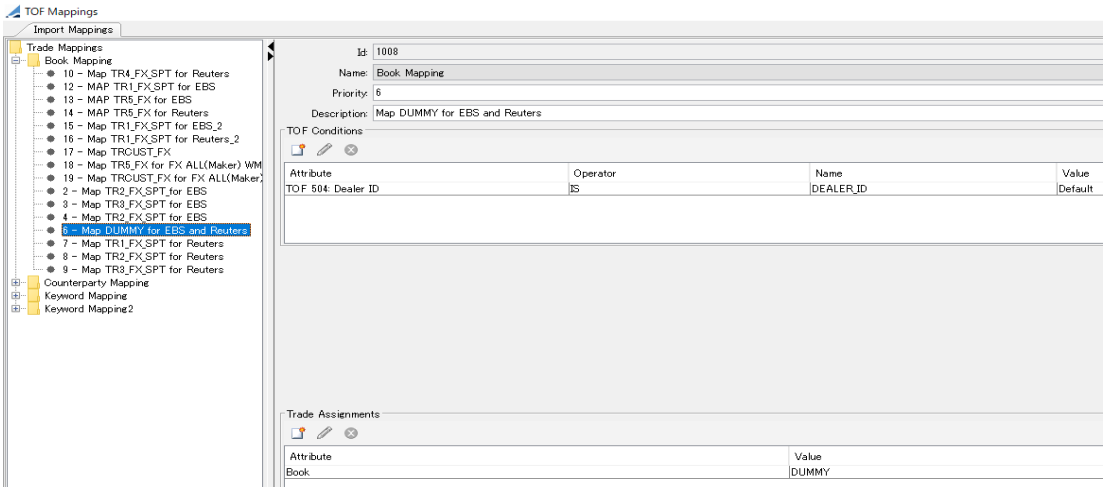
Relationship between TOF field and FIX tag:

TOF field	FIX tag
500: Source of Data	11001
504: Dealer ID	448 when 452=36 (entering trader) Or 452=11 (executing trader)

In the following cases of book mapping,







TOF Interpretation

When 500 = 1 or 2 and 504 = JS, then Book = TR4_FX_SPT

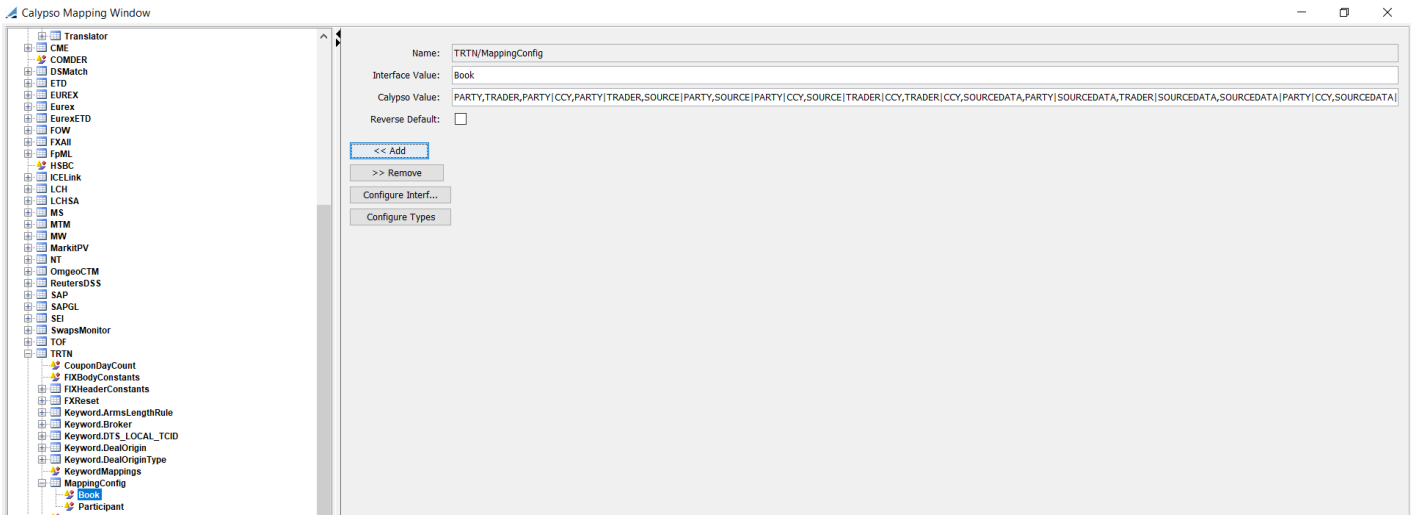
TRTN Mapping

When 11001 = 1 or 2, then get the value of FIX tag 448 having 452=36 or 452=11. This can be achieved via MappingCriteria.

MappingCriteria

User needs to define a MappingConfig:

Interface Value = Book
Calypso Value =
PARTY,TRADER,PARTY|CCY,PARTY|TRADER,SOURCE|PARTY,SOURCE|PARTY|CCY,SOURCE|TRADER|CCY,
TRADER|CCY,SOURCEDATA,PARTY|SOURCEDATA,TRADER|SOURCEDATA,SOURCEDATA|PARTY|CCY,SOU
RCEDATA|TRADER|CCY



Calypsos provided a mechanism to map Book using Calypso Mapping Window. It also provides multiple mapping of Calypso Entities to TRTN Parties using the Mapping window with the following 6 criteria. These criteria are used in combination to identify the Book in Calypso.

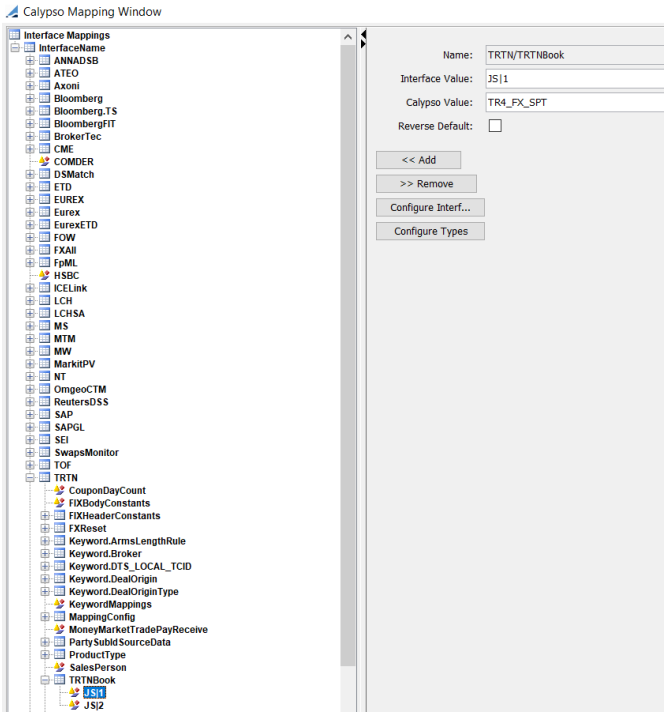
Criteria	Description
PARTY	It is the party information coming from the Platform in the fix file. Fix value : 448 when 452=3
CCY	It is the currency information coming in the fix file. Fix value : Tag 15
CCYPAIR	It is the currency pair information present in the incoming fix file. Fix value : Tag 55
TRADER	It is the trader information present in the fix file Fix value when 452=36 or 452=11
SOURCE	It is the source is the platform from which we are receiving the message. Example: FIX
SOURCEDATA	This is the value of tag 11001

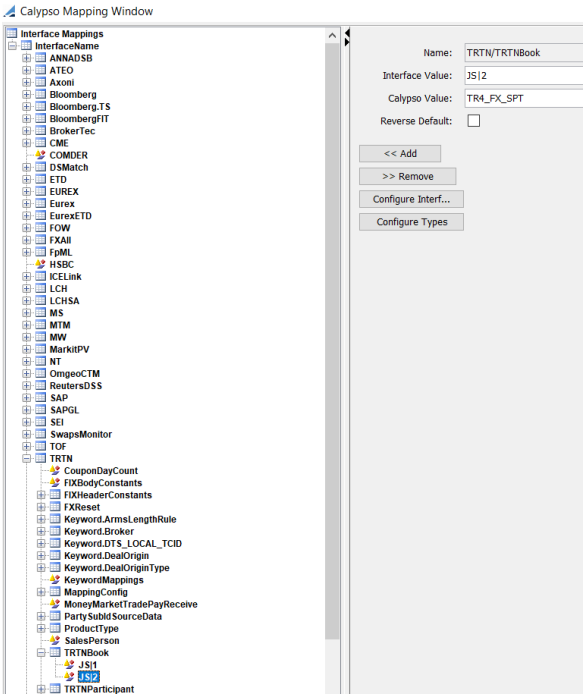
The multiple combinations should be separated by coma and defined using any separator. The same separator that we use in mapping config needs to be provided in TRTNBook for incoming. In case of multiple combinations of criteria, then priority is given to the order in which the combinations are defined in the MappingConfig.

Since in this case we have 2 conditions, i.e. 11001 = 1 , then get the value of FIX tag 448. So we need to have PARTY|SOURCEDATA mapping in MappingConfig for Book as shown above.

Interface Value : JS|1 (This is the PARTY|SOURCEDATA mapping)

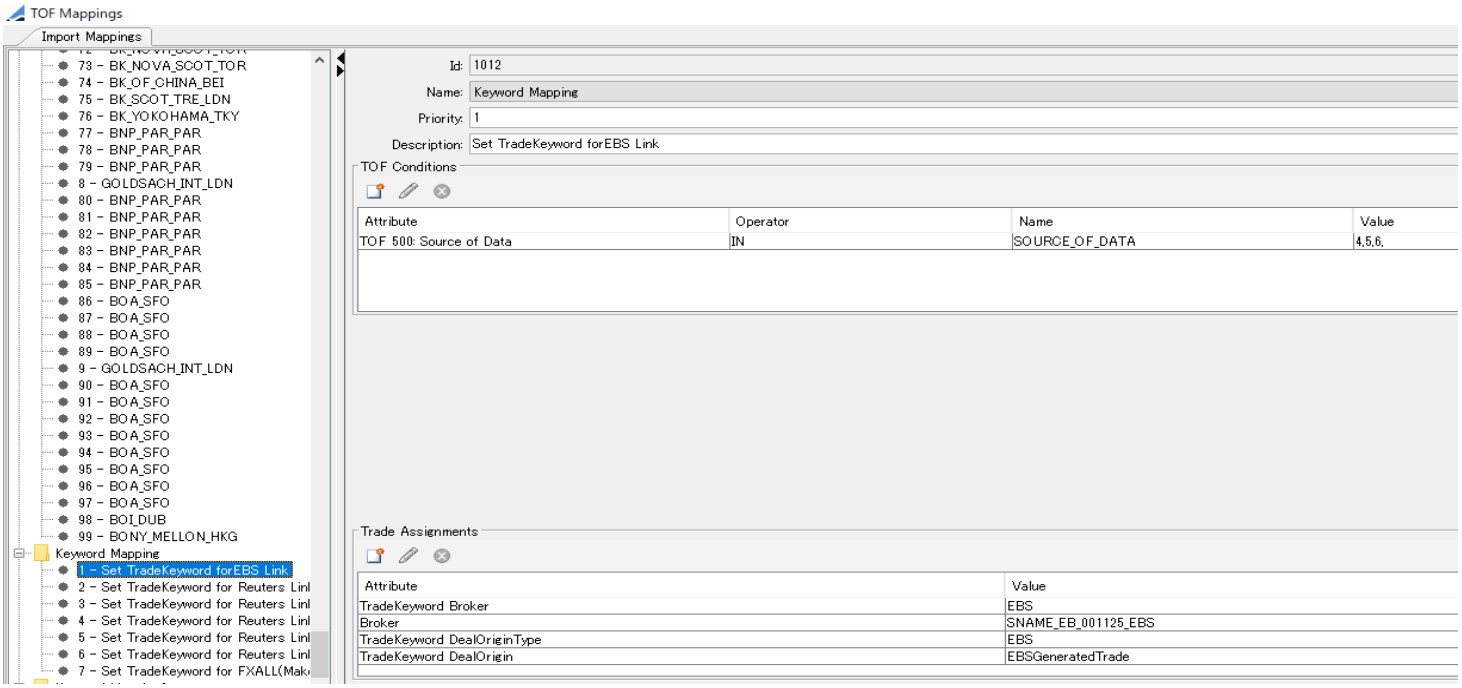
Calypso Value : TR4_FX_SPT


Calypso Mapping Window
Interface Mappings
InterfaceName
ANNADSB
ATEO
Axoni
Bloomberg
Bloomberg.TS
BloombergFIT
BrokerTec
CME
COMDER
DSMatch
ETD
EUREX
Eurex
EurexETD
FOW
FXAll
FpML
HSBC
ICELink
LCH
LCHSA
MS
MTM
MW
MarkitPV
NT
OmgeoCTM
ReutersDSS
SAP
SAPGL
SEI
SwapsMonitor
TOF
TRTN
CouponDayCount
FIXBodyConstants
FIXHeaderConstants
FXReset
Keyword.ArmsLengthRule
Keyword.Broker
Keyword.DTS_LOCAL_TCID
Keyword.DealOrigin
Keyword.DealOriginType
KeywordMappings
MappingConfig
MoneyMarketTradePayReceive
PartySubIdSourceData
ProductType
SalesPerson
TRTNBook
JS|1
JS|2
Name:
TRTN/TRTNBook
Interface Value:
JS|1
Calypso Value:
TR4_FX_SPT
Reverse Default:
<< Add
>> Remove
Configure Interf...
Configure Types

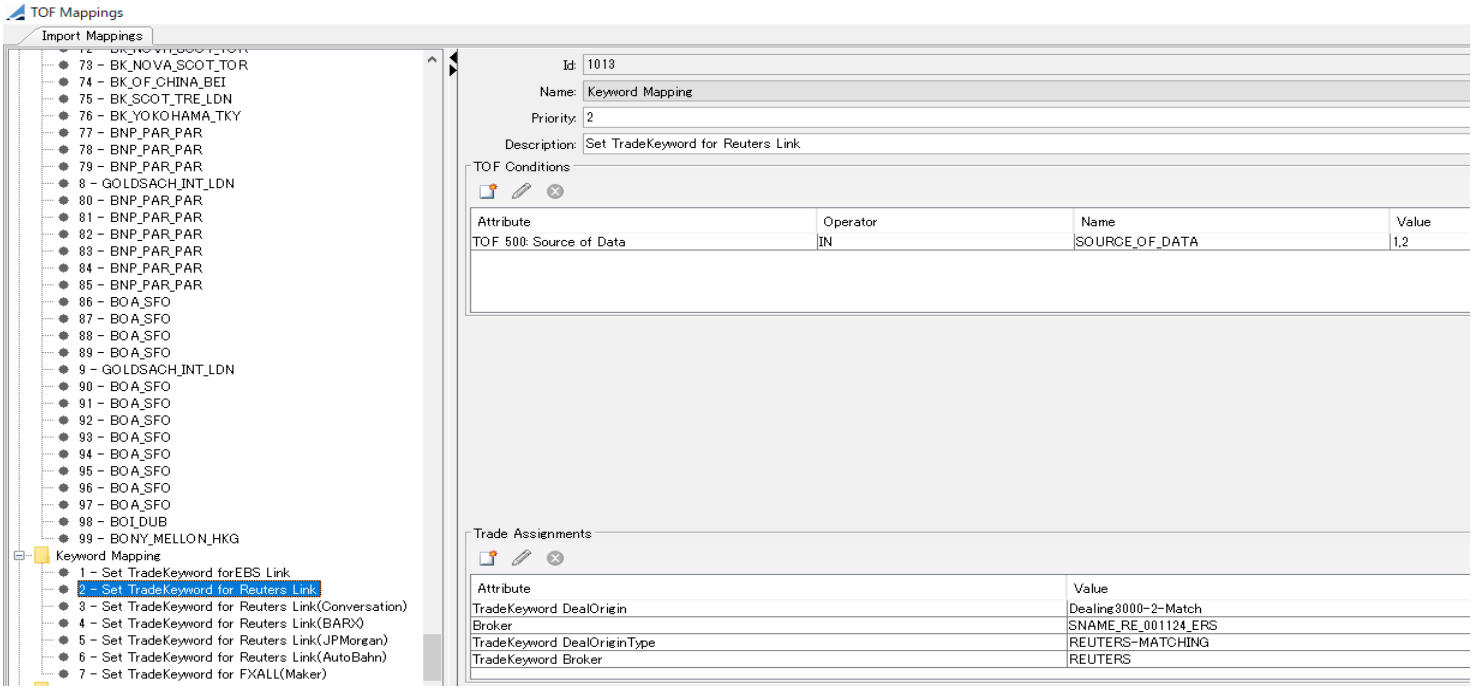

Calypso Mapping Window
Interface Mappings
InterfaceName
ANNADSB
ATEO
Axoni
Bloomberg
Bloomberg.TS
BloombergFIT
BrokerTec
CME
COMDER
DSMatch
ETD
EUREX
Eurex
EurexETD
FOW
FXAll
FpML
HSBC
ICELink
LCH
LCHSA
MS
MTM
MW
MarkitPV
NT
OmgeoCTM
ReutersDSS
SAP
SAPGL
SEI
SwapsMonitor
TOF
TRTN
CouponDayCount
FIXBodyConstants
FIXHeaderConstants
FXReset
Keyword.ArmsLengthRule
Keyword.Broker
Keyword.DTS_LOCAL_TCID
Keyword.DealOrigin
Keyword.DealOriginType
KeywordMappings
MappingConfig
MoneyMarketTradePayReceive
PartySubIdSourceData
ProductType
SalesPerson
TRTNBook
JS|1
JS|2
TRTNParticipant
Name:
TRTN/TRTNBook
Interface Value:
JS|2
Calypso Value:
TR4_FX_SPT
Reverse Default:
<< Add
>> Remove
Configure Interf...
Configure Types

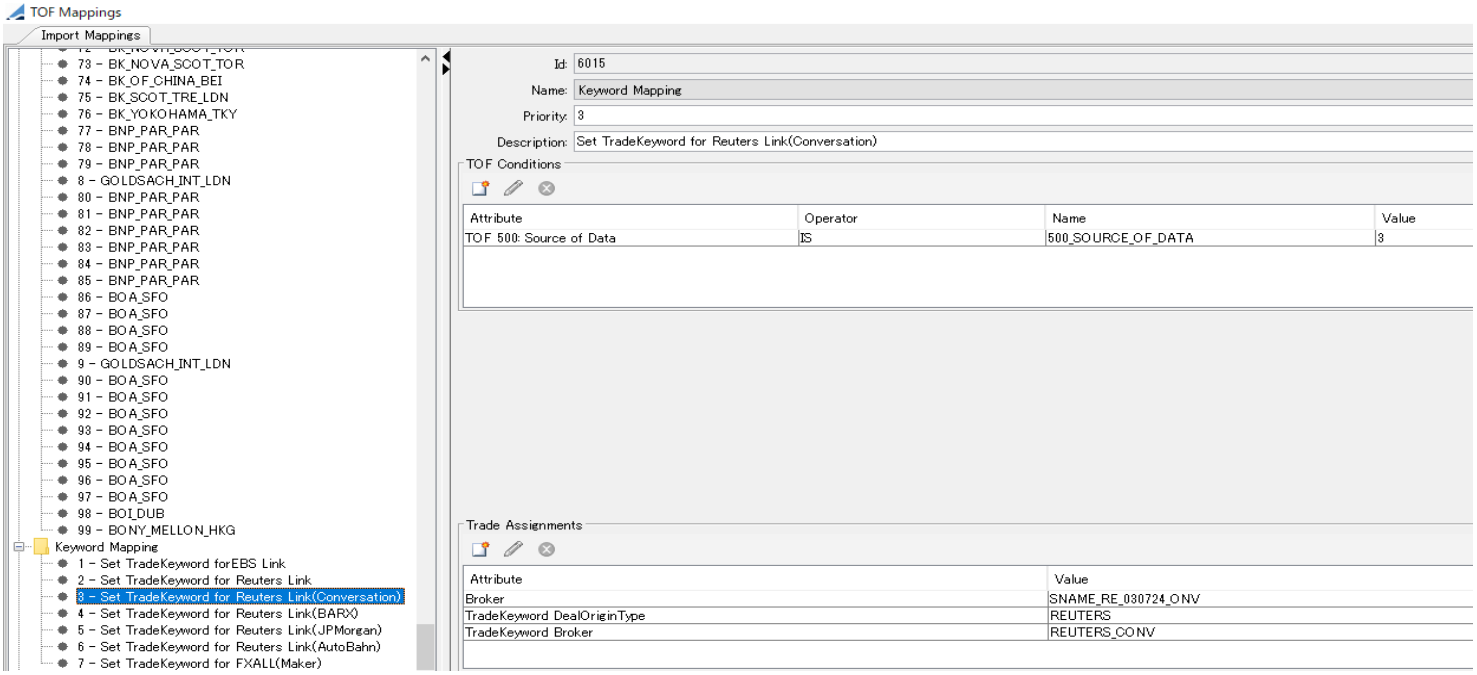
3.9.3 Trade Keywords Mapping

In the following cases of book mapping,



Broker has same value as TradeKeyword Broker in Calypso. Only need to consider Tradekeyword Broker in RTN interface.





TOF Interpretation

When 500=4 or 5 or 6, then map the following keywords :

- TradeKeyword.Broker → EBS
- TradeKeyword.DealOriginType → EBS
- TradeKeyword.DealOrigin → EBSGeneratedTrade

TRTN Mapping

Equivalent of TOF tag 500 is 11001. When 11001=4 or 5 or 6, then map the TradeKeywords

- Broker
- DealOriginType
- DealOrigin

This should be done using following steps:

To map static values, user will need to do 2 level of mappings:

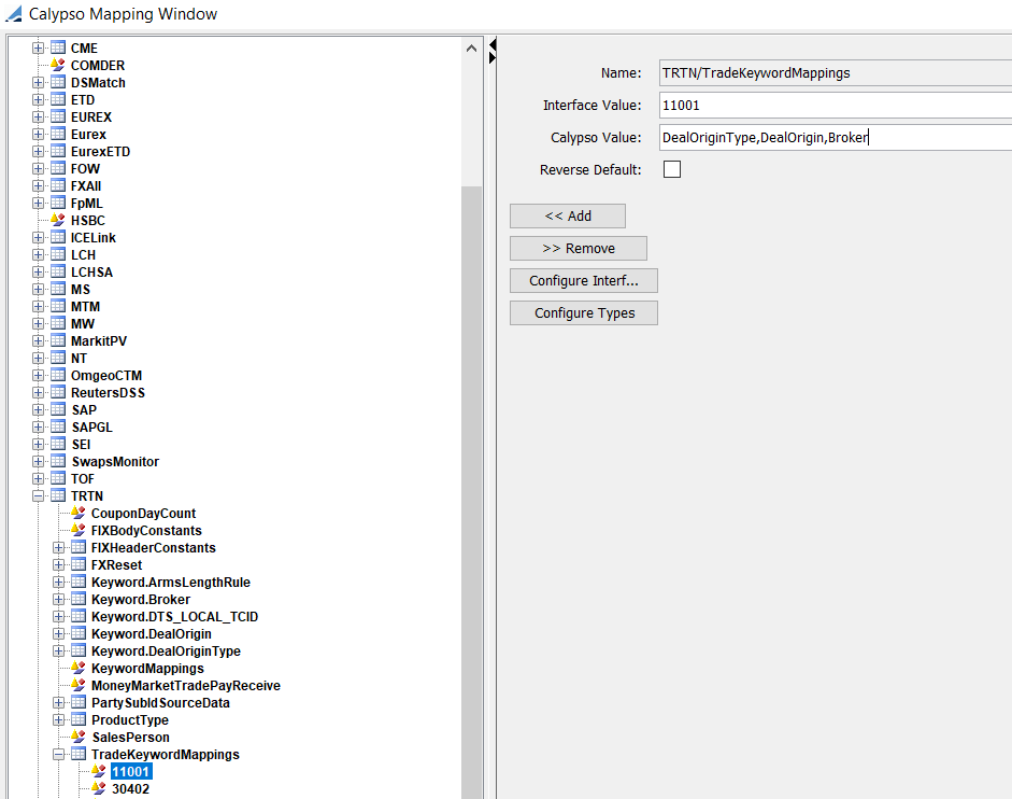
1. Map tag 11001 with the keyword names with which they want in TradeKeywordMappings.
2. Define a new Interface type Keyword.<KeywordName> e.g. Keyword.DealOriginType. When the value of 11001 is 4, then map Keyword.DealOriginType to value : “EBS”.

Step 1: 1st Level of mapping:

Interface Type: TradeKeywordMappings

Interface Value: 11001

Calypso Value: Broker, DealOriginType, DealOrigin



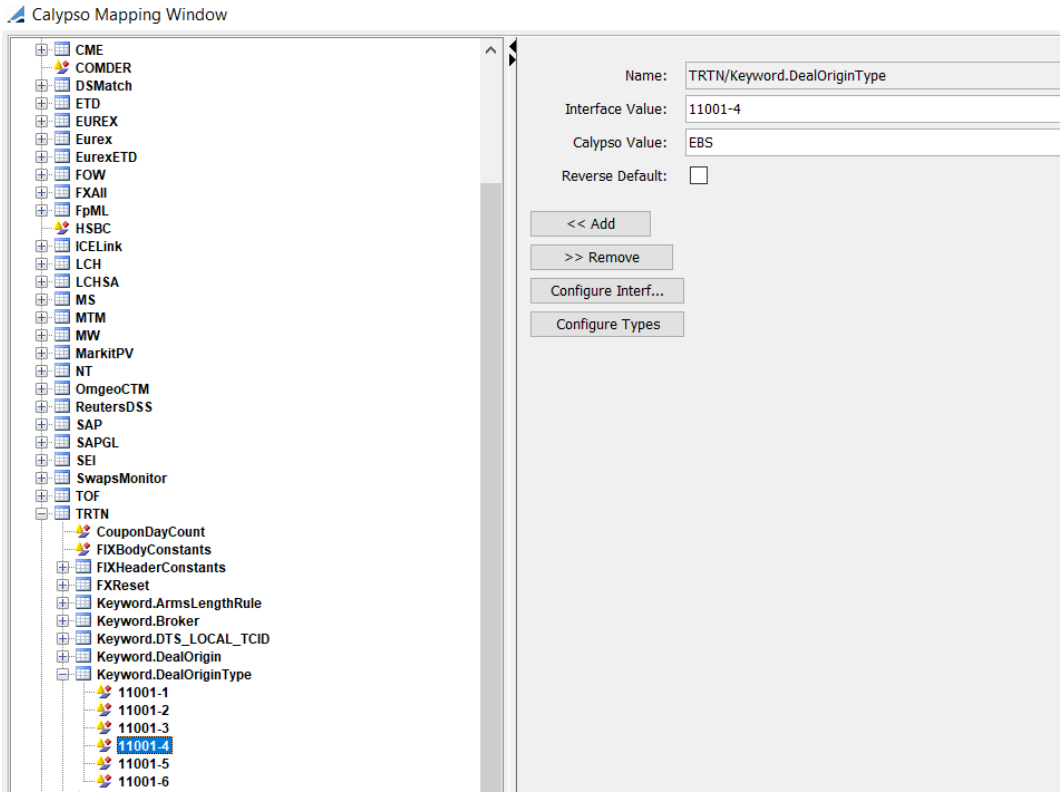
Step 2: 2nd Level of mapping:

Interface Type: Keyword.DealOriginType

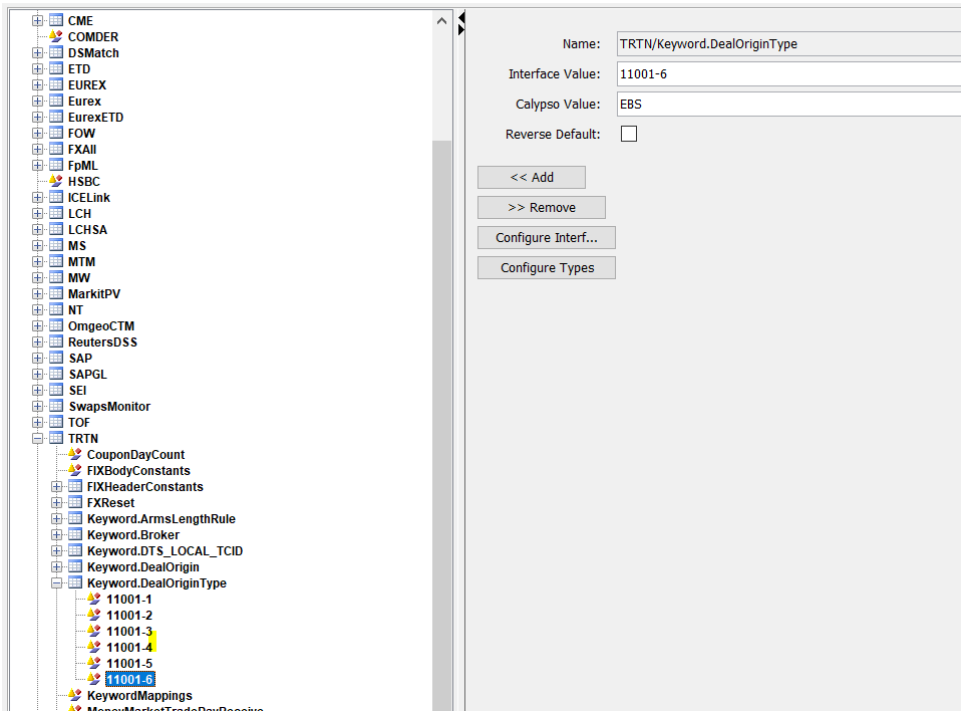
Interface Value: 11001-4

Calypso Value: EBS

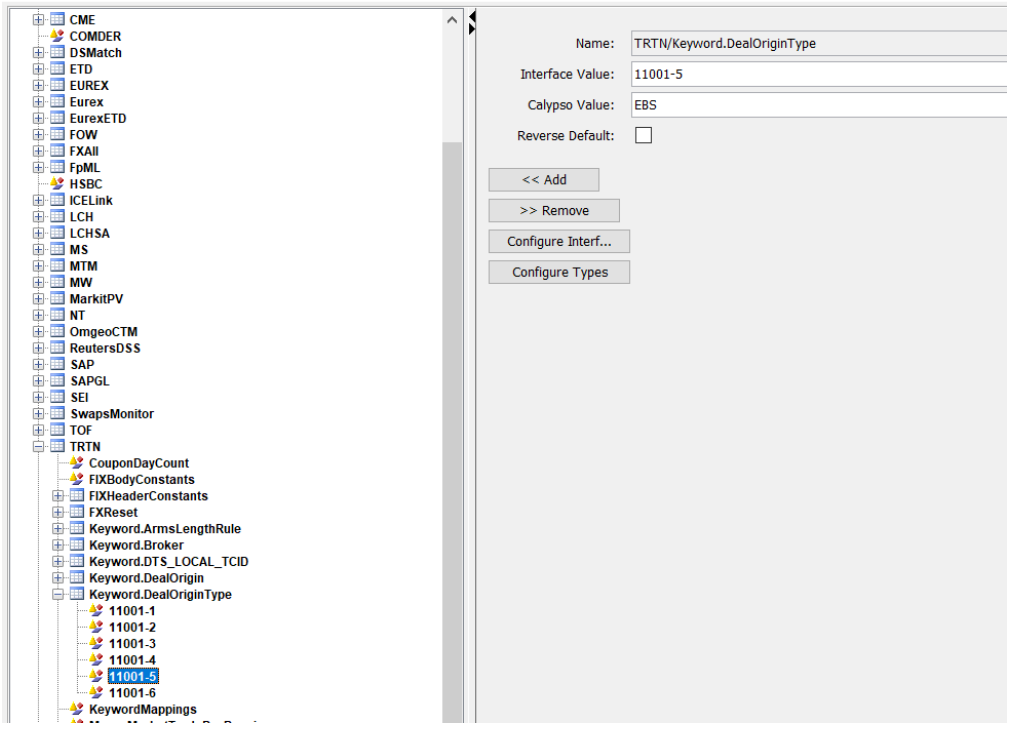
Keyword.DealOriginType


Calypso Mapping Window
CME
COMDER
DSMatch
ETD
EUREX
Eurex
EurexETD
FOW
FXAll
FpML
HSBC
ICELink
LCH
LCHSA
MS
MTM
MW
MarkitPV
NT
OmgeoCTM
ReutersDSS
SAP
SAPGL
SEI
SwapsMonitor
TOF
TRTN
CouponDayCount
FIXBodyConstants
FIXHeaderConstants
FXReset
Keyword.ArmsLengthRule
Keyword.Broker
Keyword.DTS_LOCAL_TCID
Keyword.DealOrigin
Keyword.DealOriginType
11001-1
11001-2
11001-3
11001-4
11001-5
11001-6
Name:
TRTN/Keyword.DealOriginType
Interface Value:
11001-4
Calypso Value:
EBS
Reverse Default:
<< Add
>> Remove
Configure Interf...
Configure Types

Calypso Mapping Window

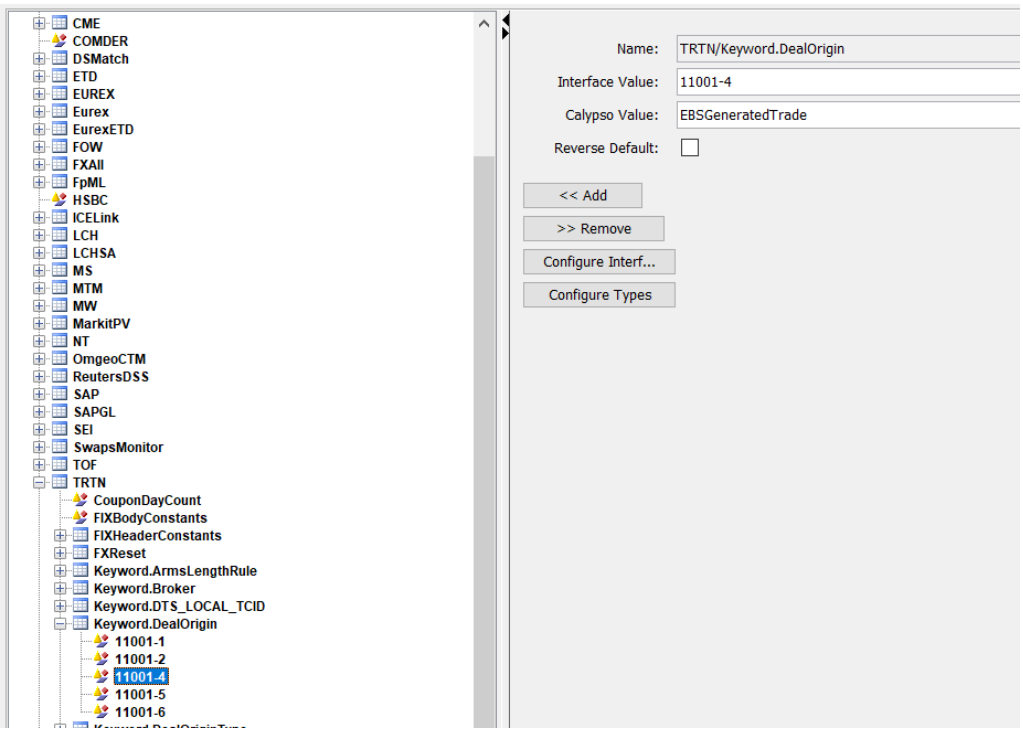


Calypso Mapping Window

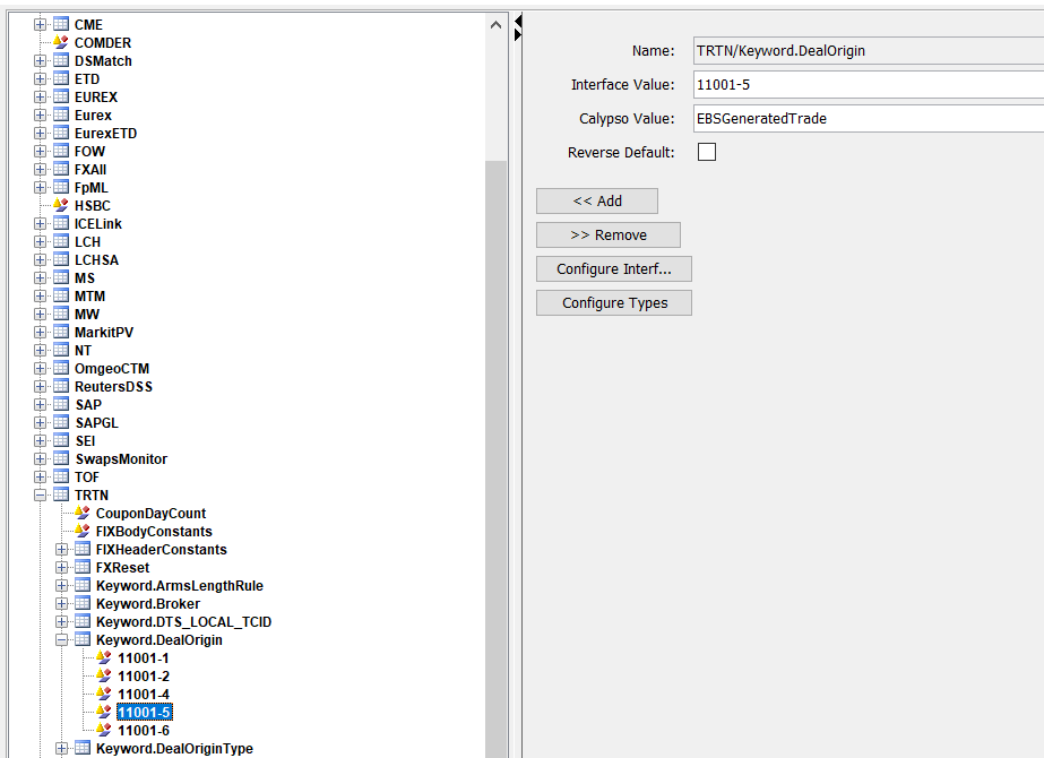


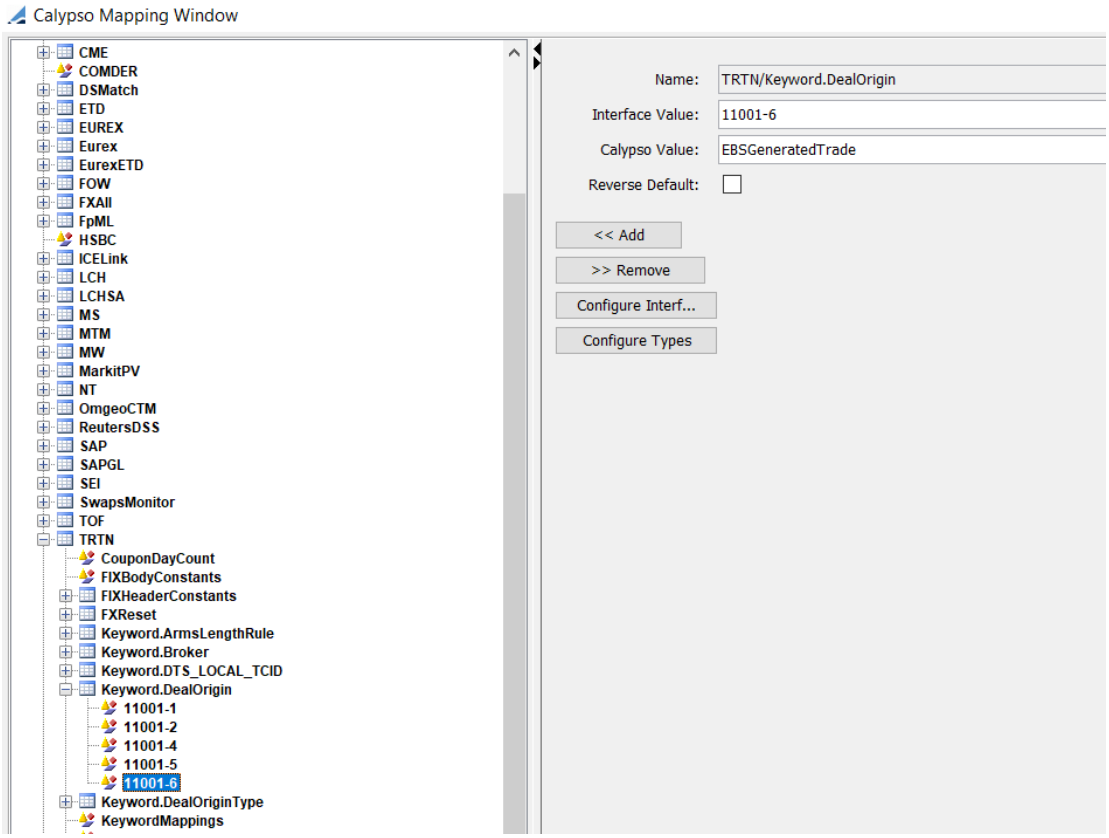
Keyword.DealOrigin

Calypso Mapping Window

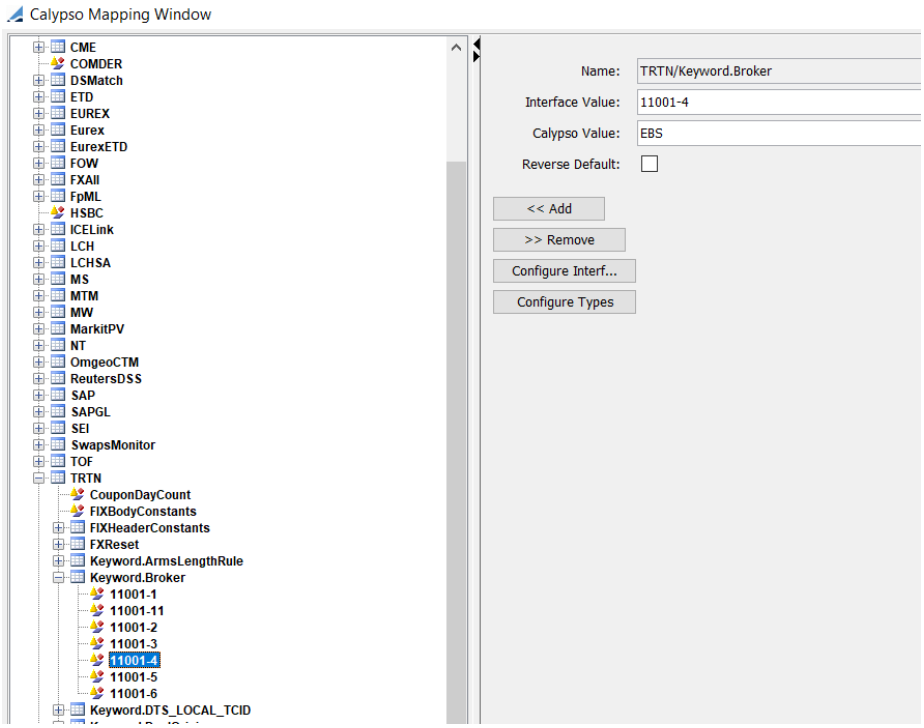


Calypso Mapping Window

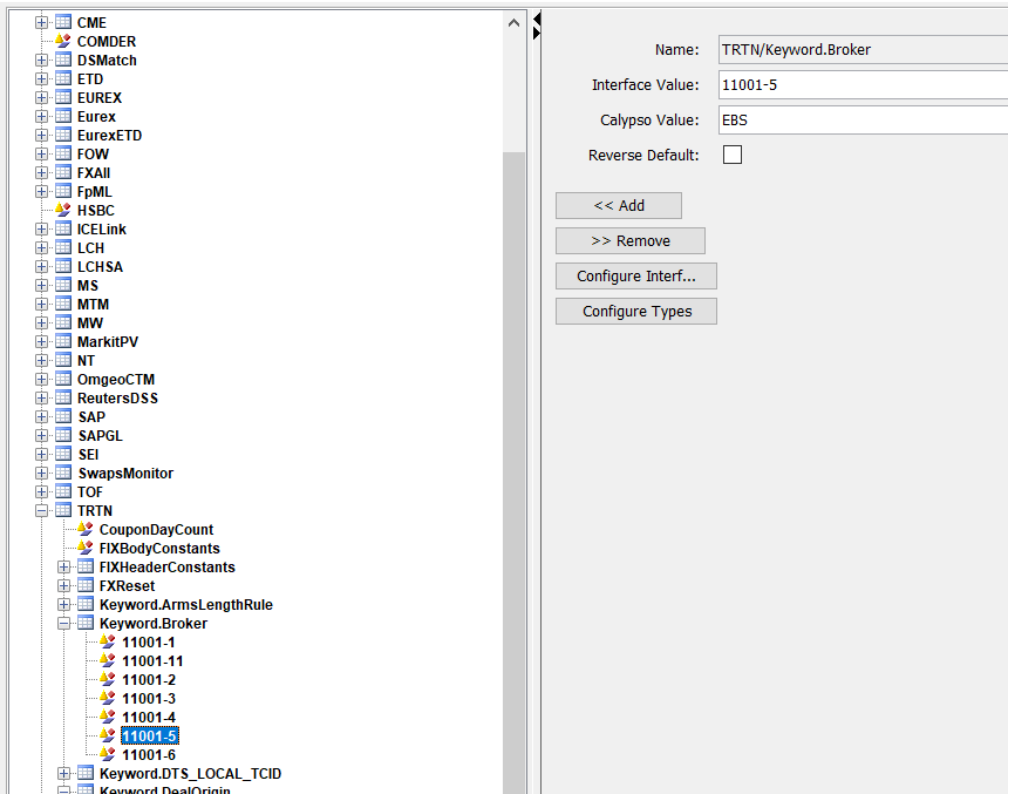




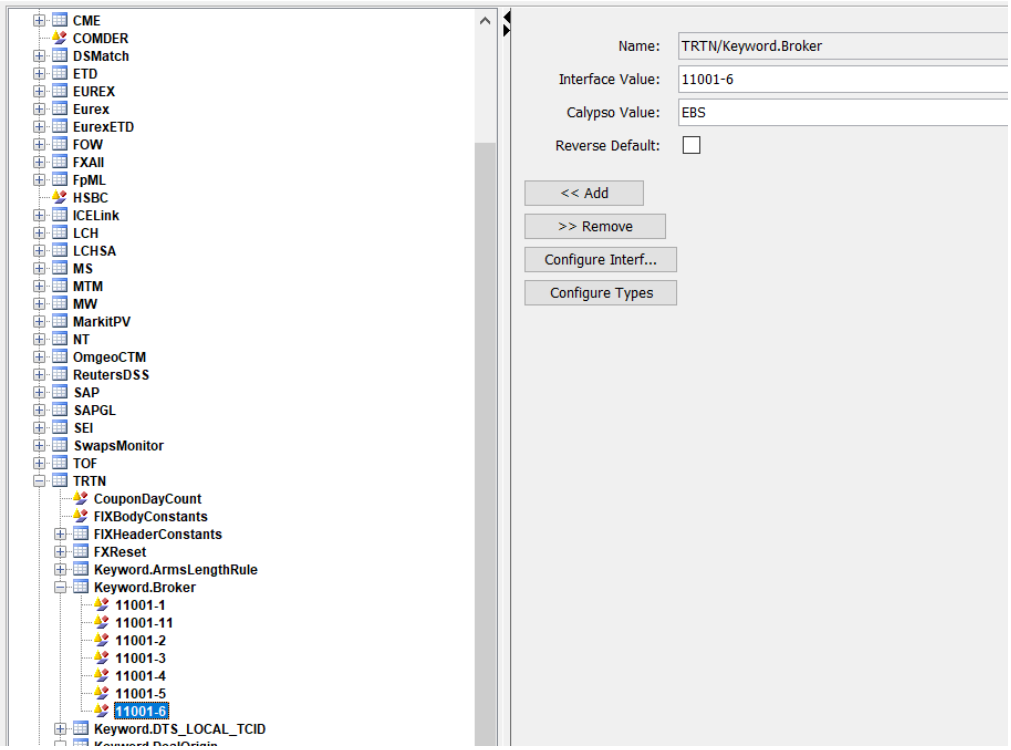
Keyword.Broker:



Calypso Mapping Window

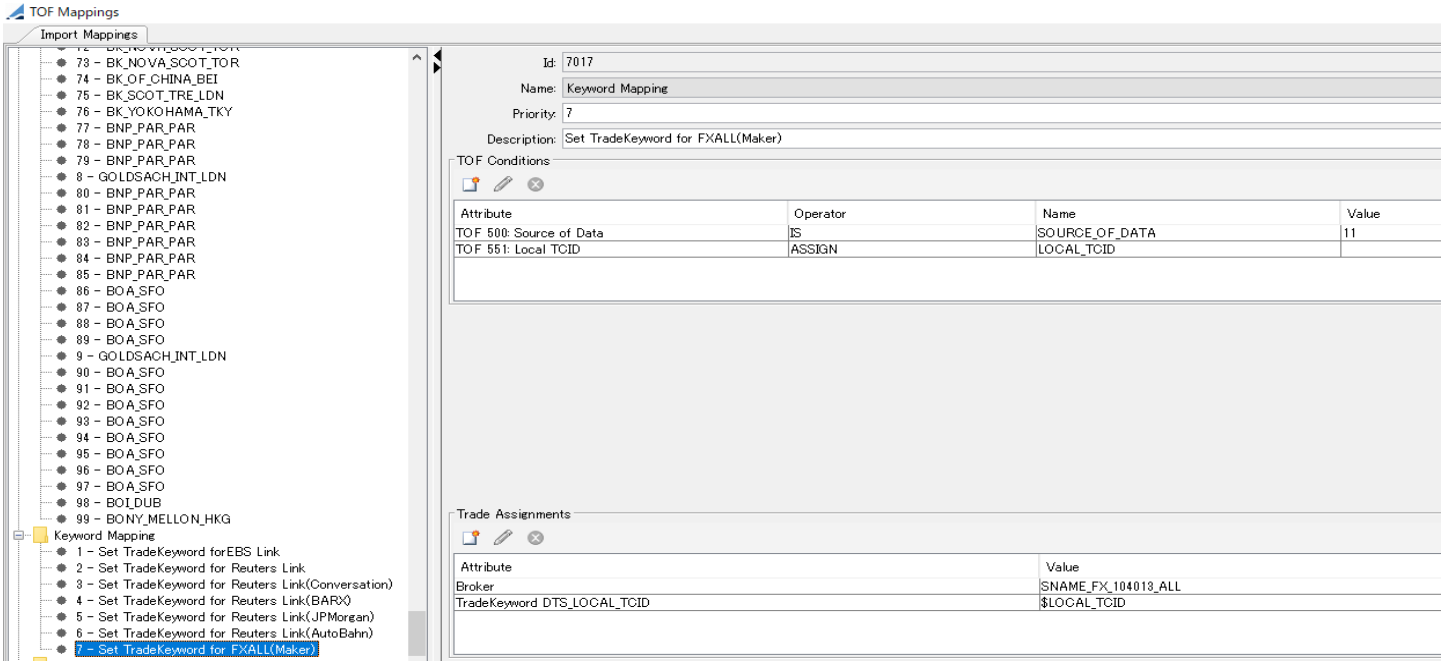


Calypso Mapping Window



In the following case:

TOF field	FIX tag
551: Local TCID	448 when 452=3



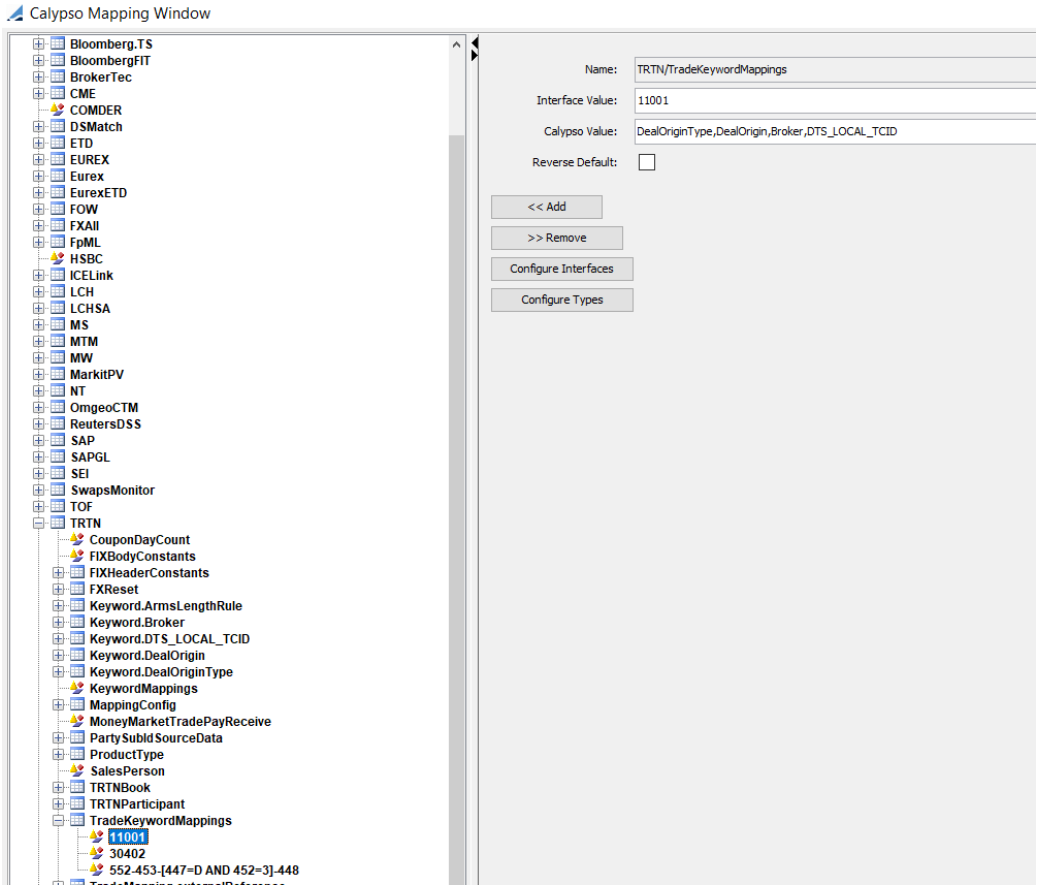
TOF Interpretation

When TOF tag 500=11, then assign the value of Tag 551 to DTS_LOCAL_TCID Trade keyword.

TRTN Mapping

This is a 2 level of mapping

1st Level:

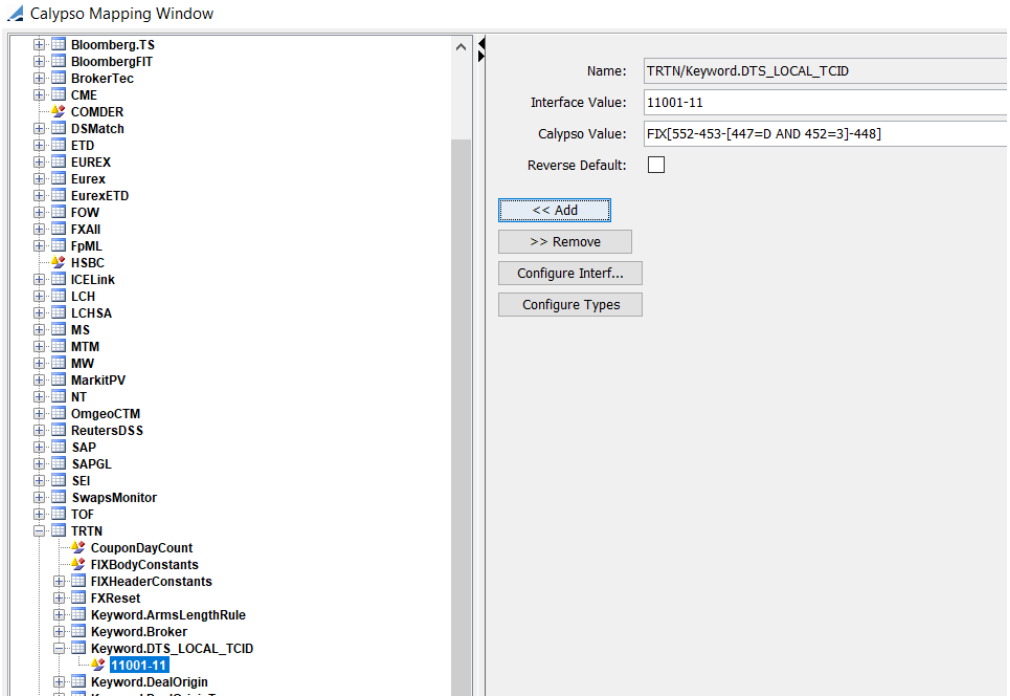


2nd Level:

We need to get the value of 448 tag when 452=3 when 11001=11 (equivalent of TOF tag 500).

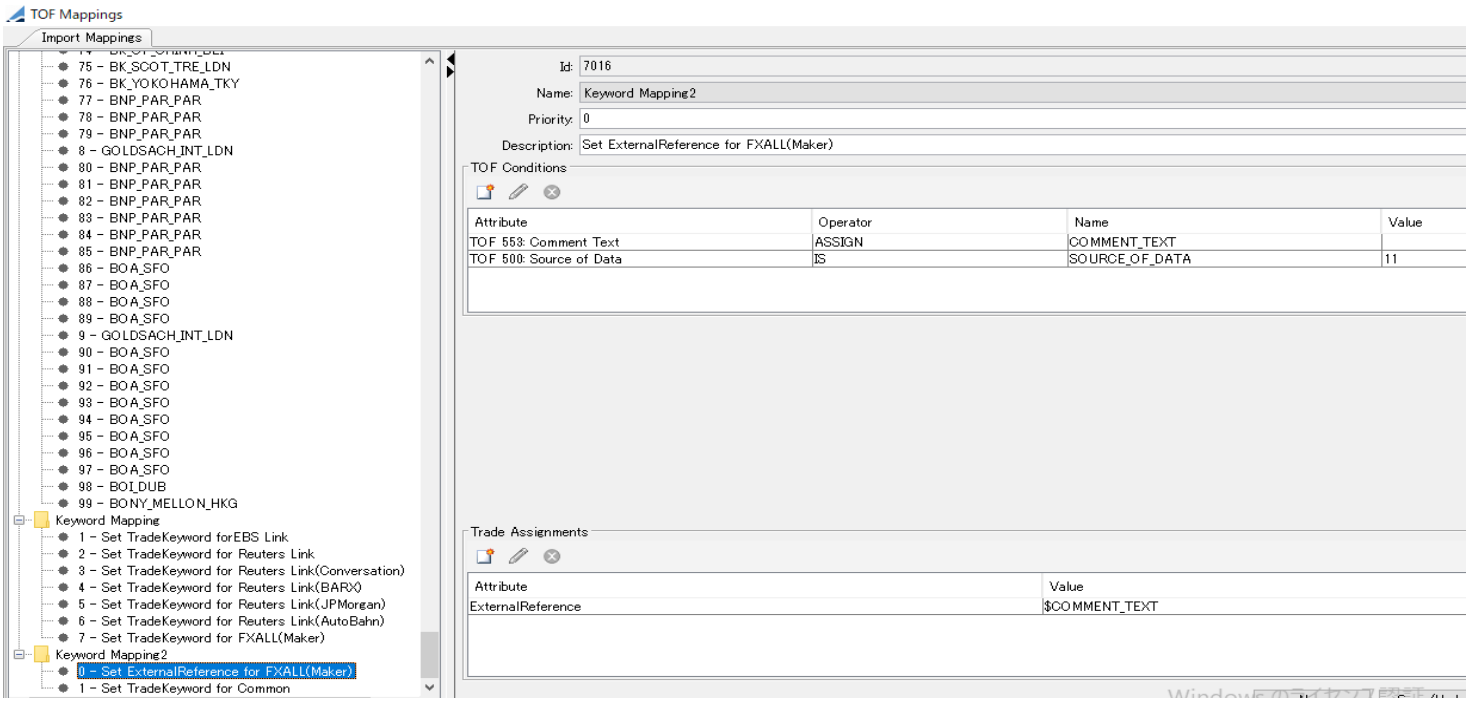
This is a conditional based mapping where the user will have to provide the FIX tag for mapping which means:

1. 552 - is the main group in the FIX message
2. 453 - is the subgroup of 552
3. [447=D and 452=3] - 448 - take the value of 448 tag having 452=3 and 447=D



In the following case:

TOF field	FIX tag
553: Comment Text	11057+11058 (Dealing & Matching)



TOF Interpretation

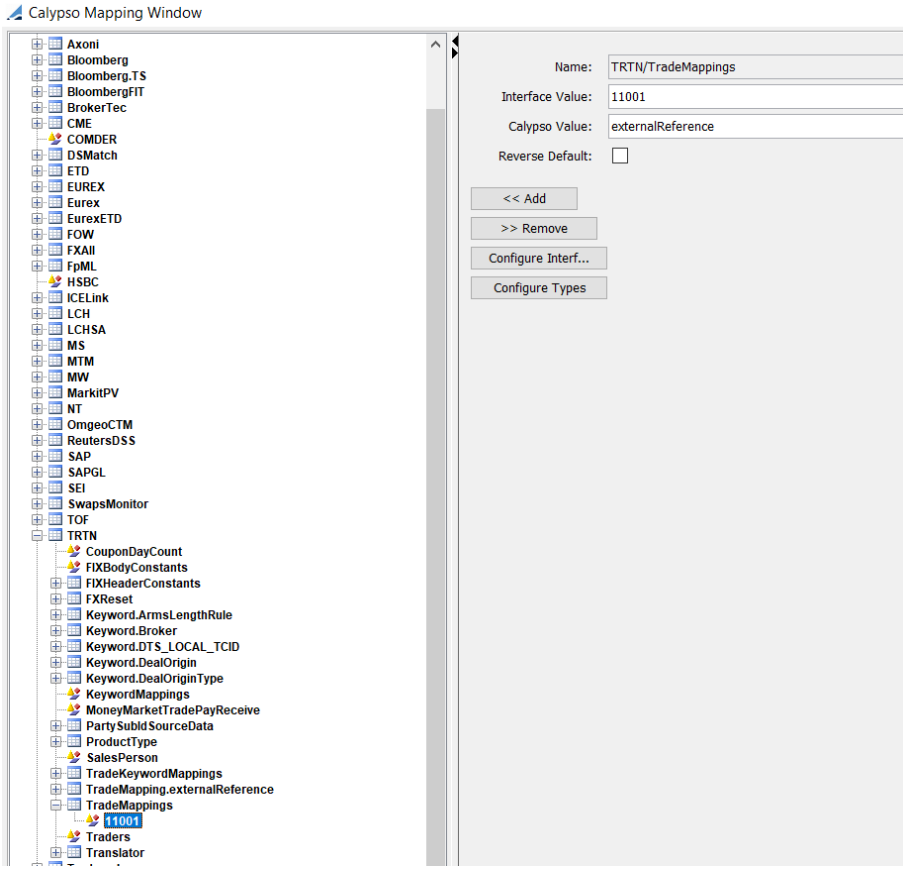
When TOF tag 500=11, then re-map External Reference with value of TOF tag 553.

TRTN mapping

When 11001=11 (TOF equivalent of tag 500), then re-map the External Reference with values of combination of 11057 and 11058 FIX tags.

There are 2 levels of mapping.

1st Level: User will provide the FIX tag number in Interface Value under TradeMappings type and Calypso Value will be the field that needs to be mapped.e.g externalReference.

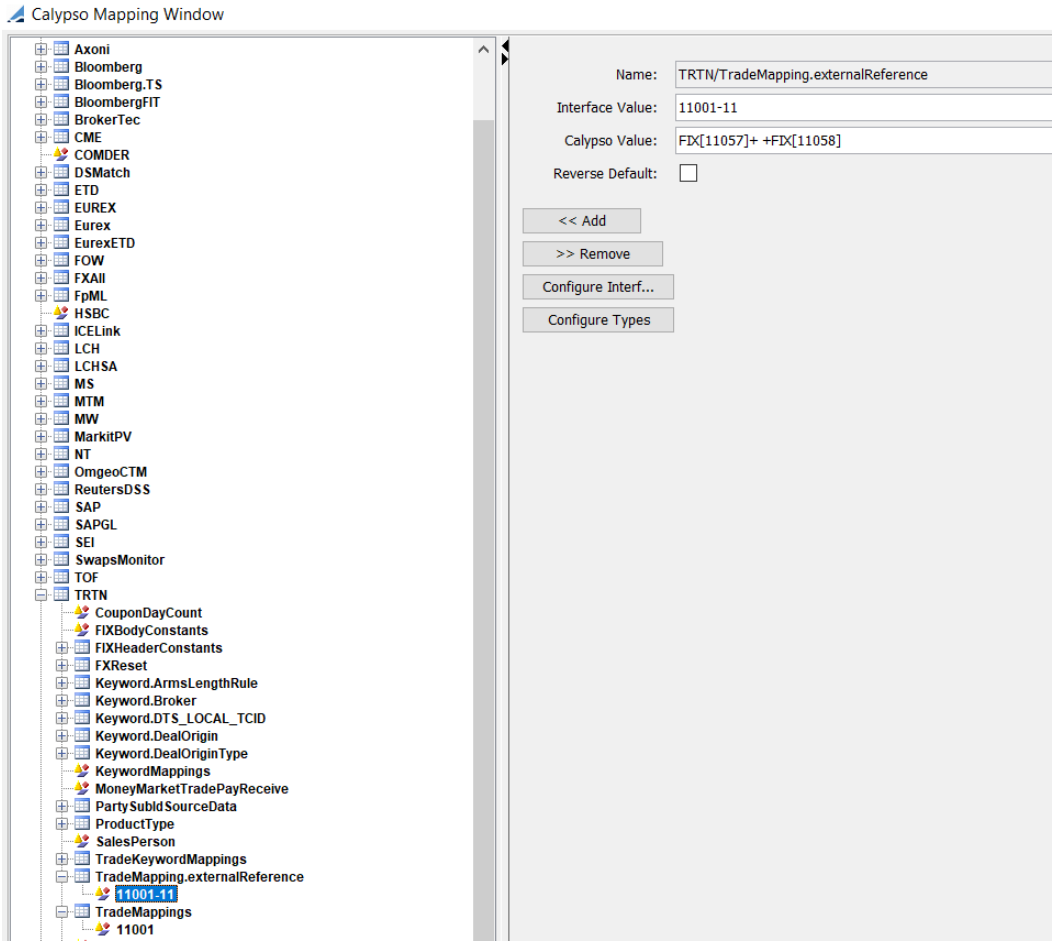


2nd Level Mapping: User needs to create another mapping of Type TradeMapping.<FieldName>.e.g TradeMapping.externalReference.

Interface Value : 11001-11 → This is the FIX tag and its corresponding value

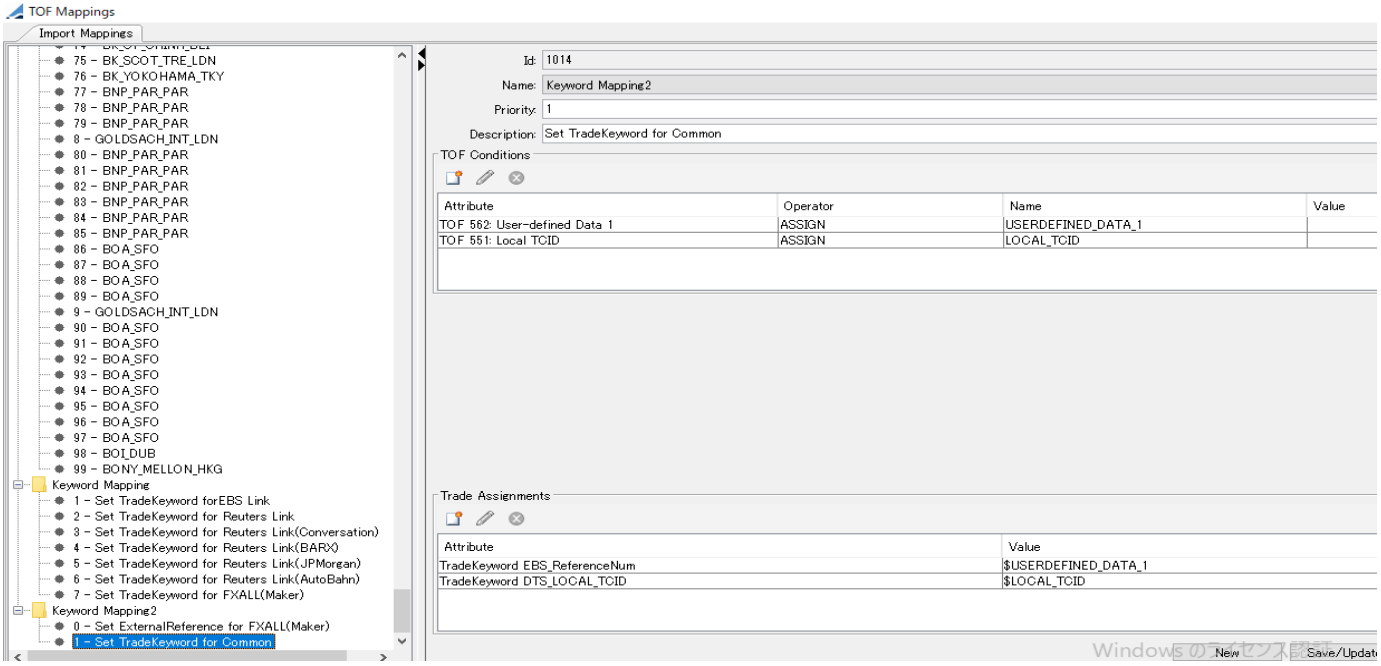
Calypso Value : FIX[11057]+ +FIX[11058] → This the tag number from which external reference needs to be taken.

When 11001 = 11, then re-map the external reference with the values of FIX tags 11057 and 11058



In the following case,

TOF field	FIX tag
562: User-defined Data 1	30402
551: Local TCID	448 when 452=3

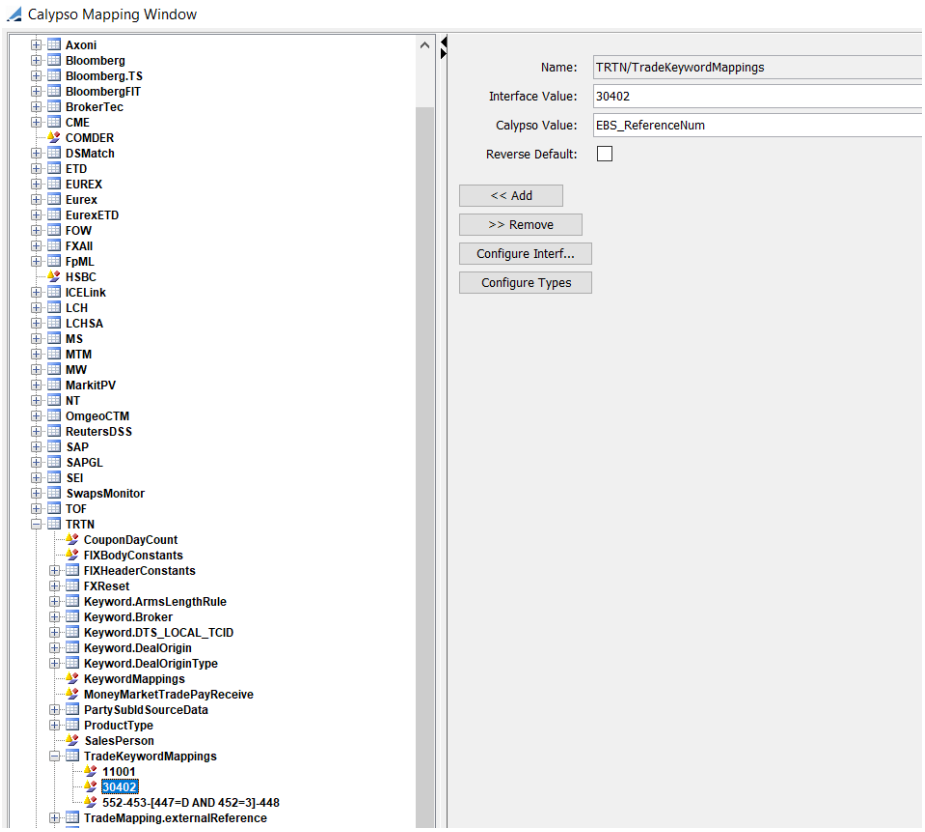


TOF Interpretation

1. Assign the value of TOF tag 562 to TradeKeyword EBS_Reference
2. Assign the value of TOF tag 551 to TradeKeyword DTS_LOCAL_TCID

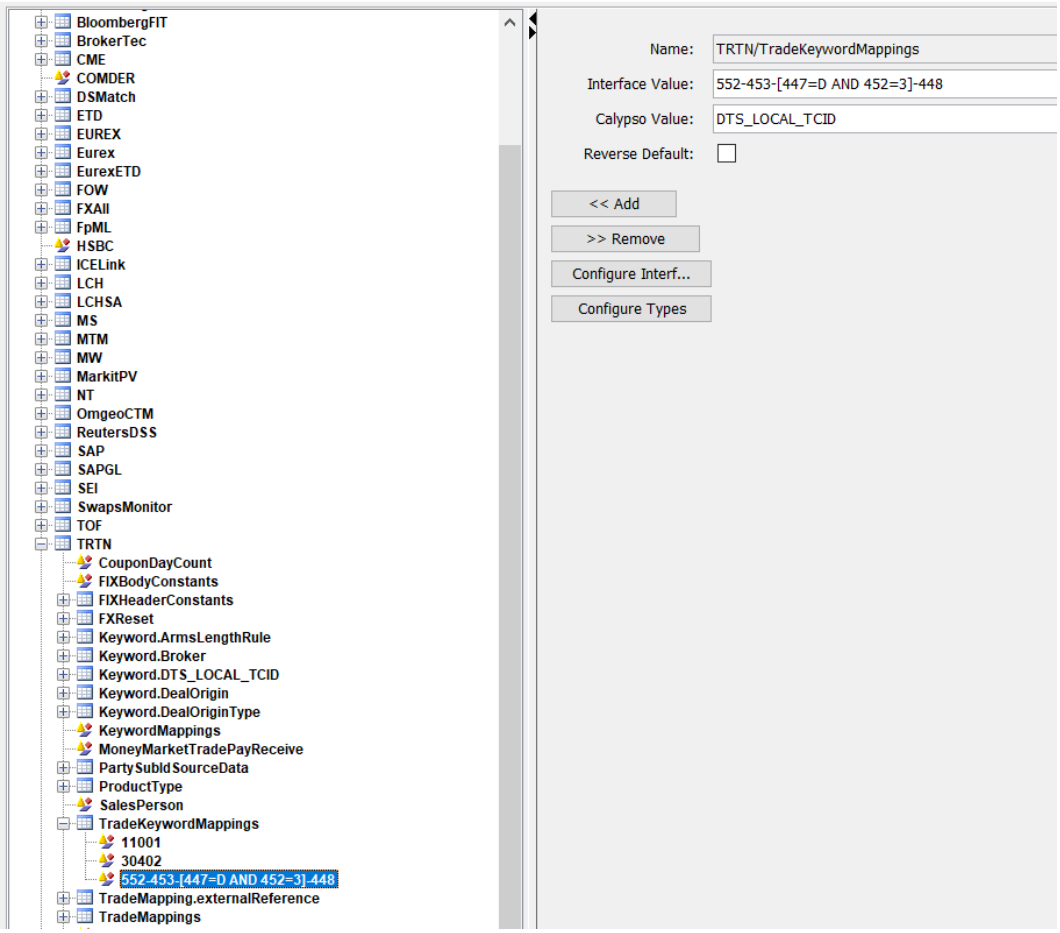
TRTN Mapping

1. Map TradeKeyword EBS_Reference with the value of FIX tag 30402
   Value of FIX tag 30402 will be mapped with EBS_ReferenceNum keyword.



2. Map the value of TradeKeyword DTS_LOCAL_TCID with the value of 448 when 452 = 3. User will have to provide the FIX tag for mapping which means:
    a. 552 - is the main group in the FIX message
    b. 453 - is the subgroup of 552
    c. [447=D and 452=3] - 448 - take the value of 448 tag having 452=3 and 447=D

Calypso Mapping Window



4 Fix-Engine Configuration

The FIX engine is responsible for getting messages from the Refinitiv Trade Notification platform and handing them off to the appropriate workflows.

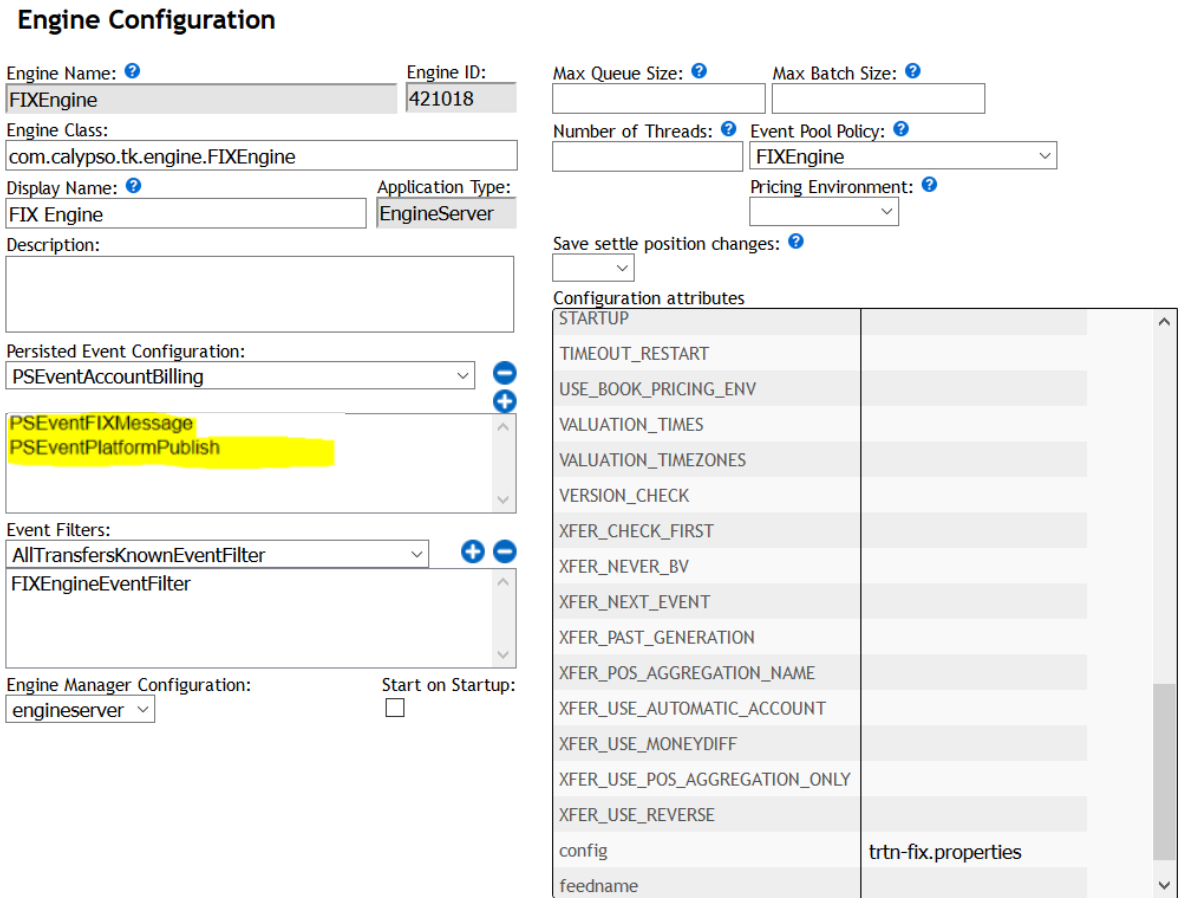
All setup steps are listed below.

Please review also the standard Calypso documentation for engine setup to read about several useful engine parameters (such as thread count) and how to set them.

4.1 Configuring the Engine

All the database-based Engine configuration is completed as part of applying the schema, including the engine name, event subscription, event filter, event policy, as well as assigning a unique id to the Engine.

You may refer to the schema file for more details.



PSEventPlatformPublish is used to send acknowledgements to TRTN.

Engine parameters:

config = trtn-fix.properties

OPTIONAL_FEATURE = trtn

4.2 Setting Up the FIX Config File

To run the FIX engine out-of-the-box, you need a properties file with the name “trtn-fix.properties” with the appropriate FIX connection settings.

A sample file is included under $CALYPSO_HOME/client/resources with the name “trtn-fix.properties.sample”.

You will need to rename the file to “trtn-fix.properties”.

All property files that have been modified need to be copied to `<calypso home>/tools/calypso-templates/resources`.

You will need to re-deploy your environment to your application servers so that they can be included.

Please refer to the Calypso Installation Guide for details on deployment.

Note that, as previously mentioned, the FIX engine uses the QuickFIXJ library for the FIX connectivity implementation. The QuickFIXJ library has many options that can be configured on a FIX session, using a standard properties file. The FIX engine uses this same file for internal settings as well.

For simplicity, we have provided a sample trtn-fix.properties file and will only refer to the minimum settings that must be changed to work with Refinitiv Trade Notification connectivity. You can view all the available settings on the QuickFIXJ Configuration page located at their documentation site at:
http://www.quickfixj.org/quickfixj/usermanual/1.5.3/usage/configuration.html

4.2.1 Sample Properties File

The sample “trtn-fix.properties” file appears similar to the following example:

```
# Default settings for sessions.
# These are inherited by each session defined below
# unless they are overridden in the session settings.
[DEFAULT]
ConnectionType=initiator
ReconnectInterval=10
HeartBtInt=20
LogonTimeout=20
LogoutTimeout=20
Calypso.LogOnInterval=5000
Calypso.LogOnRetryCount=5

# SSL Support
SocketUseSSL=Y
```

```
SocketKeyStore=trtn_dropcopy.jks
SocketKeyStorePassword= password123
EnabledProtocols=TLSv1.2

Calypso.UploadMode=Local
Calypso.PersistMessages=All

# DUMMY_TRTN session definition (TRTN client)
[SESSION]
Calypso.FIXMessageType= TRTN
BeginString=FIXT.1.1
DefaultApplVerID=8
DefaultCstmApplVerID=01.005.00
SenderCompID= CALYPSO
TargetCompID= RTNSFIXUAT
DataDictionary=DD_TRTN.xml
AppDataDictionary=DD_TRTN.xml
SocketConnectHost= pts-uat.trading.thomsonreuters.net
SocketConnectPort=16008
FileLogHeartbeats=Y
FileIncludeMilliseconds=Y
FileIncludeTimeStampForMessages=Y
ValidateIncomingMessage=N
RequiresOrigSendingTime=N
ResetOnLogon=Y
ResetOnLogout=Y
ResetOnDisconnect=Y
StartTime=01:00:00
EndTime=23:00:00
TimeZone=America/New_York
```

4.2.2 QuickFIXJ Settings

To connect to the Refinitiv Trade Notification platform successfully, you need to change the **SenderCompID, TargetCompID, SocketConnectHost** and **SocketConnectPort** connection properties to the correct values for your setup. Please contact Refinitiv Trade Notification support for these details.

Additional points to note regarding the core QuickFIXJ settings:

- The QuickFIXJ settings allow you to configure multiple sessions in a single properties file. This means if you have multiple session logins, you can use a single FIX engine to connect to all of them.

- FileStorePath and FileLogPath are defaulted to USER_HOME/Calypso/FIXEngine/Store and $USER_HOME/Calypso/FIXEngine/Log respectively. These may be overridden at the DEFAULT or SESSION level within the config file. There is no support for other Store or Log mechanisms currently.

4.3 Launching the FIX Engine

4.3.1 Adding Logging Categories

To see logging messages for the Data Uploader and Refinitiv Trade Notification, you need to set the following log categories:

- UPLOADER: Set this to see logging for the Data Uploader translation from the internal Calypso xml format to the actual trade object.
- TRTN: Set this to see logging for the translation from the external format to the internal Calypso xml format.
- FIX: Set this to see logging for the shared FIX connectivity & message processing pieces of the FIX engine.

4.3.2 Running the FIX Engine

With the previous steps completed, you are now ready to run the FIX engine.

To start/stop the engine use the Calypso Engine Server Admin Web Console.

With the engine operating, you can then allege trades through the FXall Trading Terminal. The engine will process the trade messages and create corresponding trades in Calypso.

The Task Station will display any errors that may occur.

4.3.3 Daily Stop/Restart

The FIX server is shutdown daily after business hours and startup again at the start of business the next day. As part of this daily cycle, the Sequence Numbers for the FIX connections are reset as well.

The FIX engine handles this for you automatically, based on the values set in the trtn-fix.properties settings file for the properties StartTime, EndTime, and TimeZone. These properties control when the engine determines that a new session should be started & the Sequence Numbers reset.

For more details on these settings, please refer to the QuickFIXJ documentation site.

5 Test Tool Setup: FileWatcher

Note The details in this section are provided for testing purposes only, and not recommended for production use.

As mentioned in the previous section, the FIX engine also supports processing FIX messages from files. To achieve this, you must run the Data Uploader FileWatcher in Refinitiv Trade Notification mode, so that it will load files from a specified location and pass them on to the FIX engine.

The steps below assume you have already setup the Data Uploader.

5.1 Setup the FileWatcher Config File

To run the FileWatcher for the Refinitiv Trade Notification interface, you need a properties file with the appropriate settings. A sample file "trtnuploader.properties" is included under $CALYPSO_HOME/client/resources with the name ".sample" as the suffix.

Please change the fileDir as required for the polling directory. Also note that the fixSettings property must point to the FIX engine's property file.

For other details, please refer to the Calypso Data Uploader Integration Guide.

All property files that have been modified need to be copied to `<calypso home>/tools/calypso-templates/resources`.

You will need to re-deploy your environment to your application servers so that they can be included.

Please refer to the Calypso Installation Guide for details on deployment.

5.2 Launching FileWatcher

5.2.1 Adding Logging Categories

To see logging messages for the Data Uploader and Refinitiv Trade Notification, you need to set the following log categories:

- UPLOADER: Set this to see logging for the Data Uploader FileWatcher component.
- TRTN: Set this to see logging for the FileWatcher component.
- FIX: Set this to see logging for the shared FIX FileWatcher component.

5.2.2 Running FileWatcher

With the previous steps completed, you are now ready to run the FileWatcher.

To start/stop the engine use the Calypso Engine Server Admin Web Console.

With the FileWatcher operating, you can then place a '.fix' file to import in the watched directory (specified by fileDir in the properties file). At the end of the current wait interval, the FileWatcher will notify the Data Uploader which will then load the file and hand it off to the FIX Engine.

The Task Station will display any errors that may occur.

Please also review requirements for FileWatcher from the Calypso Data Uploader Integration Guide.

6 Test Tool Setup: GUI

Note The details in this section are provided for testing purposes only, and not recommended for production use.

The Refinitiv Trade Notification interface is built on the Data Uploader framework, and therefore supports uploading FIX files through the Data Uploader GUI. This can be useful for testing and does not require you to run the FIX Engine.

The steps below assume you have already setup the Data Uploader as per the Calypso Data Uploader Integration Guide, including adding the GUI window to your menu.

6.1 Setup the GUI config file

Note The need for this step will be removed in a future release.

To upload FIX files through the GUI, you need a properties file with the appropriate settings. A sample file "trtn-datauploader-gui.properties" is included under $CALYPSO_HOME/client/resources with the name ".sample" as the suffix.

Note that the fixSettings property must point to the FIX Engine's property file, although the engine itself does not need to be running.

You must also ensure that the 3rd party jars have been installed on the client side.

6.2 Uploading via the GUI

With the previous steps completed, you are now ready to upload FIX files using the GUI. Simply launch the Data Uploader GUI from the menu, choose the Source/Format, browse to select your '. fix' file, and upload.

For further details on using the Data Uploader GUI, please refer to the Data Uploader Setup Guide.

Note The uploaded file must have a '. fix' extension, not '.xml'

7 Troubleshooting

This section contains details on how to troubleshoot any issues you may encounter.

7.1 Connectivity

The FIX engine automatically attempts to reconnect if a connection with the Refinitiv Trade Notification platform is lost. On reconnect, it will first process any queued messages, and then be available to process new messages.

In case messages are not being received by the engine, please check the log files produced by the QuickFIXJ library. As mentioned previously, the default location for these logs is $USER_HOME/Calypso/FIXEngine/Log. You can also check the Calypso FIX engine logs.

7.2 Message Processing

For message processing failures, check the Task Station of UPLOADSOURCEMSG or GATEWAYMSG workflow errors. Please review the installation section of this document for details.

7.3 Debug Logging

Additional logging can be configured to help with debugging errors; just set the following log categories:

- FIX_DEBUG_XML: Set this category to create xml files of the incoming messages from the Refinitiv Trade Notification platform as well as the interim Calypso xml format created by the translation under $USER_HOME/Calypso/TRTN. Note that these files can be used for loading via the File Watcher.
- FIX_DEBUG_API: Set this category to see additional logging statements for the FIX connectivity, including Login/Logout and Admin messages (such as resend requests). These logging statements are helpful to debug any FIX connectivity issues, as well as see headers of the messages coming in from the Refinitiv Trade Notification platform before they are interpreted by the FIX engine.

7.4 Reporting Issues to Calypso

If you still need to contact Calypso, please ensure that your ticket contains the following information:

- All available logs, including:
  - QuickFIXJ connectivity logs
  - FIX engine logs
  - FileWatcher logs
  - DataServer and EventServer logs
- FIX messages, if applicable
- Clear description of the issue, including:

- What steps were executed to produce the issue, both in the FXall Trading Terminal and Calypso
- What attempts, if any, were made to debug the issue, and what were the results

8 FAQ

1. Unable to connect to Refinitiv Trade Notification platform

Ans: It can be a network issue. Try Telnet on the IP and port provided by the Refinitiv Trade Notification platform. If Telnet fails kindly check firewall and contact Refinitiv support.

2. FIX Session frequently disconnect

Ans: It is mostly due to poor network connectivity. Kindly contact Refinitiv Trade Notification support and ask them to confirm session heartbeats.

3. How to enable logs for Refinitiv Trade Notification

Ans: Add debug log categories 'TRTN, FIX' on EngineServer/Navigator.

4. Unable to connect to FIX session during certain time.

OR

FIX connectivity stops after certain period.

Ans: Refinitiv Trade Notification provides StartTime and EndTime, only during which user can connect to FIX session. User will not be able to connect to fix session before StartTime and after EndTime.

This connectivity time frame can be controlled via fix properties.

Example:

StartTime=07:00:00

EndTime=23:00:00

TimeZone=America/New_York

For StartTime and EndTime value, please contact Refinitiv Trade Notification support.

5. FIX message gets rejected due to missing or unorder tag.

OR

Data-Dictionary outdated / out of sync.

Ans: Refinitiv Trade Notification interface keeps it data-dictionary in sync with Refinitiv Trade Notification platform. But in case the data-dictionary is outdated and fix messages start getting rejected from FIX session, then user is advised to disable the data-dictionary validation by adding property 'ValidateIncomingMessage=N' in fix config file. Disabling the validation check, will stop the fix messages from getting rejected. The Refinitiv Trade Notification interface has its own data validation check and if any required data is missing or invalid on fix message, the Refinitiv Trade Notification interface will log an error in task-station.