Nasdaq Calypso

Traiana Matching Integration Guide

Version 5.3.0

Revision 7.0
January 2024
Approved

Document History

Revision	Published	Summary of Changes
1.0	August 2019	First edition for version 2.0.1
2.0	November 2019	Second edition for version 2.0.2
3.0	March 2020	Third edition for version 3.0.1
4.0	November 2020	Fourth edition for version 2.0.5, 3.2.1
5.0	September 2021	Fifth edition for version 3.3.0
6.0	February 2022	Sixth edition for version 3.5.0, 4.1.0.
7.0	January 2024	Seventh edition for version 5.3.0 (compatibility with version 18)

This document describes the Calypso Traiana-matching Interface setup. The Traiana-matching interface allows buy-side client (Calypso) to send out trades booked in Calypso to Traiana Harmony platform for equity CFD which can further be matched inside Traiana platform.

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

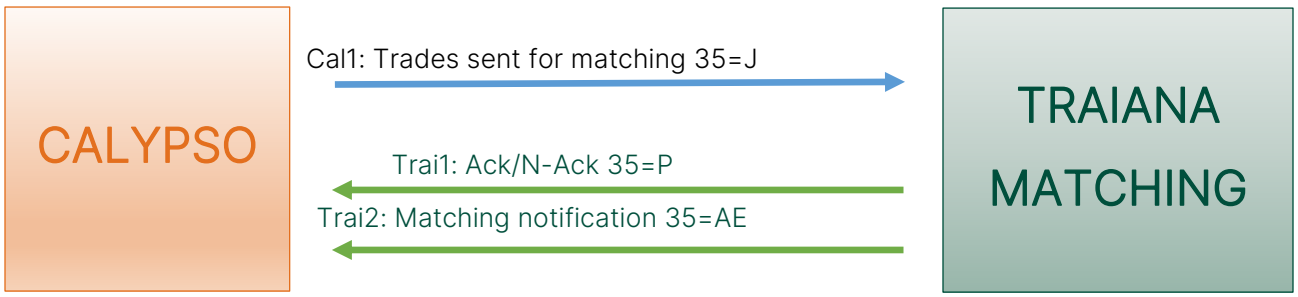
Introduction

This document describes the Calypso Traiana-matching Interface setup. The Traiana-matching interface allows buy-side client (Calypso) to send out trades booked in Calypso to Traiana Harmony platform for equity CFD which can further be matched inside Traiana platform.

This document describes the configuration required to setup the Traiana-matching interface to run successfully.

Note - The Traiana-matching interface is distributed as part of the Traiana-matching module, which may support other Traiana-matching interfaces which must be licensed separately.

1.1 Calypso Traiana-matching Interface Flow



Flow explanation:

- Cal1: Trades from Calypso (Buy-side client) sent for matching to Traiana
  1. Trades in Calypso are sent to Traiana platform individually for matching
  2. These trades are sent to Traiana using a fix message
  3. The fix message created is unique for every trade and its identified uniquely through Calypso trade-id which is generated in fix message when it is sent to Traiana platform.
  4. For this outgoing message the type of fix message generated is of type 35=J
  5. Following are the type of fix message generated from Calypso based on trade attribute PlatformSubmitStatus

Trade msg type	Fix representation	PlatformSubmitStatus
New	71 = 0 (AllocTransType)	When this keyword is not present on trade it is considered that this is new trade allege to Traiana
Replace	71 = 1 (AllocTransType)	When this keyword is present on trade it is considered as amend trade allege to Traiana

Trade msg type	Fix representation	PlatformSubmitStatus
Cancel	71 = 2 (AllocTransType)	When this keyword is present on trade with value "Confirmed" then we are alleging trade for un-match to Traiana of type Cancel

- Trai1: Ack/N-Ack received from Traiana
  1. For the trades sent from Calypso to Traiana are first evaluated by the platform for correct statics and Traiana acknowledges them by sending back Ack or N-Ack
  2. Based on the type of acknowledgement received trade attributes are updated on trade with status and reason
  3. For this incoming message the type of fix message we receive is of type 35=P
  4. Acknowledgement type received by Calypso for the trades sent in CAL1

Ack type	Fix representation	Update on Calypso trade
Ack	87 = 0 (AllocStatus)	Trade attribute is updated with status as Pending
N-Ack	87 = 1 (AllocStatus)	Trade attributes are updated with status as Rejected and reject reason coming in fix field 58

- Trai2: Matching confirmation received from Traiana
  1. Post acknowledging the trade request sent from Calypso, Traiana performs matching on the trades which are acknowledged without any error.
  2. Traiana sends the result of the matching confirmation performed internally to Calypso via this matching notification.
  3. For this incoming message the type of fix message we receive is of type 35=AE
  4. Matching notification can be of following three types based on which trade keywords are updated with matching status

Matching msg type	Fix representation	Update on Calypso trade
Matched	573 = 0 (MatchStatus)	Trade attribute are updated with status as Confirmed
Unmatched	573 = 1 (MatchStatus)	Trade attribute are updated with status as Unmatched
Mismatched	573 = 2 (MatchStatus)	Trade attribute are updated with status as Dispute

1.2 Calypso Traiana Matching Interface Features

1.2.1 Supported

Product Type:

The Traiana-matching interface supports the following Calypso trade types:

- Portfolio Swap trade

Trade lifecycle:

- NEW
- AMEND
- CANCEL

Note - Trade messages which are sent out for matching are not part of allocation lifecycle executed in Calypso could be a scenario client would receive them as part of external allocation, but the interface does not consider the allocation parent-child relation.

Fix Messages:

- Outbound message (Calypso-Traiana) ➡

Calypso trades which are intended to be sent to Traiana is sent via fix message generated of type 35 = J

- Inbound message (Traiana-Calypso) ⬅
  - Ack/N-ack

    For trades sent by Calypso to Traiana we receive acknowledgement from Traiana of type 35 = P
  - Matching confirmation

    For trades successfully acknowledged by Traiana further matching notifications are sent via fix message of type 35 = AE

Note - The interface only supports processing of message 35=P & 35=AE, messages other than this for example 35=8 are ignored.

1.2.2 Not Supported

The following are not supported by the Traiana-matching for interface:

- Allocation performed in calypso and sending block and child trades in a single message for matching.

Setup Instructions

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

The module supports specific versions of Calypso. In addition, your implementation must have the current Hotfixes applied. Before downloading the Traiana-matching module, please refer to the release notes to determine which module versions are applicable for your implementation. The cal-upload.jar is bundled with the Data Uploader and no longer has a separate listing in the table. cal-upload.jar must still appear in the CLASSPATH.

2.2 Installation Instructions

2.2.1 Setup config data using Execute SQL

Add the following files to Execute SQL from $CALYPSO_HOME/bin/dbscripts:

- GatewaySchemaBase.xml
- GatewaySchemaData.xml
- FIXSchemaData.xml
- TraianaSchemaData.xml

2.2.2 Message Workflow Setup

The Traiana module uses the UPLOADSOURCEMSG, GATEWAYMSG and PLATFORMMSG workflows when processing messages. These should have been setup as part of the Data Uploader Setup Guide.

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.

PLATFORMMSG workflow is use for publishing the trades to external platform from Calypso.

2.2.3 Task Station Setup

The Traiana-matching module 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 see the Data Uploader Setup Guide for how to add the appropriate messages and exceptions to the Task Station.

Legal Entity Mapping

The outgoing trade FIX messages contain Legal Entity identifiers for all parties involved in the trade (Party and Counterparty).

The Legal Entity for the Party and Counterparty and Executing Broker identifiers populated in the FIX message need to be configured in Calypso using the Legal Entity Attribute '**TraianaParticipant**', and the value must be the BIC code provided by Traiana.

However, clients can generate their own BIC code map it against the calypso legal-entity attribute as mentioned above and ask Traiana to configure same legal entities.

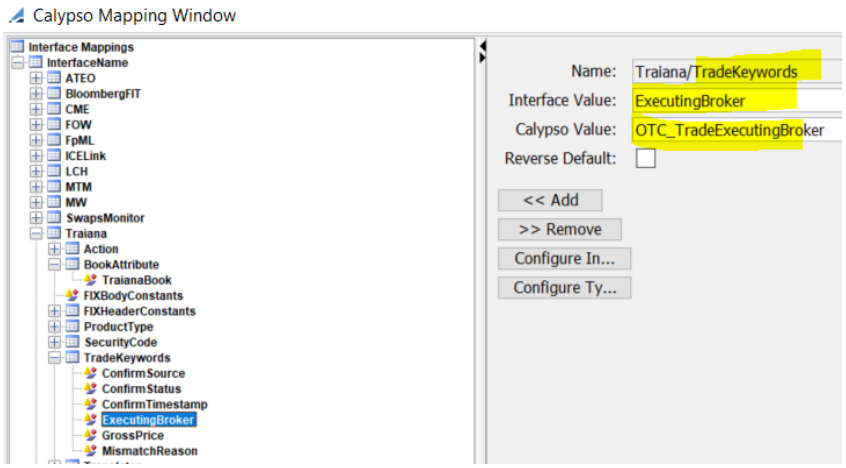
The logic will look for the attribute value TraianaParticipant configured on the Legal Entity involved in trade booking (Book-LE, CP-LE and Broker-LE) which will used to generate the fields for legal entity.

This lookup logic will be applied to PO (Book-LE), Counterparty and Executing Broker keyword lookups. If no Calypso Legal Entity is found using the rules above, an error will be raised.

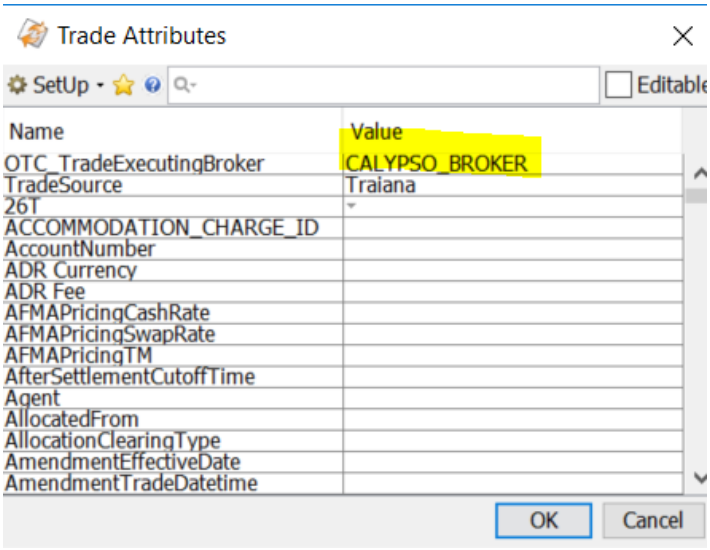
3.1 Broker Mapping

For the Executing Broker keyword provided on the Calypso trade, the mechanism to handle and process the mapping is described below.

1. Identify the Executing Broker keyword name from the mapping whose value needs to be chosen from the keywords shown below:

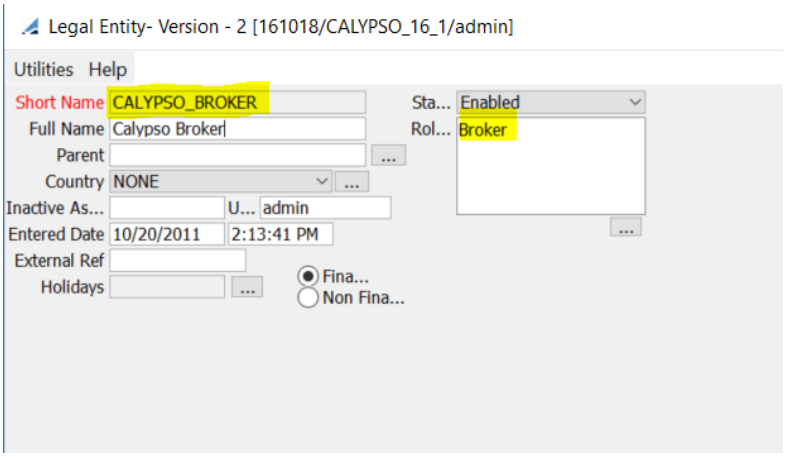


2. Knowing the keyword name from the mapping, get the value of it from the Calypso trade to be alleged out to Traiana as shown below:

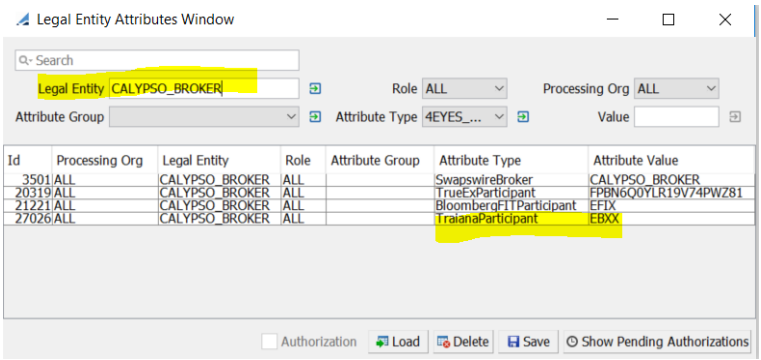


3. The value to be alleged will be decided using the following logic:

- Find the Legal Entity in Calypso with a short name the same as the keyword value and with the Role of Broker mapped to it.



- When a boker LE is found, the value configured in the LE attribute **TraianaParticipant** is retrieved and used to allege in the message.



- If any configuration is missing, such as if an LE is not found, an LE with the specific role is not found or an LE does not have the attribute TraianaParticipant configured, the keyword value is sent as it is in the outgoing message to Traiana.

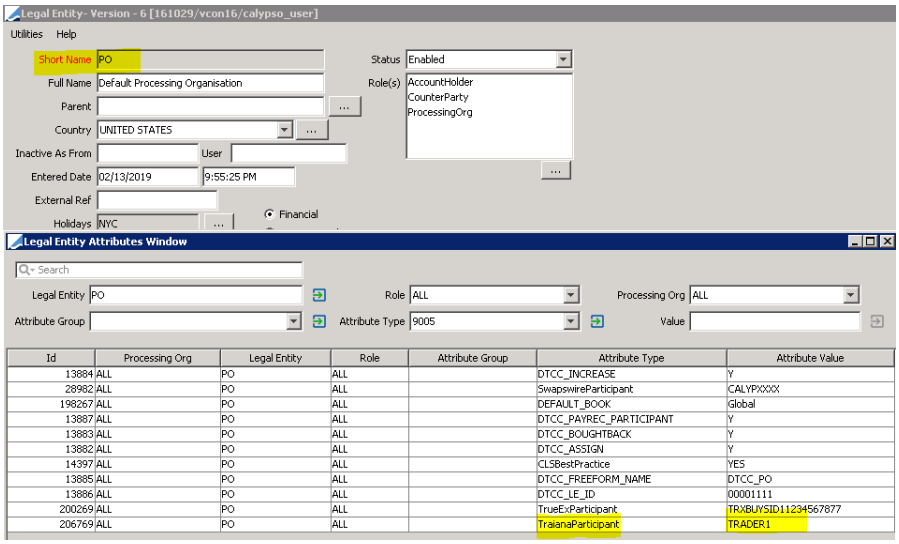
The value “EBXX” is populated in the fix message under fields:

448=EBXX|447= C | 452=1

The field PartyRole -> 452 =1 signifies that EBXX should act or be mapped in Traiana to an Executing Firm.

3.2 PO Mapping

For the Calypso legal entity mapped on the Book which is intended to be used in trade booking must be mapped to attribute **TraianaParticipant** with the BIC code value configured in Traiana platform. See below illustration.



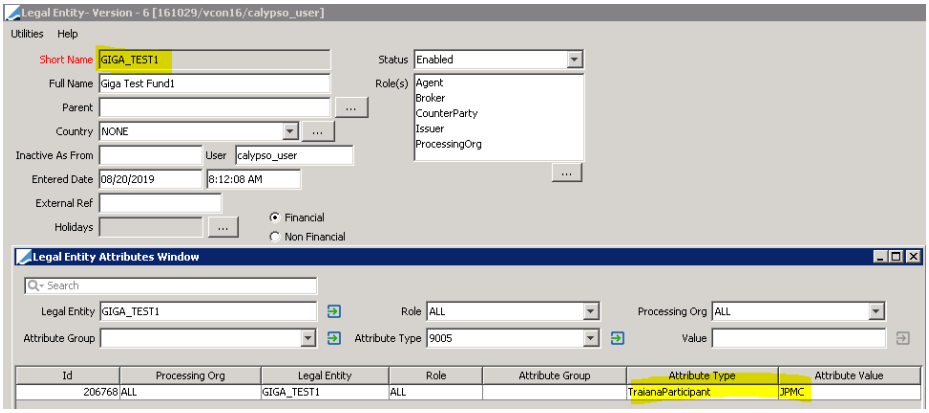
The value “TRADER1” will be populated in fix message under fields:

448=TRADER1|447=C |452=3

The field PartyRole -> 452 =3 signifies that TRADER1 should act or mapped in Traiana to a ClientID.

3.3 Counterparty Mapping

For the Calypso legal entity which is intended to be used as counterparty in trade booking must be mapped to attribute **TraianaParticipant** with the BIC code value configured in Traiana platform. See below illustration.



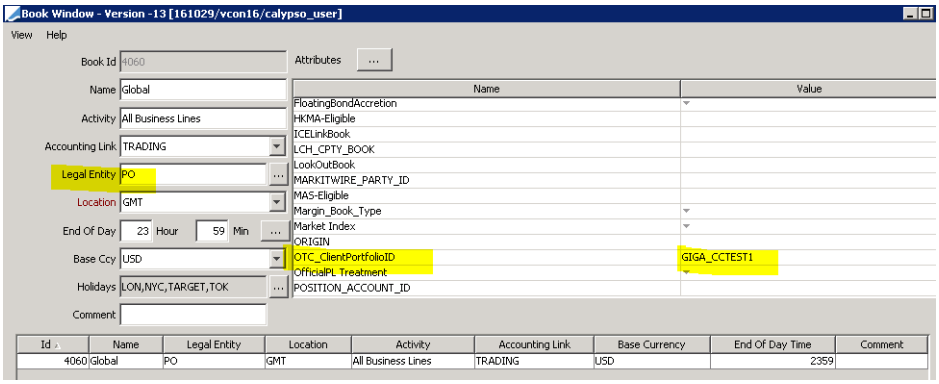
The value “JPMC” will be populated in fix message under fields:

539=1|524=JPMC|525=C|538=4

The field PartyRole -> 538 =4 signifies that JPMC should act or mapped in Traiana to a Prime Broker.

Book Mapping

In Calypso the book that is to be used for trade bookings for Traiana related trades must be mapped to the legal entity as mentioned in PO Mapping.



The Calypso value configured in the mapping window under Traiana >BookAttribute > TraianaBook is looked up on the trade book and populated under tag 79.

When the book Global is the intended book to be used on the trade with the LE mapped to PO, then the value of the book attribute (GIGA_CCTEST1 in this case) found from the Calypso mapping (OTC_ClientPortfolioID) is populated under tag 79, which corresponds to a fund level account.

78=1|79=GIGA_CCTEST1

Product Security Code Mapping

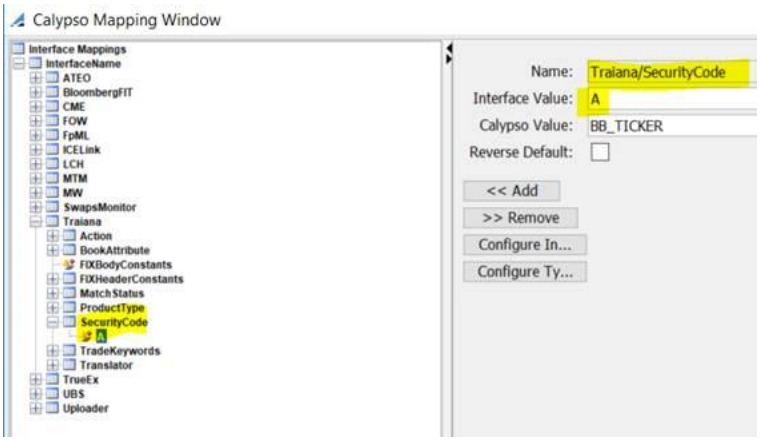
On the underlying product selected in trade when there are multiple security code/value pair configured then below preference is followed to choose one and generate its value in Fix tag 22 and tag 48:

- **Bloomberg Symbol** which is sent as 'A' in tag 22 and tag 48 = <security value of underlying for Bloomberg Symbol> in fix message
- **ISIN** which is sent as '4' in tag 22 and tag 48 = <security value of underlying for ISIN> in fix message
- **SEDOL** which is sent as '2' in tag 22 and tag 48 = <security value of underlying for SEDOL > in fix message
- **RIC** which is sent as '5' in tag 22 and tag 48 = <security value of underlying for RIC > in fix message
- **CUSIP** which is sent as '1' in tag 22 and tag 48 = <security value of underlying for CUSIP > in fix message
- **CUSTOM**, see below for details.

Above are default mappings, however you can customize them for any of following two requirements:

- Override Bloomberg Symbol - A mapping with some other security code, suppose from underlying product if we need to send some other code in place of Bloomberg Symbol then configure following mapping.

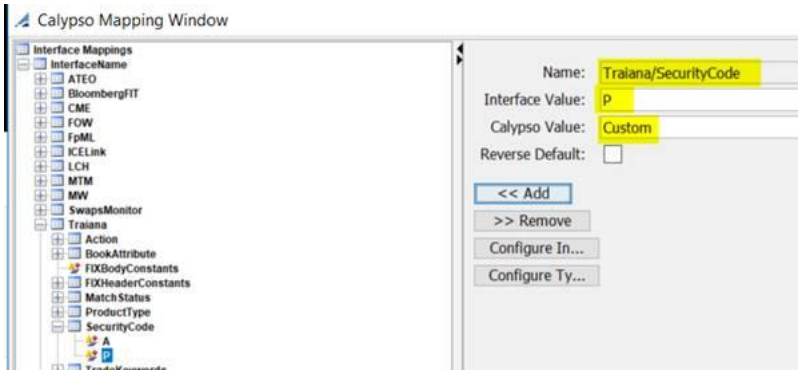
With this mapping if BB_TICKER is found on your underlying its security value will be sent in fix message in tag 48 and tag 22 will be sent as A



- If there is a requirement to send in fix message an altogether separate security code and value apart from these defaults ones -> A (Bloomberg Symbol), 4 (ISIN), 1 (CUSIP), 2 (SEDOL), 5 (RIC),

Following mapping is needed and make sure the underlying has no other default security codes mentioned above otherwise they would be given more priority over this.

Note: Custom security code is used for understanding purpose can be some actual business code.



In this case when we find Custom on underlying then it will be sent in fix messages as tag 22=P and tag 48 = <Security value of underlying for Custom>.

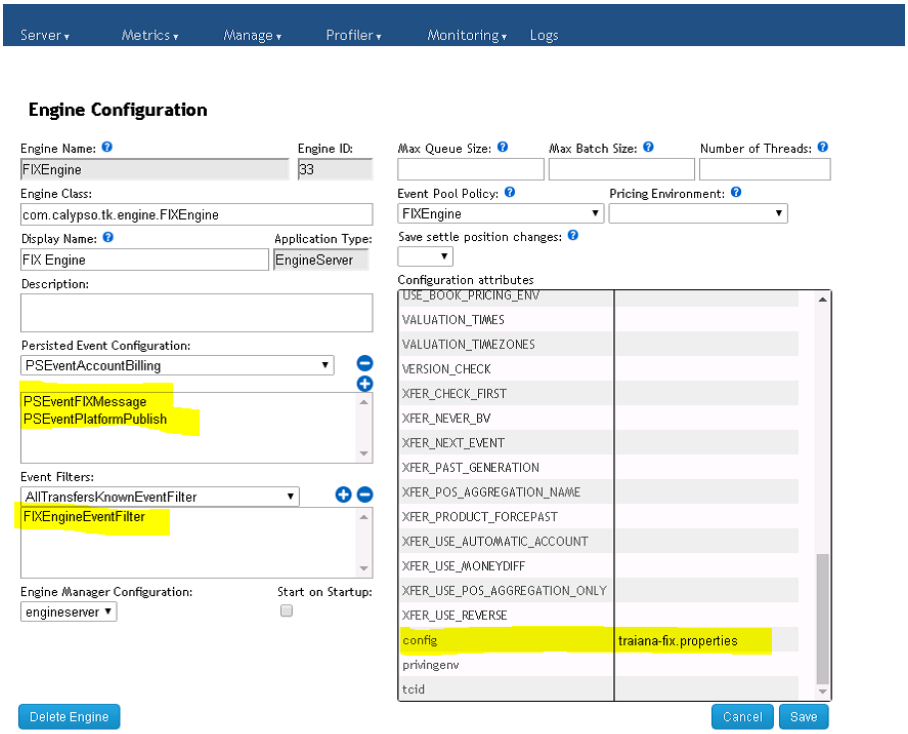
Fix-Engine Configuration

Calypso FIX Engine is responsible for sending and receiving messages from the Traiana-matching platform.

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

6.1 Configure the Engine

Following is screen snap which shows parameters need to be configured on Fix-engine for connecting to Traiana platform and events with filter need to be present for sending and receiving messages to Traiana



6.2 Setup the FIX Properties File

To run the Traiana-matching FIX Engine out-of-the-box you will need a properties file with the name "**Traiana-fix.properties**" with the appropriate FIX connection settings.

A sample file is included under $CALYPSO_HOME/client/resources with the name "traiana-fix.properties.sample".

You will need to rename the file to "traiana-fix.properties.properties".

Once modified, copy the property file to "`<calypso home>/ tools/calypso-templates/resources`".

You will then need to deploy the files to your applications servers.

Please refer to the Calypso Installation Guide for details.

Note that, as previously mentioned, the Traiana 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 Calypso Traiana-matching FIX Engine uses this same file for internal settings as well.

For simplicity, we have provided a sample Traiana-matching-fix.properties file and will only refer to the minimum settings that must be changed to work with Traiana-matching 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

6.2.1 Sample Properties File

The sample “traiana-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=Calypso-Adapter-Fix-FIX.1019122.jks
SocketKeyStorePassword=273nbGSb
EnabledProtocols=TLSv1.2

Calypso.UploadMode=Local
Calypso.PersistMessages=All

# DUMMY_SEFCLIENT session definition (Traiana-matching SEF client)
[SESSION]
Calypso.FIXMessageType= Traiana
BeginString=FIX.1.1
DefaultApplVerID=7
SenderCompID= CALYPSO
```

```
TargetCompID= TRAIANA
DataDictionary=DD_Traiana.xml
AppDataDictionary=DD_Traiana.xml
SocketConnectHost=199.201.113.144
SocketConnectPort=443
FileLogHeartbeats=Y
FileIncludeMilliseconds=Y
FileIncludeTimeStampForMessages=Y
ValidateIncomingMessage=N
RequiresOrigSendingTime=N
ResetOnLogon=Y
ResetOnLogout=Y
ResetOnDisconnect=Y
StartTime=07:00:00
EndTime=23:00:00
TimeZone=Europe/London
```

6.2.2 QuickFIXJ Settings

To connect with Traiana platform successfully, you will need to change the SenderCompID, TargetCompID, SocketConnectHost, SocketConnectPort, SocketKeyStore and SocketKeyStorePassword connection properties to the correct values for your setup. Please contact Traiana-matching support for these details.

For values in properties SocketKeyStore and SocketKeyStorePassword traiana-connectivity need to contact as they will provide the ". jks" which will contain the key-store certificates needed to create fix-session for the given Sender and Target CompId

Additional points to note regarding the core QuickFIXJ settings:

- 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 at this time.

6.3 Launching the Traiana FIX Engine

6.3.1 Adding Logging Categories

To see logging messages for the Data Uploader and Traiana-matching modules 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.
- Traiana: Set this to see logging for the Traiana 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 Traiana-matching FIX Engine.

Additional debugging categories are listed in the Troubleshooting section of this document.

6.3.2 Running the Traiana FIX Engine

With the previous steps completed, you are now ready to run the Traiana-matching FIX Engine.

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

With the Traiana Engine operating, you can then allege trades from Calypso with workflow configuration showed in Section 6. The Traiana fix Engine will process the trade messages and will send them to Traiana platform for matching. Based on the response received from Traiana platform details will be updated on corresponding trades in Calypso.

The Task Station will display any errors that may occur.

6.3.3 Daily Stop/Restart

The Traiana FIX server's 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 Calypso Traiana-matching FIX Engine handles this for you automatically, based on the values set in the traiana-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.

Trade Workflow

This section describes the Traiana-matching interface trade workflow. It is important to understand these details so that the Calypso Trade Workflow can be customized accordingly. Please read and follow all setup instructions carefully to ensure a successful installation.

7.1 Trade Workflow Setup

To support all the transitions required by the Traiana-matching workflow, the trade workflow setup for processing Traiana-matching trades must support the following transitions:

- NEW (for creation of new trades)
- Amend (for updating the incoming notifications on trade)
- Cancel (for cancelling trade)

Workflow rules:

- TraianaSource trade rule:
  - This rule gets populated in domain as part of running Traiana schema on executeSQL
  - This rule when applied on a transition say PENDING-VERIFY-VERIFIED updates the trade with keyword TradeSource=Traiana
  - The TradeSource keyword need to have the value Traiana because it helps to generate publish events which can be then be processed by fix-engine for sending the trade details to Traiana based as fix message.
  - When this workflow rule is not present on trade users have to manually add the keyword value on trade before alleging it to Traiana platform.
- PlatformAllege trade rule:
  - This rule gets populated in domain as part of running datauploader schema on executeSQL
  - This rule alleges calypso trade to target platform
  - The rule when applied on a transition say VERIFIED-PO_ALLEGE-VERIFIED basically compares the keyword value TradeSource=Traiana against the values in domain UploadMessageSourceTypes. If the domain has the exact TradeSource value in our case which is Traiana the rule generates PSEventPlatformPublish events which are listened by fix-engine
  - Fix-engine on receiving the intended publish events processes them and send the fix message to target platform.

Note - Both the above rules can be applied on any transition based on how user have designed the trade WF to send messages to platform

7.2 Trade Capture Sequence

The table below describes the sequence of Traiana-matching FIX message flows in and how are they handled.

Calypso	Traiana Matching
Step1: Create a trade in Calypso having keyword value TradeSource=Traiana When this trade is alleged a FIX, message is sent to Traiana for matching	
	Step2: On receiving the fix message Traiana will send an ACK fix message that will indicate the message were processed by it without any errors or a NACK fix message to notify the Client that the message failed for some reason
Step3: Ack: For sent trades when calypso receives an Ack it updates the trade keyword with status as **Pending** N-Ack: If calypso receives N-Ack the trade keyword is updated with status as **Rejected** with reject reason. Once the trade is rejected then user cannot send it for further matching until the rejection reason is corrected in deal.	
	Step4: Post sending the Ack/N-Ack Traiana further perform matching on trades which were processed successfully and will send matching confirmation for these trades via fix message
Step5: Match: For successfully acknowledged trades when Calypso receives matching message then the trade keyword are updated with status as **Confirmed** Mismatch: For successfully acknowledged trades when Calypso receives mismatched message then the trade keywords are updated with status as **Dispute,** now user can choose here further to cancel this trade and book a completely new trade where user must start from **Step1** or move to **Step6.**	

Calypso	Traiana Matching
Step6: **Amend trade:** In case of mismatch we can amend the trade in Calypso and re-allege same. When the trade is alleged a FIX, message is generated of type Replace or Cancel and sent to Traiana again for matching To support Cancel, you need the following mapping: Name = Traiana/Translator Interface Value = TradeStatus Calypso Value = CANCELED	⇨
	On receiving this fix message flow from **Step2** to **Step 5** is followed.
Step7A: **Cancel trade:** When the trades are matched and we need to un-match them, then cancel the trade in Calypso and re-allege same. On doing this a FIX, message is generated of type Cancel and sent to Traiana again for unmatching	⇨
⇦	**Step7B:** On receiving this fix message Traiana will sent Ack/N-ack like Step2
Step7C: On receiving the Ack/N-ack activities are performed as Step3	
⇦	**Step7D:** Post sending the Ack/N-Ack Traiana further perform unmatching on trades which were processed successfully and will send confirmation for these trades via fix message
Step7E: UnMatch: For successfully acknowledged trades when Calypso receives unmatched message then the trade keywords are updated with status as **Unmatched**	

Trade Keywords

Following are the trade attributes/keywords supported.

Note - This is an early release version and keyword names are subject to change. Such changes will be documented in the Release Notes, so please ensure you review them whenever upgrading from previous Traiana interface versions.

8.1 Supported Trade Keywords

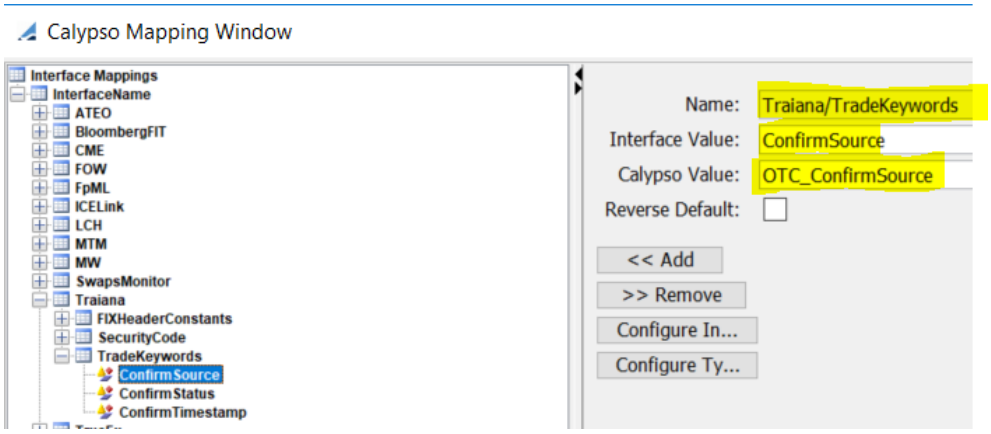
Keyword Name	Description	Comments
TradeSource	Always set to 'Traiana'	Used for engine logic behind the scenes; do not change.
PlatformAllegeType	Platform message allege confirmation	
PlatformSubmitStatus	Platform allege status	
PlatformMatchId	Platform Id generated by Traiana platform and sent to calypso in incoming notifications	
ExecutionDateTime	Date time of last notification received from Traiana platform	
Custom-ConfirmSource	Like keyword TradeSource	Need to configure calypso mapping to populate these keywords as described in section 7.2
Custom-ConfirmStatus	Like keyword PlatformSubmitStatus	
Custom-ConfirmTimestamp	Like keyword ExecutionDateTime	

8.2 Custom Trade Keyword Mapping

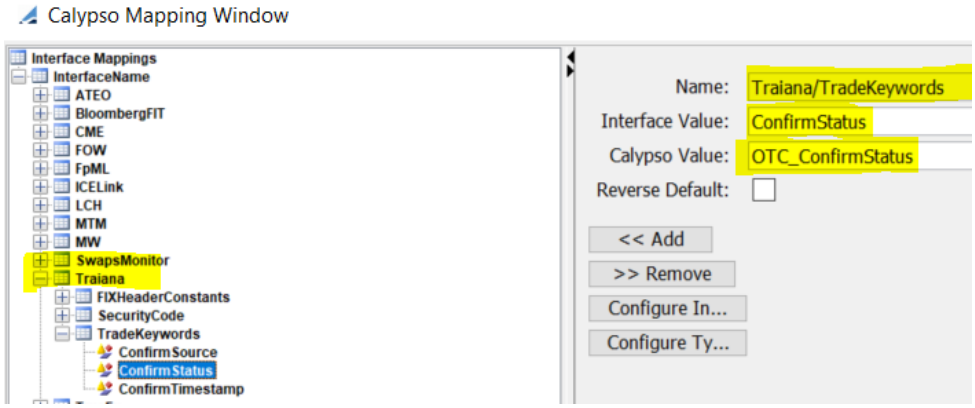
In calypso mapping window when mappings are provided under type:

Traiana/TradeKeywords, they are used to populate following keyword-name on trades

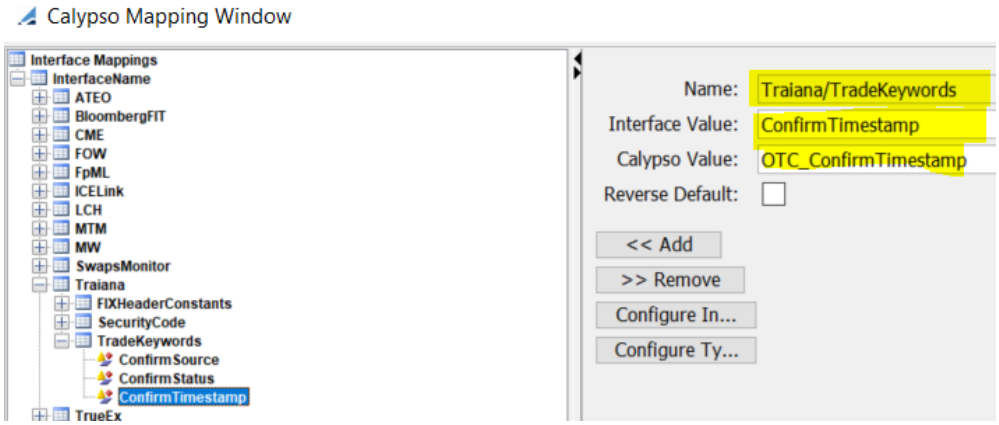
- ConfirmSource



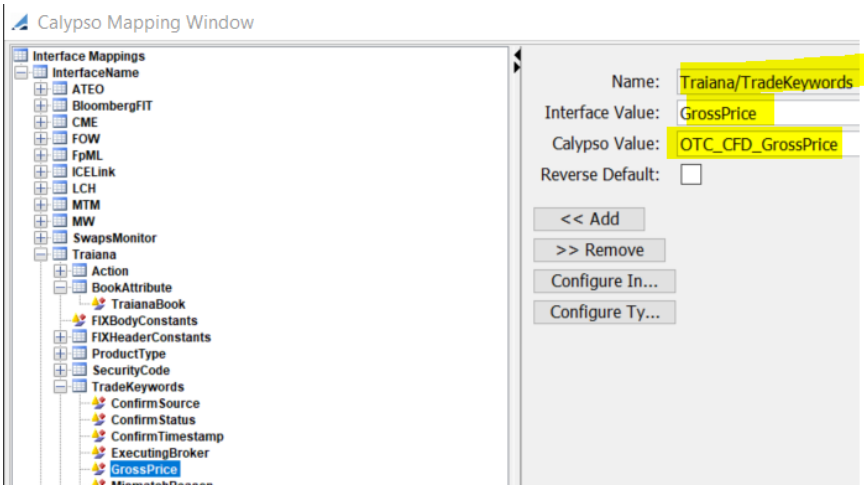
- ConfirmStatus



- ConfirmTimestamp

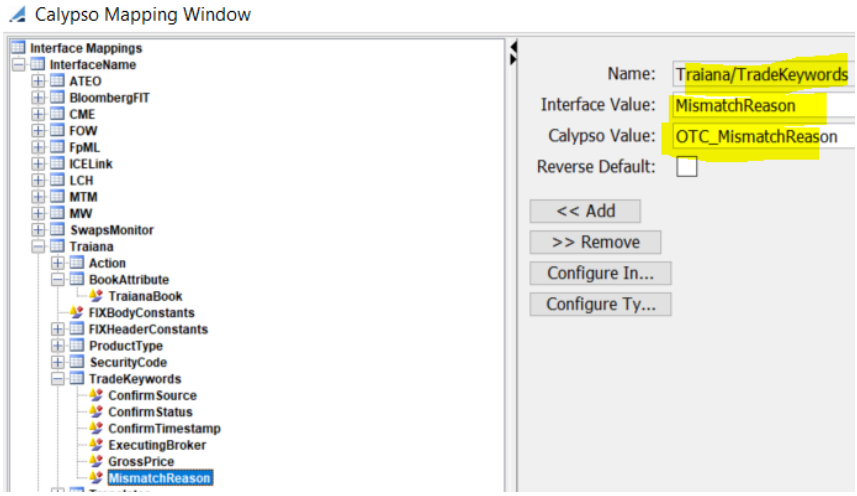


- GrossPrice



As per the above mapping value populated in keyword OTC_CFT_GrossPrice on trade will be sent out from Calypso to Traiana in tag **10220 – GrossPrice**.

- MismatchReason

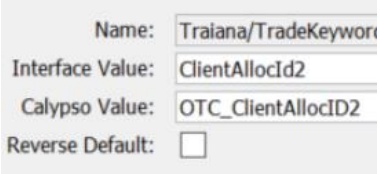


In case of trade mis-match notification coming from Traiana, the mis-match reason comin in tag-142 will be populated in the Calypso value OTC_MismatchReason, configured above as a trade keyword.

- Tag 70, 72 is populated from trade keyword and trade-id. To support this feature, the following mapping has been added.

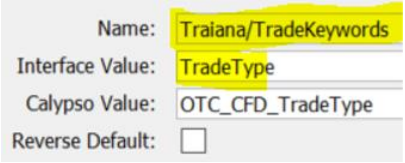
If this mapping is not provided, Tag 70, 72 is populated from trade keyword OTC_ClientAllocID by default. If trade keyword is empty, Tag 70, 72 is populated from trade-id.

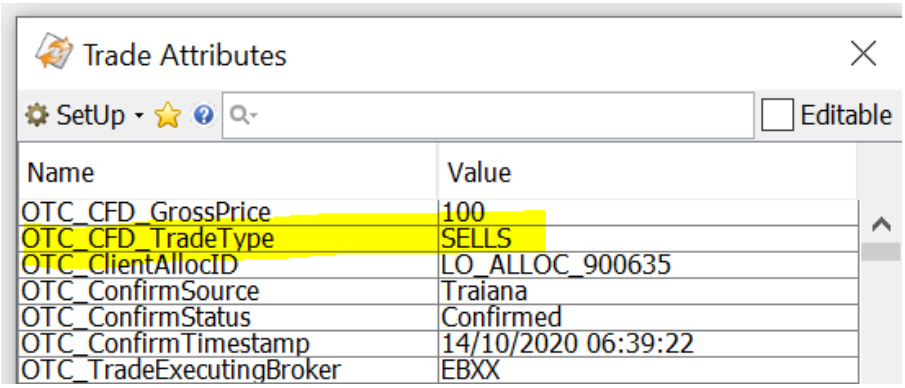
- OTC_ClientAllocID2 is sent for CANCEL messages in tag 72. To support this feature, the following mapping has been added.



If this mapping is not provided, Tag 72 is populated from trade keyword OTC_ClientAllocID2 by default.

- TradeType





As per the above mapping value populated in keyword OTC_CFD_TradeType on trade will be sent out from Calypso to Traiana in tag **10225** with interpretation to SHORT/LONG. When no mapping is provided, by default system will look in keyword OTC_CFD_TradeType.

Keyword value will be interpreted to SHORT and LONG by following logic:

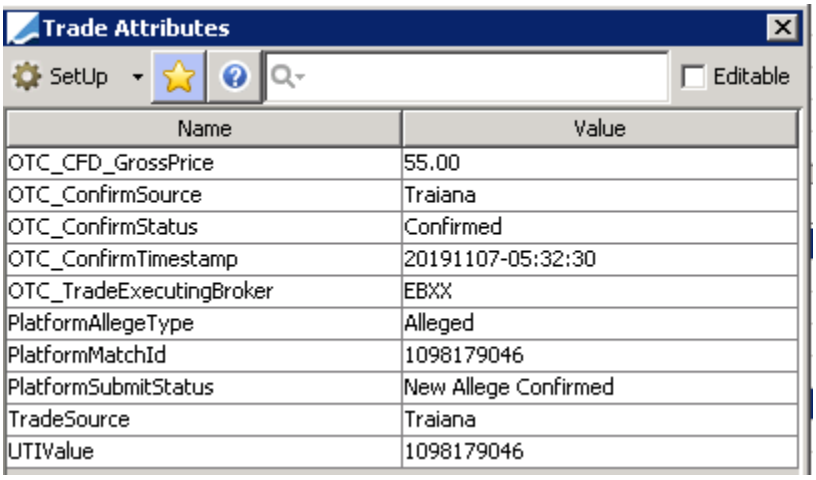
- When keyword value = BUYS or SELLS then tag 10255 = SHORT
- When keyword value = BUYL or SELLL then tag 10255 = LONG

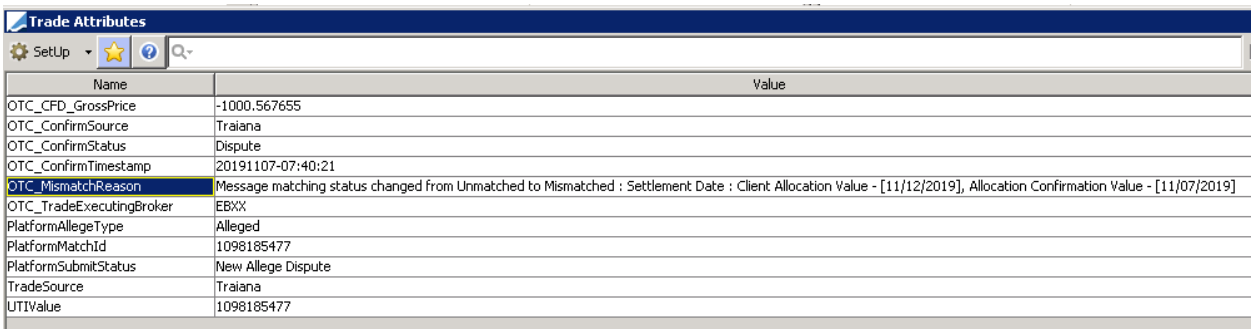
Look-up logic:

Interface value (ConfirmSource, ConfirmStatus, ConfirmTimestamp, GrossPrice, MismatchReason) need to be constant and known to us so that we look for corresponding Calypso-value and populate on trade attribute, the interface value can be dynamic which would be picked over look-up performed for mapping.

If mapping is not in place, values will be populated in platform default keywords which are PlatformSubmitSatus, TradeSource, ExecutionDateTime.

Trade attribute window snapshot with custom keywords:





Bock Level Matching:

For 35=AE messages with 828=3 notification, the following trade keywords are stored:

- OTC_BLOCKConfirmStatus – Based on the following mapping:

Name = Traiana/TradeKeywords

Interface Value = BLOCKConfirmStatus

Calypso Value = OTC_BLOCKConfirmStatus

- OTC_BLOCKConfirmTimestamp – Based on the following mapping:

Name = Traiana/TradeKeywords

Interface Value = BLOCKConfirmTimestamp

Calypso Value = OTC_BLOCKConfirmTimestamp

- PlatformBLOCKMatchId – Tag 880
- PlatformBLOCKRejectReason – Tag 142

Custom Tag Support based on Mapping

This feature can help a user to send fix tags in the fix message based on mapping configuration. User simply need to provide mappings as per below instructions and fix tag with its corresponding value will be populated in the outgoing fix message.

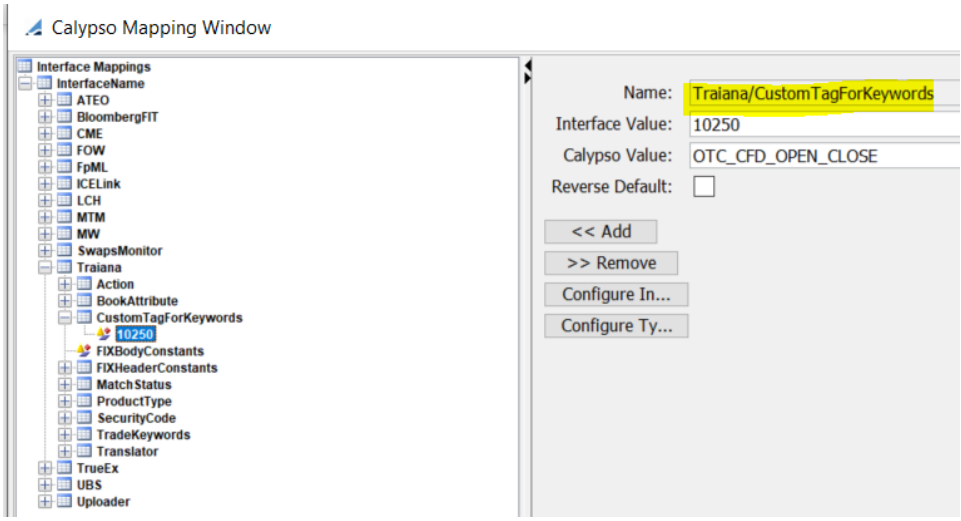
9.1 Fix Tags for Keywords

In this type we map a Fix tag to a Keyword-name in following way:

Mapping Type: CustomTagForKeywords

Interface Value: Fix tag

Calypso value: Trade Keyword name



For trades sent to traiana if this keyword is present, its value will be populated in the outgoing fix message.

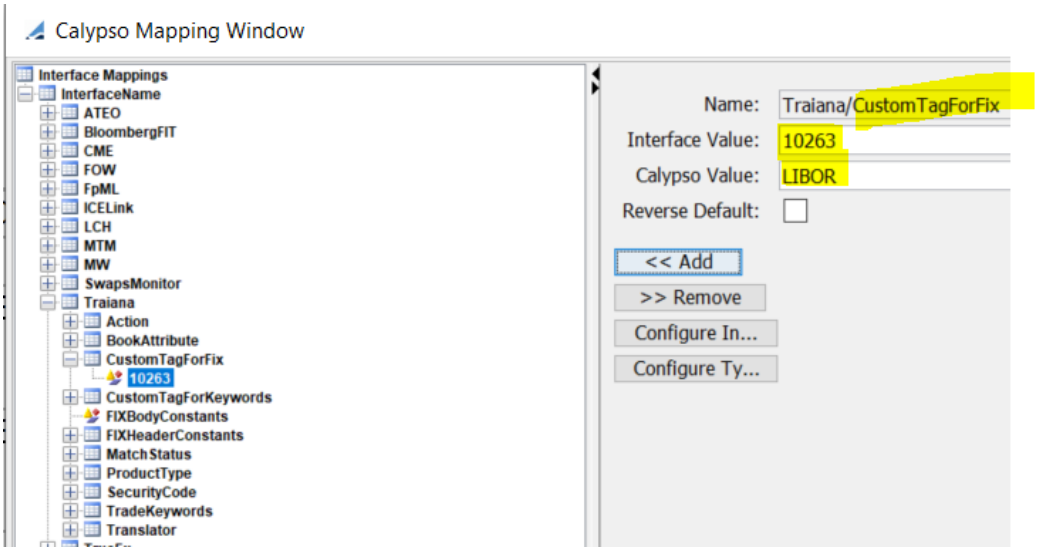
9.2 Fix tags for Mapping Value

In this type we map a Fix tag to a value in following way:

Mapping Type: CustomTagForFix

Interface Value: Fix tag

Calypso value: Valid value



For trades sent to traiana if this mapping is present, its value will be populated in the outgoing fix message.

Note - The mapping types CustomTagForKeywords and CustomTagForFix will be populated as part of Execute SQL, however the calypso and interface value need to be configured by user

Fix tag and its value specified in mapping must be valid and should be identified by Traiana matching platform.

Trade keyword-name must be valid Calypso trade keyword-name.

Test Tool Setup: GUI

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

The Calypso Traiana-matching Interface is built on the Data Uploader framework, and therefore supports uploading Traiana FIX files through the Data Uploader GUI. This can be useful for testing and does not require you to run the Traiana FIX Engine.

The steps below assume you've already setup the Data Uploader module as per the Data Uploader Setup Guide, including adding the GUI window to your menu.

10.1 Setup the GUI Config File

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

To upload Traiana FIX files through the GUI, you will need a properties file with the appropriate settings. A sample file "Traiana-matching-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 Traiana 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.

10.2 Uploading via the GUI

With the previous steps completed, you are now ready to upload Traiana-matching 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'.