Nasdaq Calypso

FCM Trade Interface

Version 16.1 – Version 18

Revision 4.0
December 2022
Approved

Document History

Revision	Published	Summary of Changes
1.0	October 2020	First revision for version 16.1 MR
2.0	July 2021	Updates for version 16.1 MR – Data Uploader version 9.10.0 – Added timezone mapping
3.0	February 2022	Updates for version 17
4.0	December 2022	Updates for version 17

This document describes the FCM Trade Interface.

Table of Contents

Overview

This document outlines FCM Trade Interface that import and processes counterparty Trade feeds into Calypso.

It allows dynamic node and contents mapping with Calypso designed structures. It uses an UBS incoming CP trade feed file as an example.

The FCM trade Interface is designed to support the XML file where the node &contents are under root data & row.

Each counterparty may have XML feed with different node & content details as well as different structures, thus, the user can achieve mappings by customizing the node and contents via Calypso Mapping.

The FCM Trade Interface accepts CP XML file feeds with trade action tag as New/Amend/Cancel and Exercise/Assign lifecycle event trades:

- NewTrade

  External Reference is considered as a Unique identifier to record the New trades in Calypso.

- Amend/Cancel

  To Amend/Cancel any trades, it looks for External Reference and applies the status of the trade, if trade feed details is for amendment then it will update the updated information and if trade feed details is for cancel then it will change the status to cancelled.

- Exercise/Assign

  The FCM Trade Interface processes the action type Exercise/Assign by calling the core API behaviour and it does not allow to record any customized keywords on the default core behaviour.

1.1 Scope

The current version supports XML files which have root contents with data & row.

Buy/Sell – Quantity node value is considered here to select the direction, e.g. if the trade Quantity (1) then it is a Buy direction and if the trade quantity (-1) then it is a Sell direction.

Option Type (Put or Call) is considered from TradeType.

APS trades price is considered from two different trade price nodes (trade price with value QuoteType and trade price with decimal).

Fee date is equal to Settle date is defined from Exchange LE attribute.

1.2 Out of Scope

Support for the XML file with fixml contents may be supported in a later version therefore it is considered out of scope.

Buy/Sell – CP XML file with different node to indicate direction is not supported on current version.

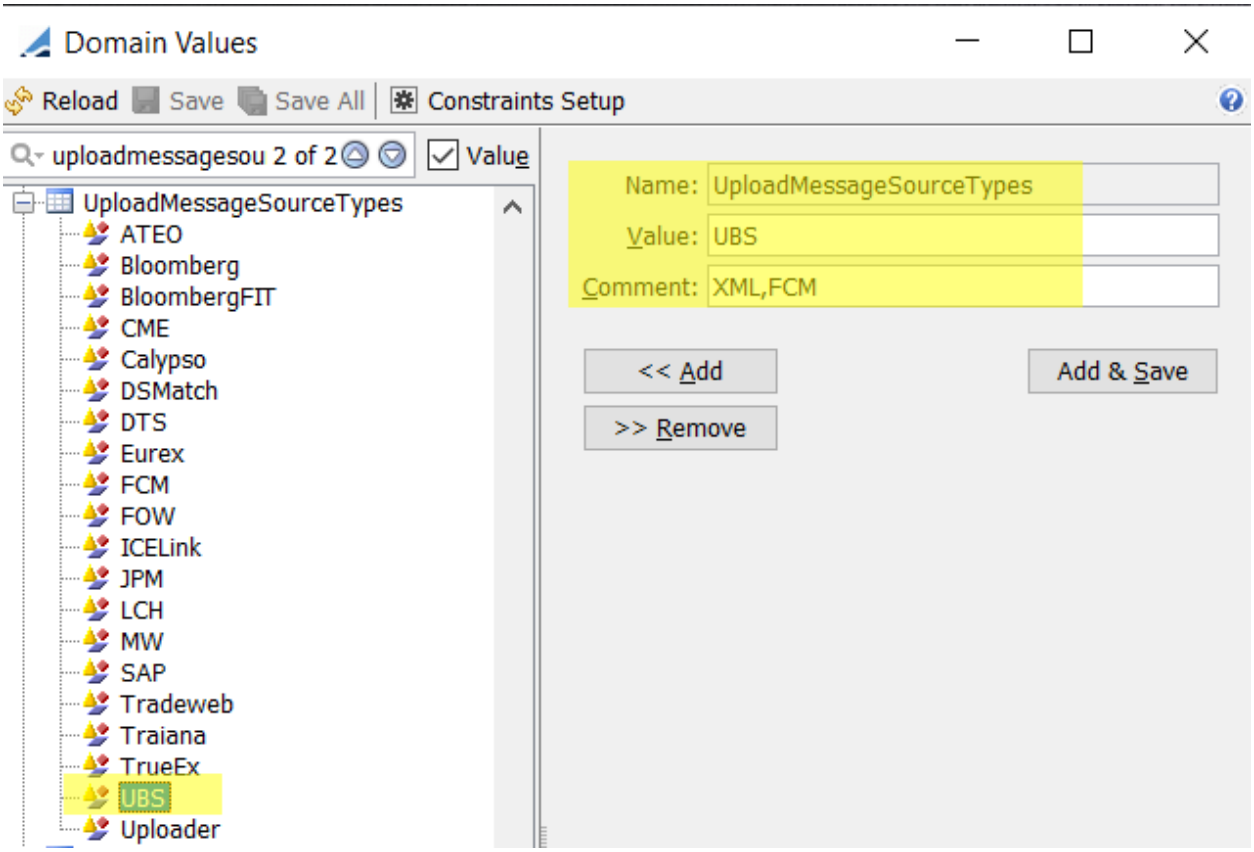
Quantity with Subtype as Short qty or Long qty to indicate additionally direction along with quantity is out of scope.

Option Type (Put or Call) – CP XML file with different node to indicate Optiontype is out of scope,

APS trades – On Current version, APS indicator is designed to capture trade price with combination of two trade price nodes keeping an example of UBS format, if any other counterparty provides XML with one trade price node instead then it is not supported and will be considered on later version.

Setup Requirements

2.1 Domain Values

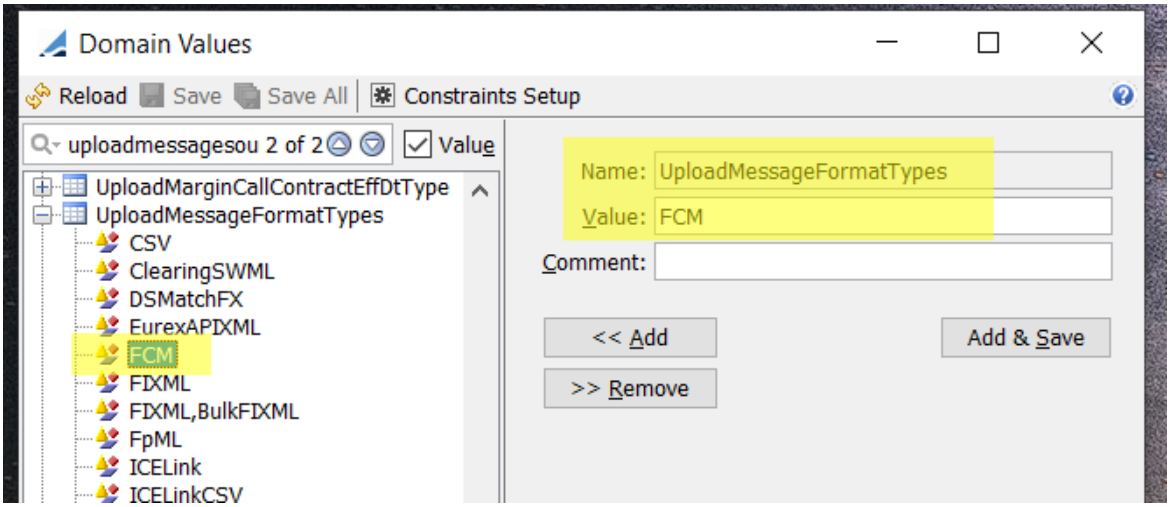


Domain name: UploadMessageSourceTypes

Value: Add value same as Interface name which is created in Calypso Mapping.

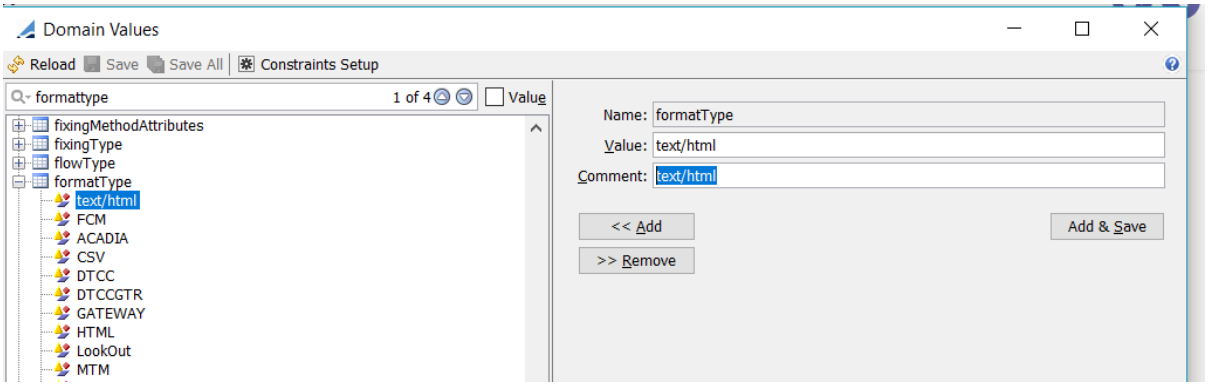
Comment: FCM

Note: If the interface name is already used with some other function, then add the comment with comma separation as given on the above screen shots.

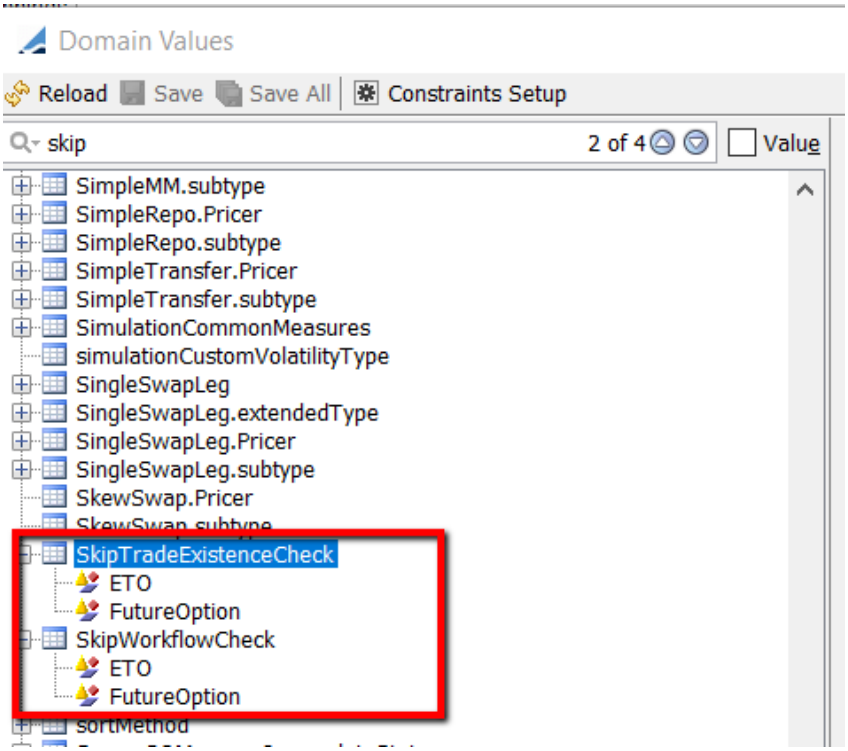


Domain name: UploadMessageFormatTypes

Value: FCM



Add the above domain value and comment, to avoid mime warning in scheduled task status "Finish with Error".



Domain name: SkipTradeExistenceCheck

Value	Comment
ETO	Exercise,Assign
FutureOption	Exercise,Assign

DTUP default design and its behavior for lifecycle event performed on a particular trade wise with External reference, but in ETD clearing lifecycle events are performed on position wise not on trade wise.

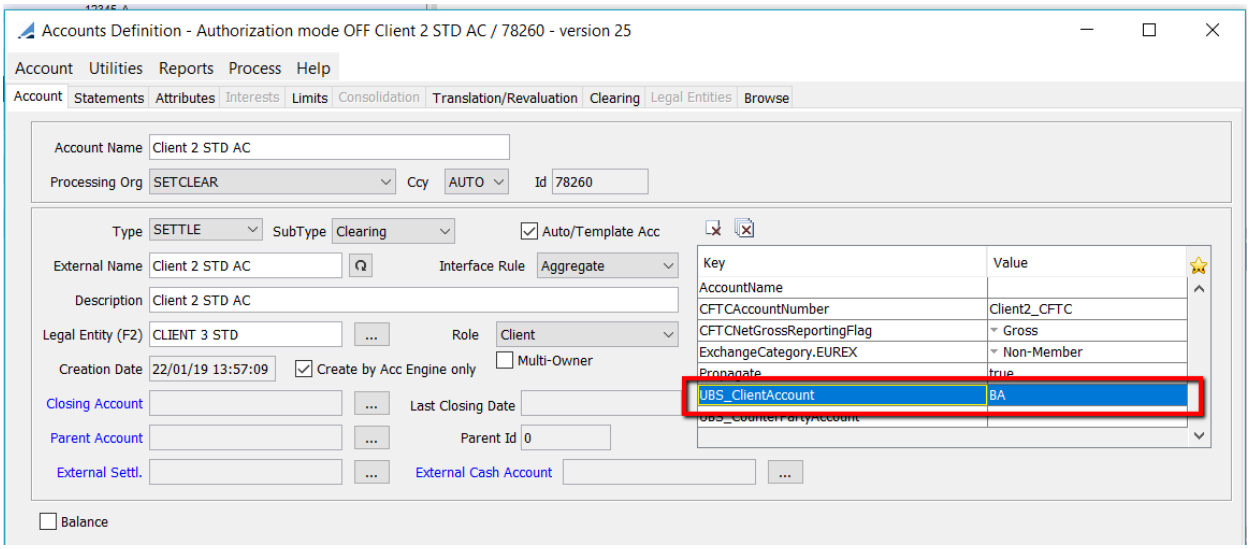
Suppress and bypass the default behavior user need to manually add the above domain values and its comments.

Domain name: SkipWorkflowCheck

Value	Comment
ETO	Exercise,Assign
FutureOption	Exercise,Assign

To process lifecycle events through DTUP module, it checks firstly whether any workflow action for Exercise/Assign exists in workflow, if not it does not allow lifecycle events to flow further. To bypass and ignore those checks, the above domain values and comments need to be added.

2.2 Client Account Configuration



Accounts without account attributes in above format will be ignored and trade will flow to Error account.

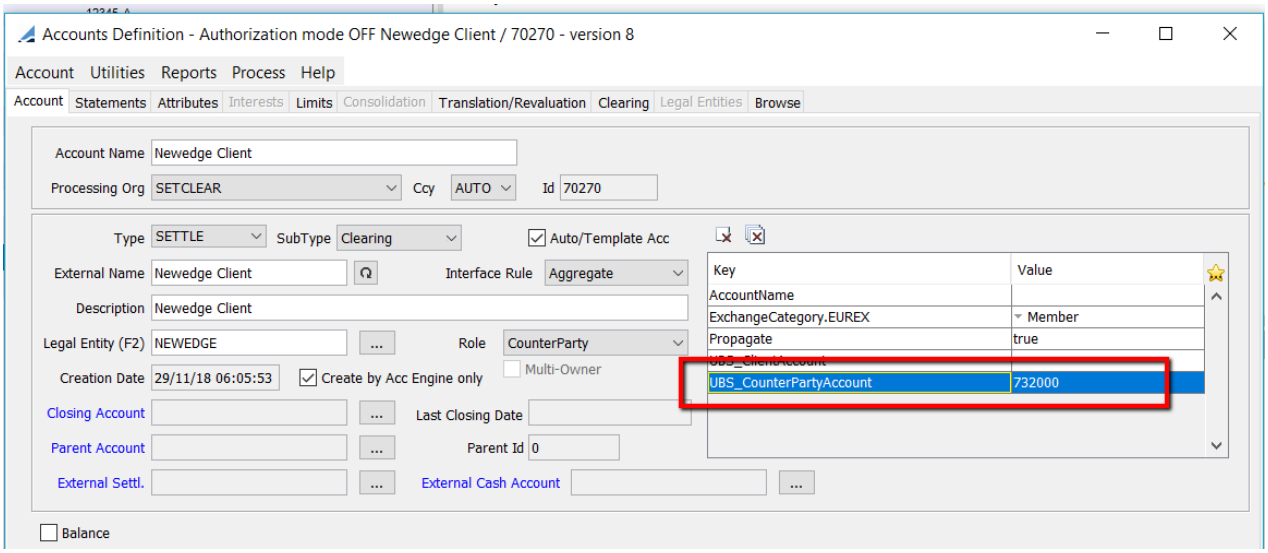
You need to define the account attributes as <interface name>_<attribute>.

For Example:

For interface name = “UBS” and attribute = ClientAccount, create the account attribute UBS_ClientAccount.

2.3 Counterparty Account Configuration

2.3.1 Account Configuration



FCM trade interface primary look is to capture Counterparty account.

Accounts without account attributes in above format will be ignored and trade will flow to Error account.

You need to define the account attributes as <interface name>_<attribute>.

For Example:

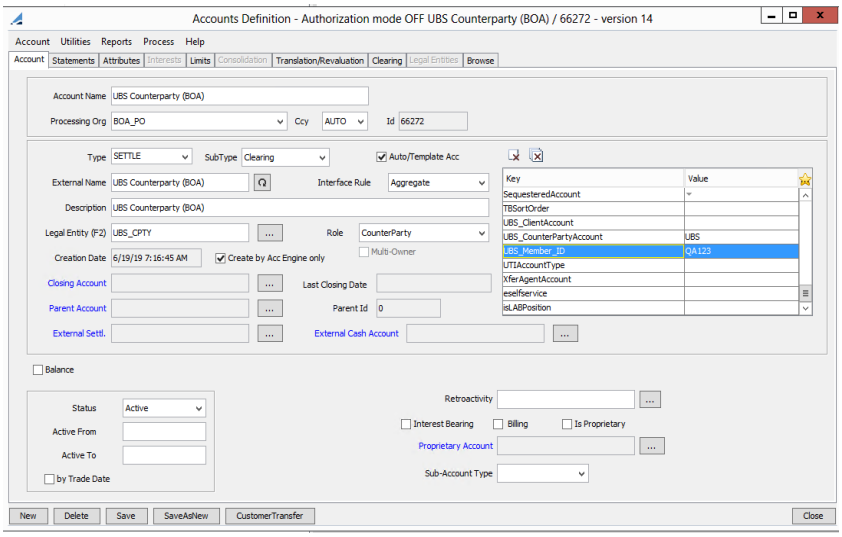
For interface name = "UBS" and attribute = CounterPartyAccount, create the account attribute UBS_CounterPartyAccount.

2.3.2 Member Id Configuration

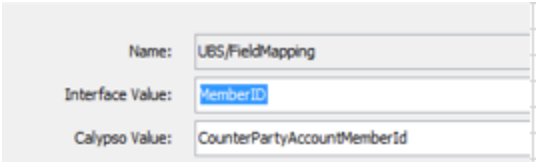
If the XML file receives with Member Id, then in that case, FCM can Configure Member Id to validate from XML file as a primary check likewise counterparty account validation.

> **Note: If FCM do not want Member Id validation then they can ignore the member Id configuration from counterparty account and Field mapping, and it will only validate the primary lookup from counterparty account.**

It can be achieved through the below additional configuration:



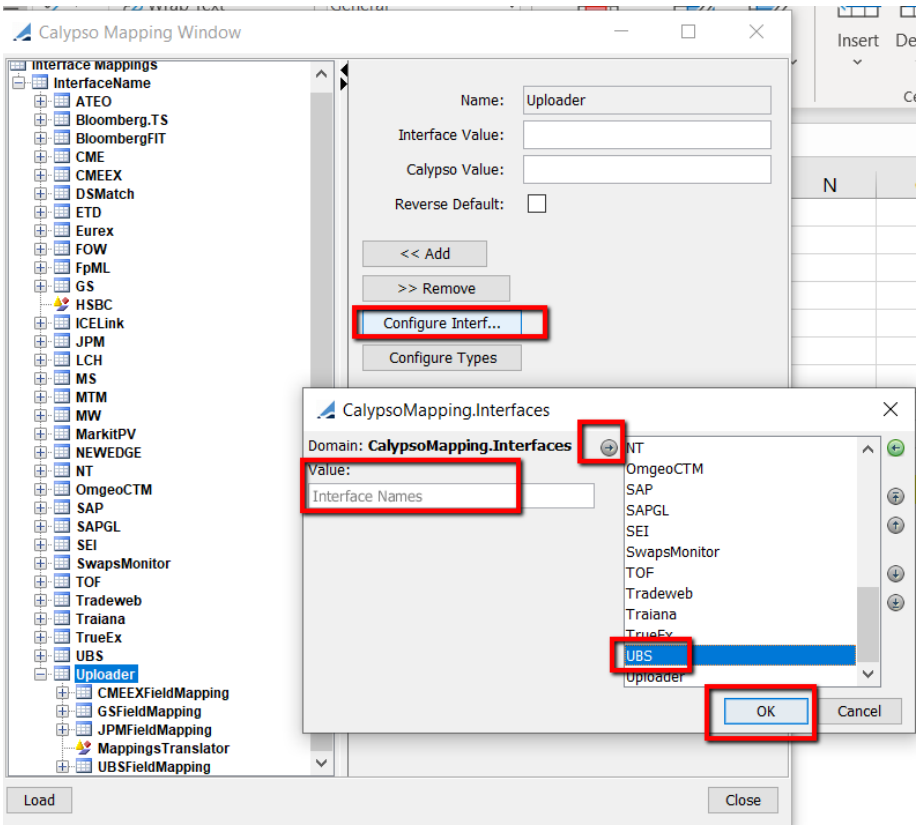
Configure account attribute as <interface name>_Member_ID, for example, "UBS_Member_ID".



Configure FieldMapping as above.

Calypso Mapping

Interface name

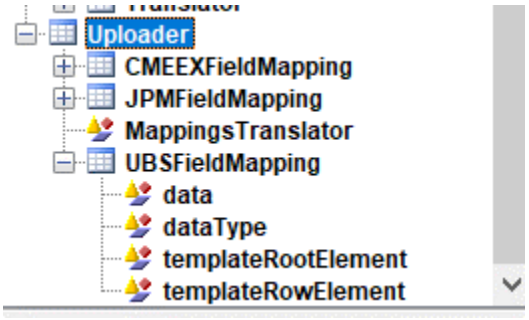


FCM needs to give an identical Interface name to each counterparty in Calypso mapping.

Illustration: We have given “UBS” as an Interface name. You can define any interface name.

3.1 Tag Mapping

Tag mapping is name is created based on Interface name (CP name).



Under “Uploader”, create tag mapping which is configurable as below example.

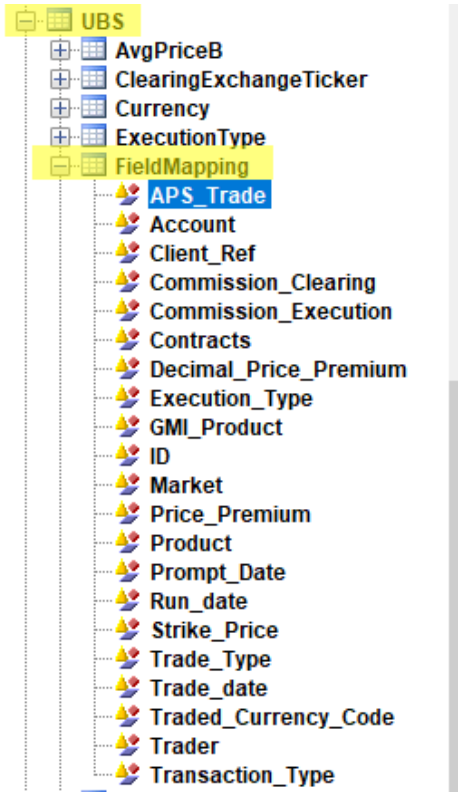
For example: Interface name (CP name) is “UBS" for this case and post prefix “UBS” word to use to create tag mapping is "FieldMapping" i.e. Type name should be in the form of "UBSFieldMapping".

Once above Type is defined, create the below interface and Calypso value inside that type.

Interface Value	Calypso Value
data	XML
dataType	XML
templateRootElement	data
templateRowElement	row

3.2 Field Mapping

Field Mapping to map the input tags coming from Counterparty XML file to that FCMUploader.



Interface Value	Calypso Value
APS_Trade	AvgPriceB
Account	Account

Interface Value	Calypso Value
Client_Ref	ClientRef
Contracts	Quantity
Decimal_Price_Premium	DecimalPricePremium
GMI_Product	GMIProduct
ID	ExternalReference
Market	Exchange
Price_Premium	PricePremium
Product	ClearingExchangeTicker
Prompt_Date	Expiration_Date
Run_date	Run_Date
Strike_Price	StrikePrice
Trade_Type	TradeType
Trade_date	Trade_Date
Traded_Currency_Code	Currency
Trader	TraderName
Transaction_Type	Action

Note: ExternalReference is a Unique Identifier is captured in Calypso with prefix of Interface name with underscore expression.

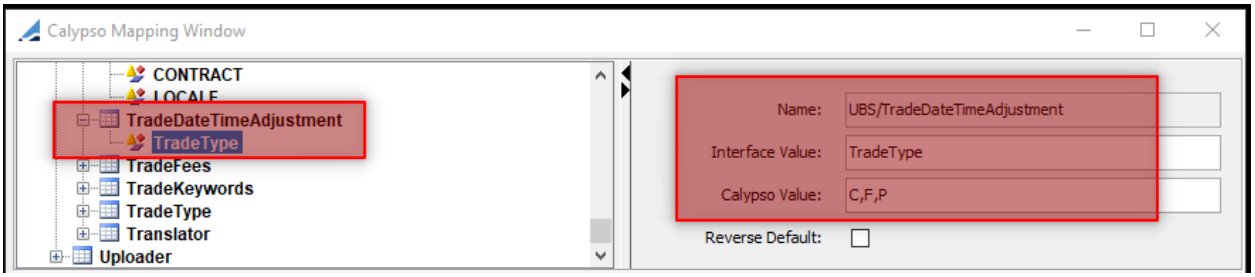
For example: If CP provides the Id as 123456 then it will be expressed in Calypso external reference as "UBS_123456".

Once the CP tags (Interface values) are mapped with Calypso tags(values), the CP value Mapping can be defined under Interface name "UBS" as per Calypso tags value name except "Action" (refer Value mapping).

3.3 Trade Date and Time

UBS provides Trade date and Trade time in different fields.

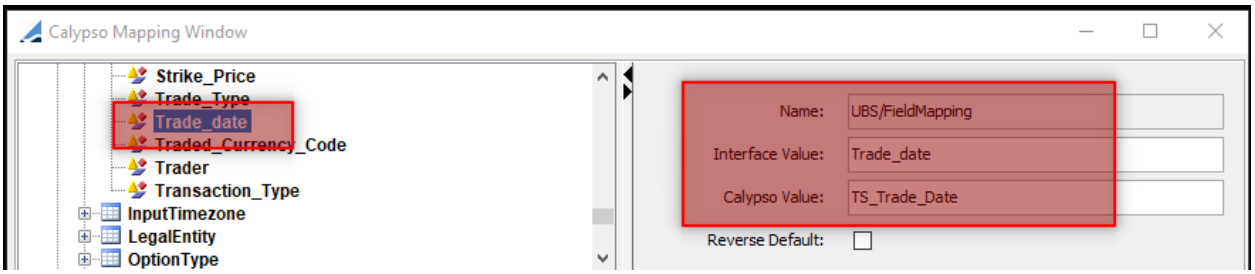
To capture trade with Trade date and Trade time, the following setup needs to be included in the Calypso mapping.

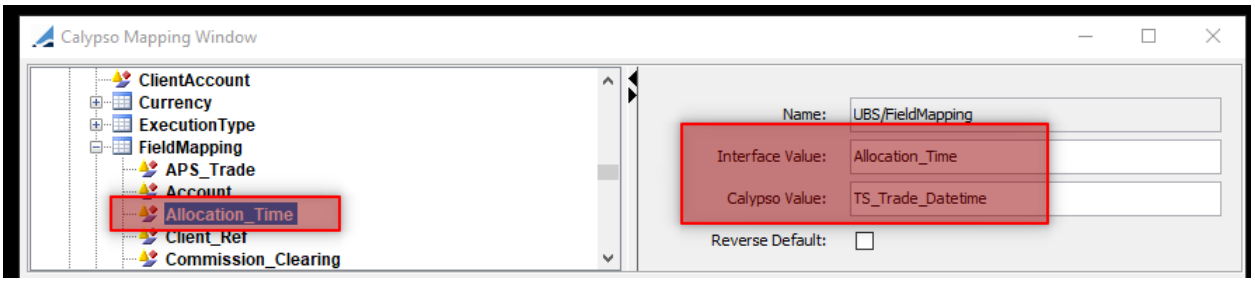


"TradeDateTimeAdjustment" is mapped with specific field value from the trade file.

If a trade is received with this specific field value, the Trade date and time are captured from two different fields. **TS_Trade_Date = Trade date** and **TS(n)_Trade_Datetime = Tradetime**

In field Mapping, Trade date and Allocation time are mapped as follows:





TS_Trade_Date and TS_Trade_Datetime

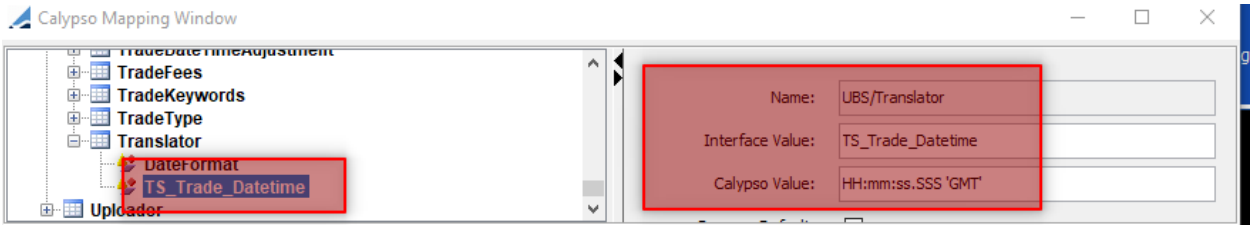
TS_Trade_Date helps capture trade date and time with combination of two fields from UBS XML file. Field Trade_Date is mapped with calypso value TS_Trade_Date, it means the trade date is picked from TS_Trade_Date field and Trade time is picked from TS_Trade_Datetime, TS1_Trade_Datetime, TS2_Trade_Datetime and so on. Based on sequence of version (TS(n)_Trade_Datetime).

When TS(n)_Trade_Datetime is mapped, the XML field date & time value is considered from different fields but also considers TradeCutOffTime logic – See below.

TS_Trade_Datetime

When TS_Trade_Datetime is mapped, the XML field date & time value is considered from same field but without TradeCutOffTime logic.

Translator:



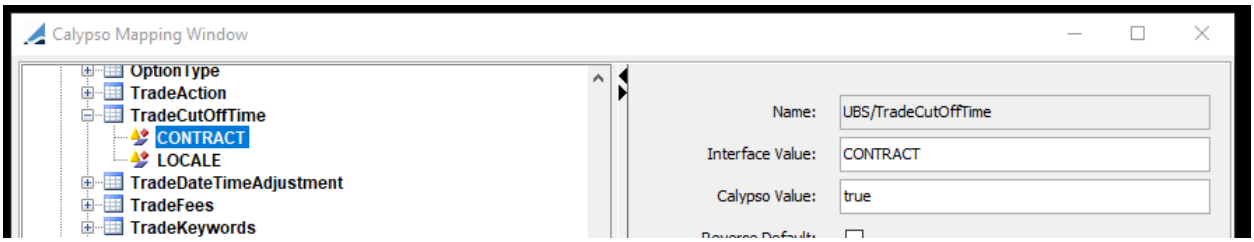
Interface Value TS_Trade_Datetime

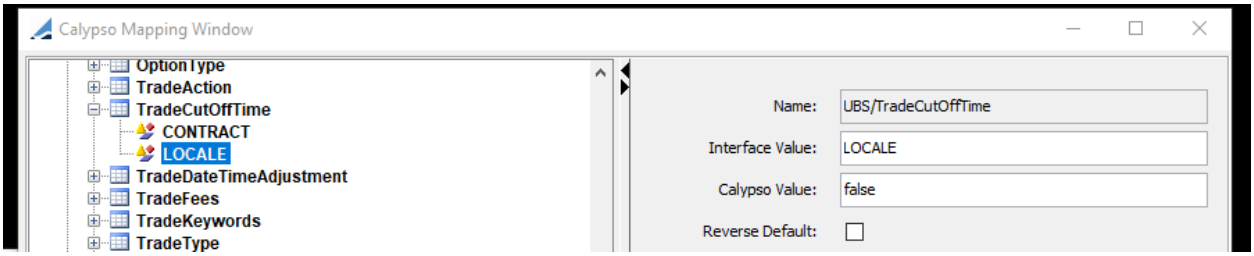
Calypso value HH:mm:ss.SSS 'GMT' (express the time format with Timezone in Translator.)

TradeCutOffTime

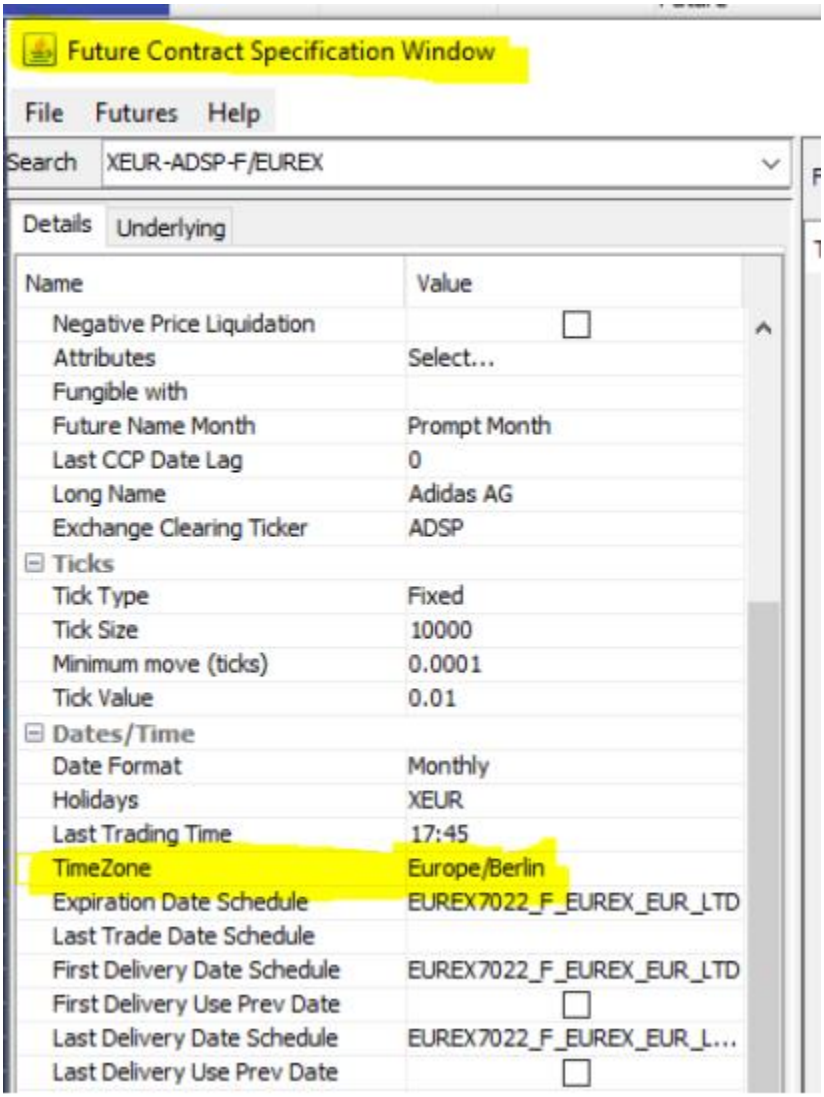
The default time 24 o'clock is considered as market closeout time and if any trade comes with trade date example 25-03-2021 after 24 hours i.e., 00:01 hr then the trade date is automatically moved to the next business day, but the exchange cutoff time is not taken into account.

To take the exchange cutoff time into account, the following configuration needs to be setup.

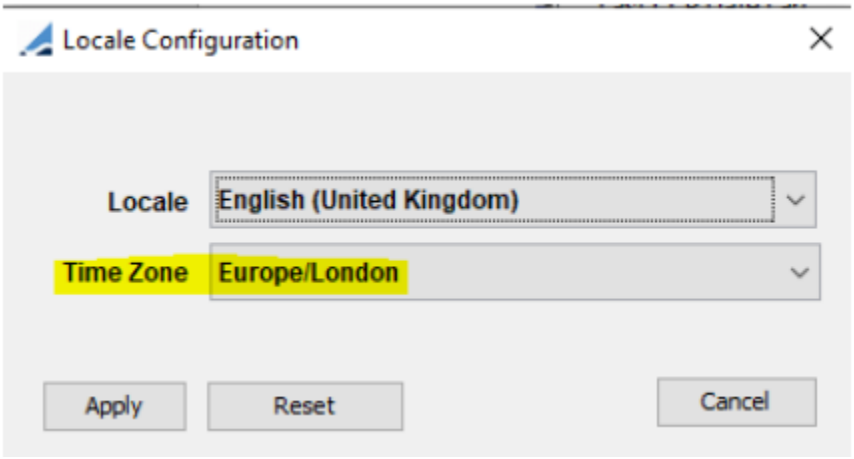




Only one of CONTRACT or LOCALE should be set to true.



In TradeCutOffTime CONTRACT = true, then PO attribute cutoff time is considered as CONTRACT Timezone based time.



If TradeCutOffTime LOCALE = true, then PO attribute cutoff time is considered as LOCALE Configuration Timezone based time.

PO Attribute

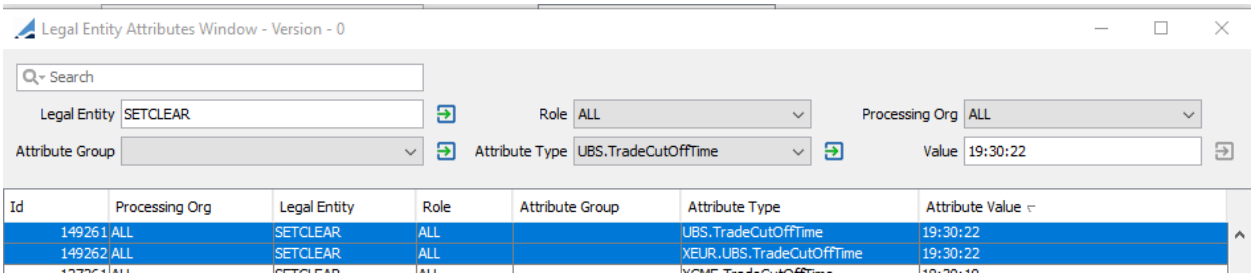
The cutoff time is defined in PO attribute "<exchange MIC>.<interface name>.TradeCutOffTime".

Example: XEUR.UBS.TradeCutOffTime

This cutoff time is considered for the XEUR exchange and UBS interface.

Example: UBS.TradeCutOffTime

This cutoff time is considered for all exchanges for UBS interface, if there is no specified cutoff time for a given exchange.



Attribute value is a time which needs to be expressed in hh:mm:ss – Example: 19:11:30

3.4 Quantity & Direction configuration

3.4.1 Supported Method #1

If the direction i.e. Buy/Sell is indicated in quantity field in CP file in the expression if negative sign in front of -1 & Positive 1 to determine direction Buy/sell.

Then the follow the below configuration in field mapping:

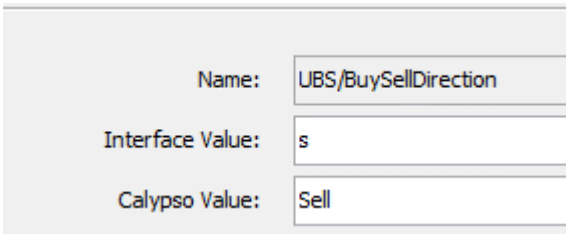
Interface Value	Calypso Value
Contracts	Quantity

3.4.2 Supported Method #2

If the direction i.e. Buy/sell comes in CP files in different tag then ignore the configuration method 1# and configure as below:

Field Mapping:

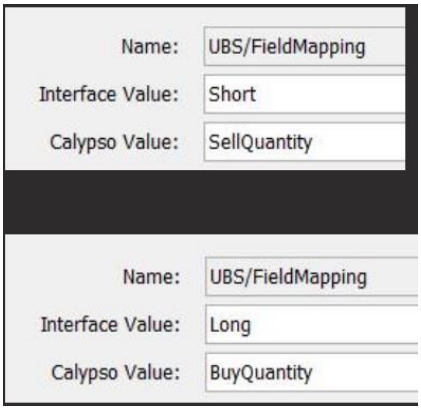
Value mappings:



3.4.3 Supported Method #3

If the direction for Buy and Sell comes in CP files in a different tag then ignore the configuration method #1 & #2 and configure as below:

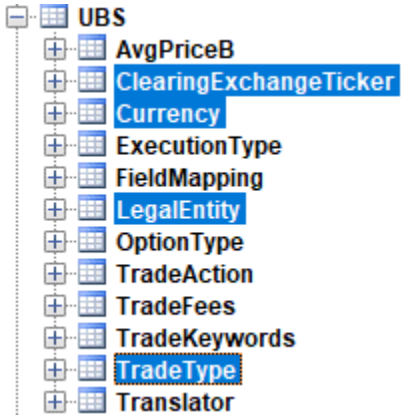
Field Mappings:



3.5 Value Mapping

Value mapping basically used when CP provides value of the particular that is different than the contract setup which is there in Calypso.

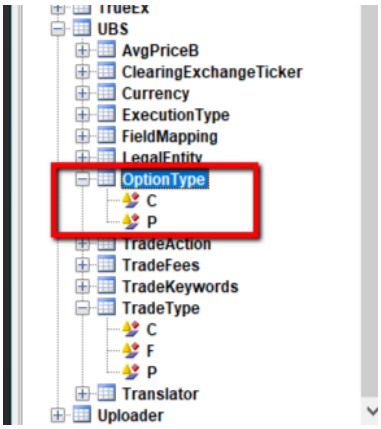
For an example, if CP provide the trade currency as AU and in calypso, we have AUD, in this case, the below mapping feature will enable to convert the value to Calypso.



Description	Example	
	Interface Value	Calypso Value
ClearingExchangeTicker - The symbol of the product code is different than what Calypso contract setup, then through this mapping we can map the value.	EDC	ED
	IRO	IR
Currency - If the currency code is different than the Calypso currency code then user can map the values for those currencies.	AU	AUD
	EU	EUR
LegalEntity - If the Exchange name is different than the calypso exchange name then user can map the value for those particular Exchange code.	XEUREX	EUREX
	XCME	CME
TradeType -TradeType -If CP XML file feed comes with one field as Call & Put for Option and Future and there is	C	Option
	F	Future

Description	Example	
	Interface Value	Calypso Value
no separate field specified for OptionType (Call or Put) then configure only TradeType as follows in right side.	P	Option
If CP XML file feed flows with TradeType Future/Option in one field and separate tag for Option Type call/Put, then additionally configure OptionType under CP node and field mapping.	C	CALL
	P	PUT
For OptionType configuration, follow right side. Note: For ClearingExchangeTicker, Currency & LegalEntity - if the CP and calypso values are same then no need to map values, it will process straight through.	Mapping not required	
	AUD	AUD
	IR	IR
	EUREX	EUREX

3.6 OptionType Mapping



OptionType is an extensive lookup of TradeType to define further Option is a Call or Put.

If CP XML file feed flows with TradeType Future/Option in one field and separate tag for Option Type call/Put, then additionally configure OptionType under CP node and field mapping (ref field mapping section).

3.7 Trade Price Mapping

3.7.1 Supported Method #1

If CP XML file provides Trade Price with QuoteType in separate field and decimal trade price in Different field, then configure as below.

Interface Value	Calypso Value
Decimal_Price_Premium	DecimalPricePremium
Price_Premium	PricePremium

If Average price flag is TRUE then the price will be picked from DecimalPricePremium if FALSE then PricePremium.

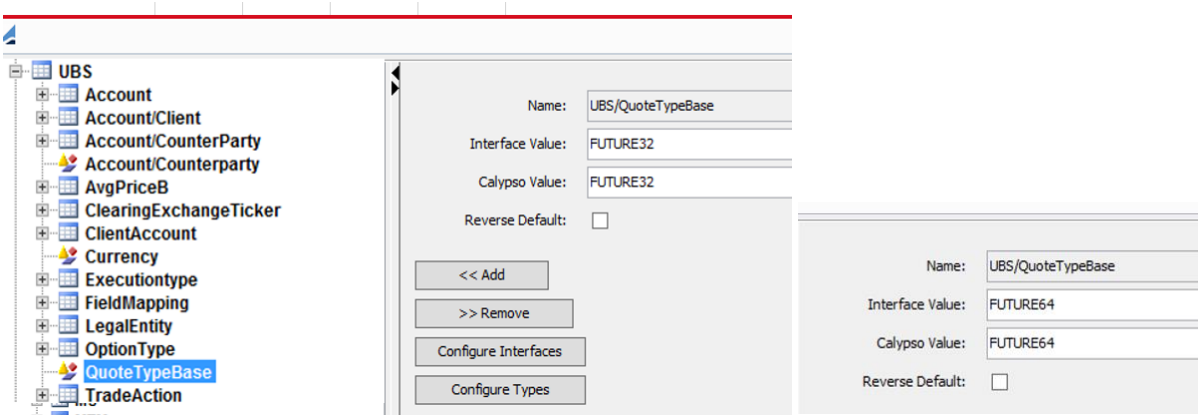
3.7.2 Supported Method #2

If CP XML file provides Trade Price for decimal and QuoteType price in single field then configure only PricePremium.

Interface Value	Calypso Value
Price_Premium	PricePremium

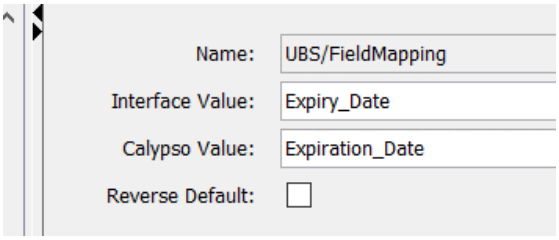
3.8 QuoteTypeBase Mapping

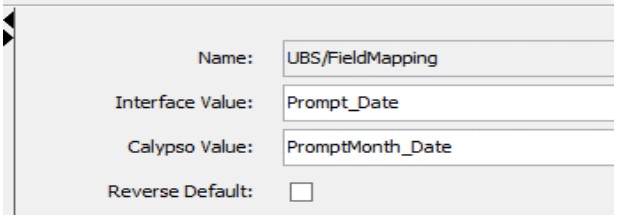
If the Trade price flows in decimal for QuoteType Price and for Future32/64 flows in QuoteType Future32/64 i.e. in same format with decimal separation, then configure as below to capture trade price as future32/64 by replacing dot (.) to Hyphen (-).



3.9 Expiry Date and Prompt Month Validation

The primary lookup will be done in Expiry date configuration, if it is not configured in field mapping or expiry date is mapped in Field mapping and there is no field value in CP XML file then Prompt month date will be looked up to pull the contract and its product.

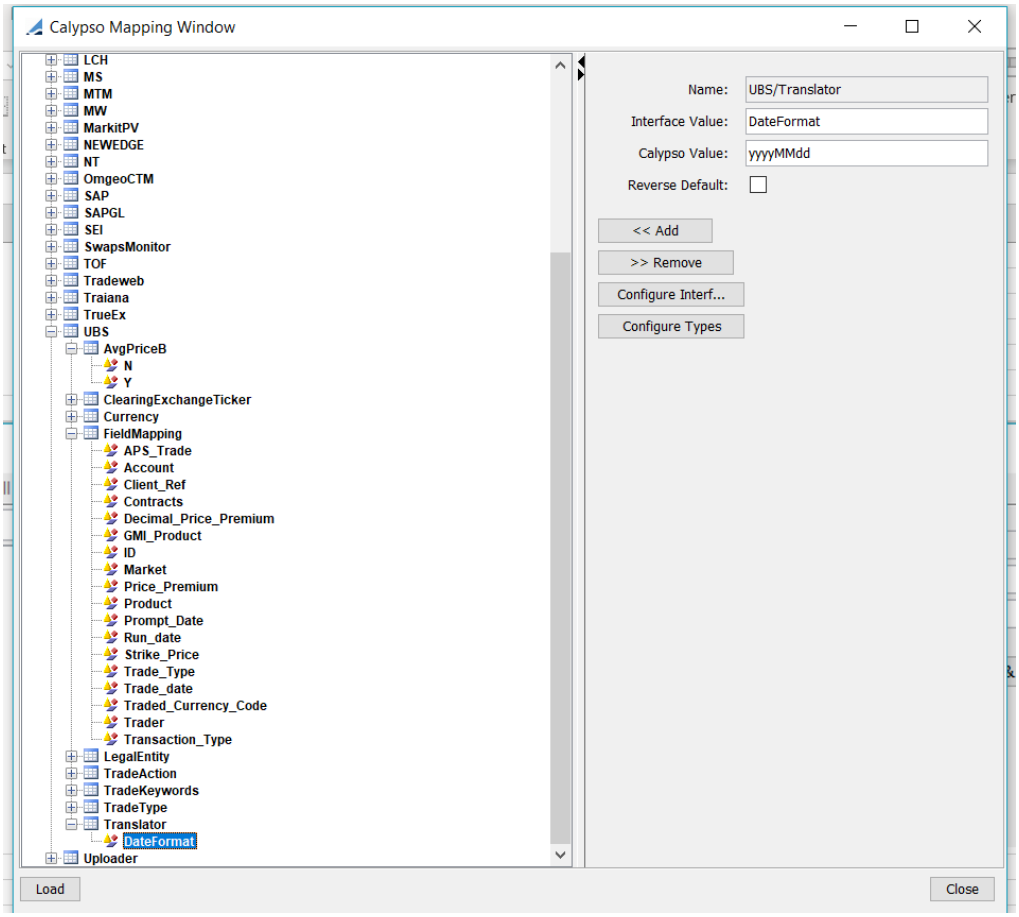




3.10 Date Format Mapping

Each CP XML file's date format may vary, Example if CP provide the date format as "yyyyMMdd" or "MMddyyyy" etc. below mapping will allow to reconstruct that format to calypso value.

Map Type name as "Translator" under Interface name.



The following formats are supported:

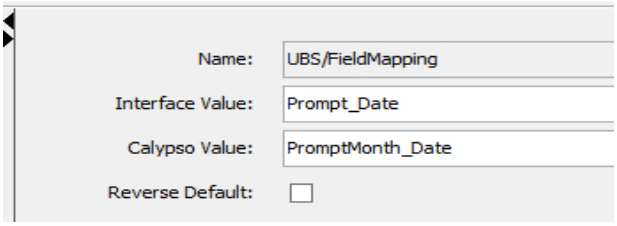
- MMddyyyy
- MMyyyydd
- ddMMyyyy
- ddyyyyMM

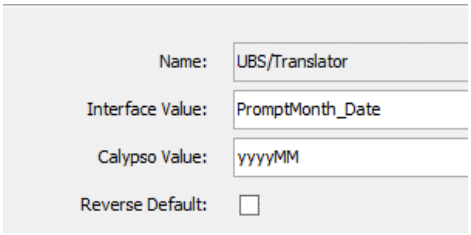
- yyyyMMdd
- yyyyddMM
- MMMddyyyy
- MMMyyyydd
- ddMMMyyyy
- ddyyyyMMM
- yyyyMMMdd
- yyyyddMMM

Note:

- MM & MMM - month should be in uppercase
- dd & yyyy – day and year should be in lowercase

If particular field has different format than the rest of the date fields, then configure that specific fields date formatting as below:

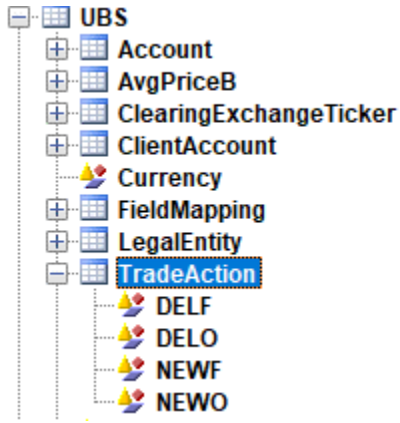




3.11 TradeAction

It indicates the action to perform on trade. You can select a different action as needed; it will be applied to the trade upon saving below mapping.

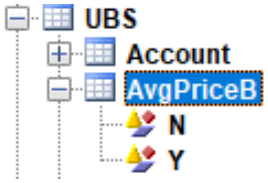
Map Type name as “TradeAction” under Interface name.



Interface Value	Calypso Value
NEWF	NEW
NEWO	NEW
DELF	CANCEL
DELO	CANCEL
ASSN	ASSIGN
EXER	EXERCISE
NEWFO	AMEND

3.12 AvgPriceB

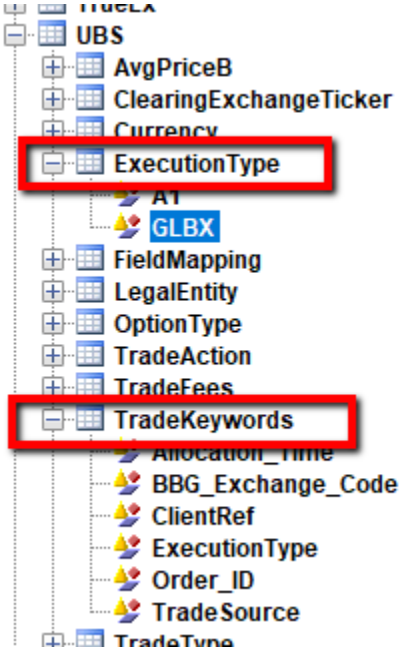
The below mapping allow trade to capture with or without average price flag in trade.



Interface Value	Calypso Value
N	FALSE
Y	TRUE

3.13 Trade Keywords

Trade keywords can be abbreviated to store the trade attributes.

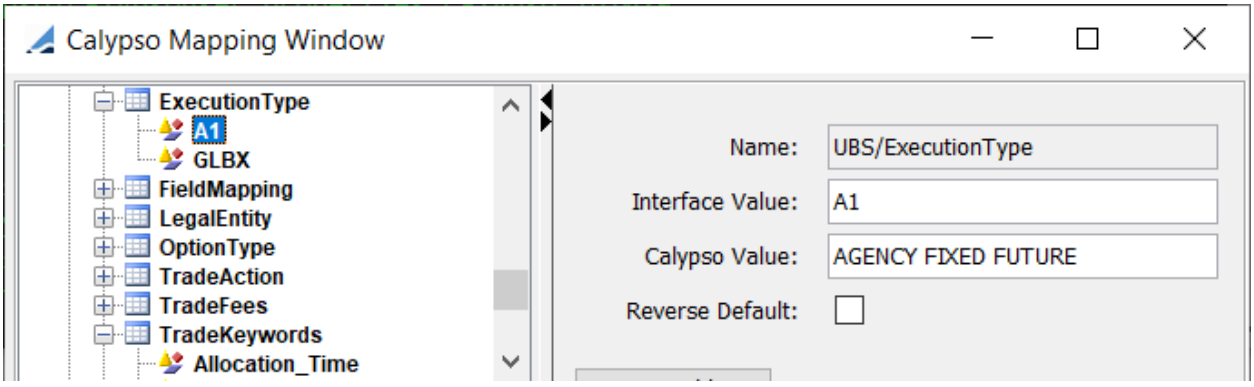


Note: Currently Trade keywords are stored only for new/amend/cancel trades and not on lifecycle event trades.

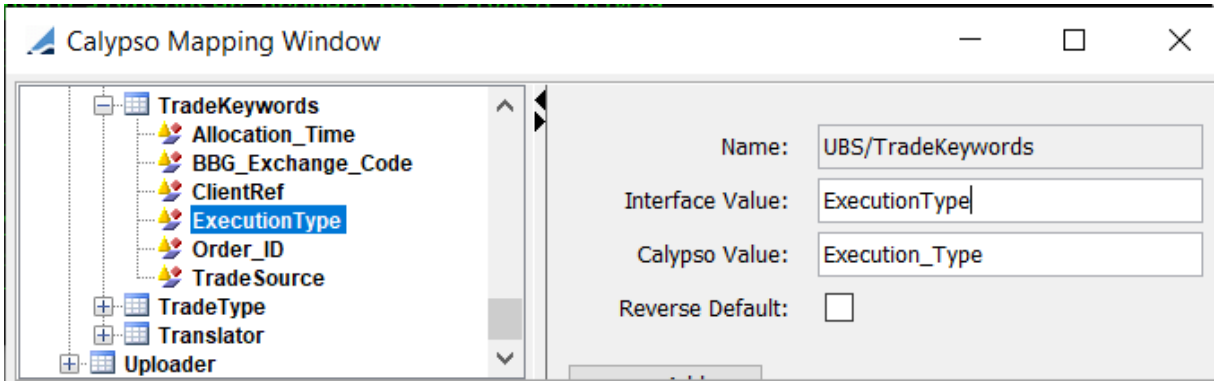
TradeSource – The default name is used to store in TradeSource is Interface name "UBS" and one can also change the default name to new value by inputting new value on TradeSource in calypso value field.

ExecutionType - If user want a keyword to be stored with the abbreviation, then map the TradeKeywords's Interface value under Interface name "UBS".

Example: "ExecutionType" is created under Interface name "UBS" as well as inside TradeKeywords.

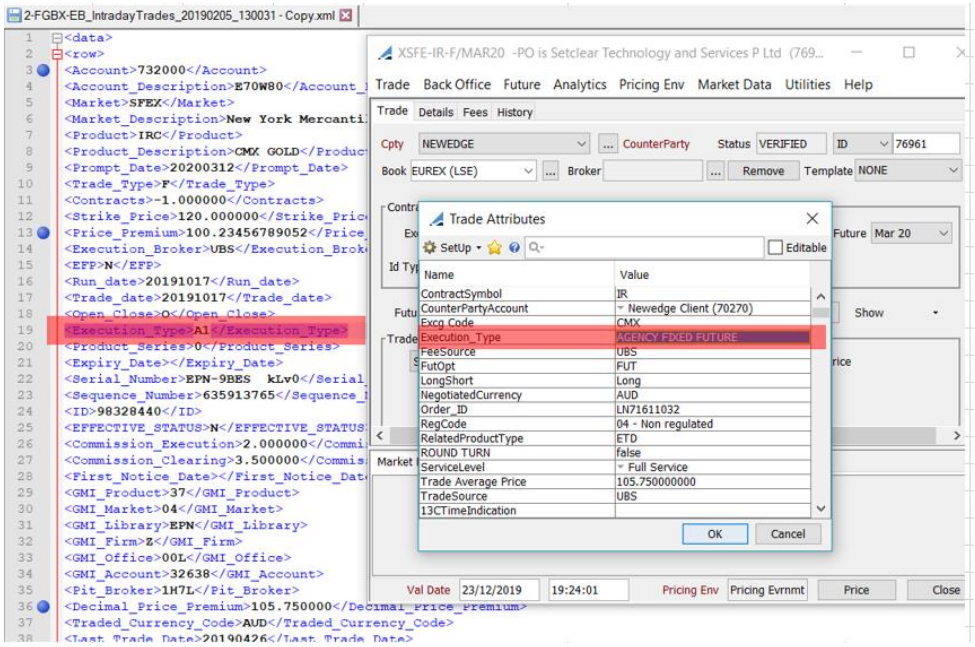


- Under Interface name, the ExecutionType is used to store UBS XML file value short name will be abbreviations.



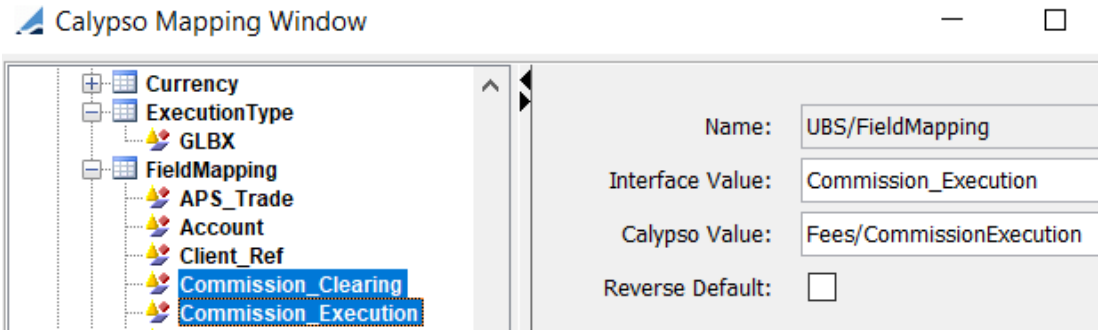
- Under TradeKeywords, store the interface value same as which is created under Interface name such as "ExecutionType" and keep the Calypso value same as Interface value or change it to other prefer Calypso value to store in trade keywords.

  See the below snapshot as how the trade keywords stored with the above mappings.



3.14 Fees Mapping

Field Mapping:



Interface Value	Calypso Value
Commission_Execution	Fees/CommissionExecution
Commission_Clearing	Fees/CommissionClearing

NOTE: Calypso Value should be prefixed with "Fees/" system to identify these field mappings are related to fees.

User can give any Calypso value after Prefixed with “Fees/” to specific the type of fees and which are needed to capture from CP XML files.

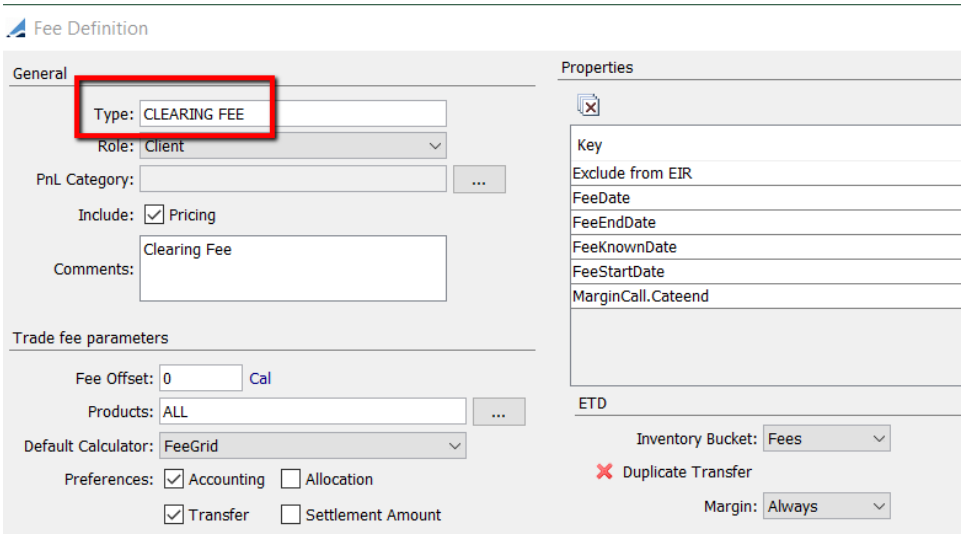
On the above example we have used “Commission_Execution” and “Commission_Clearing” fees in order to capture on trades.

Create new type “TradeFees” under Interface name “UBS”.

Interface Value	Calypso Value
CommissionClearing	CLEARING_FEE/CLEARING FEE
CommissionExecution	EXCHANGE_FEE/EXCHANGE FEE

Under “TradeFees”, create the above mappings.

Interface value should be same as field mapping for “Commission_Execution” & “Commission_Clearing” with Calypso value defined as “Fees/<fee>”.



Calypso value should be Fee definition's Type value.

Calypso value is separated with "/" to define two fee definitions.

Fee definition before "/ " should be defined where fee definition is created with CP role. and Fee definition after "/ " should be defined where fee definition is created with Client role.

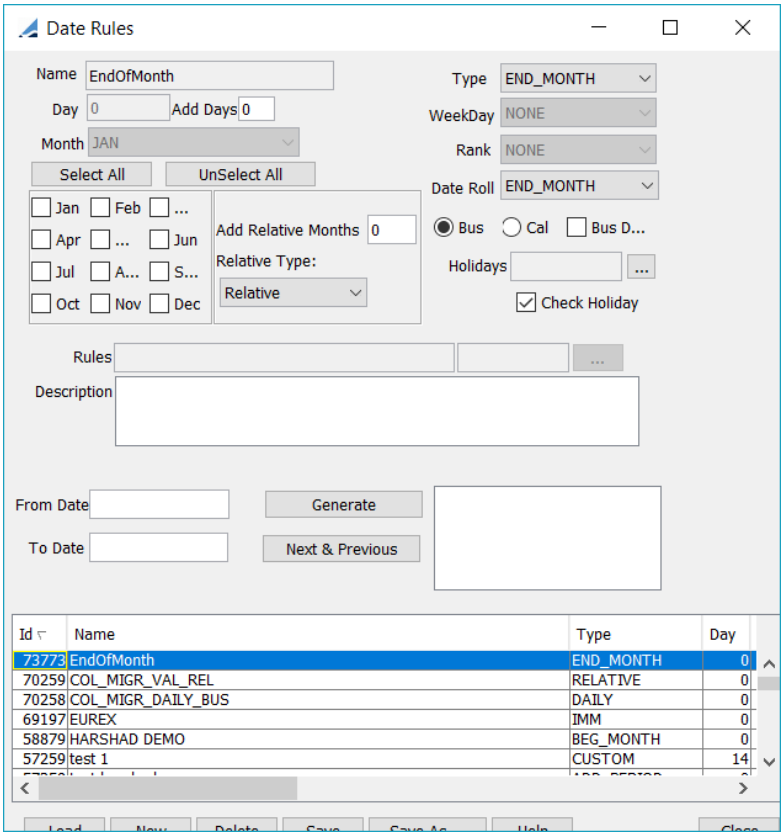
Interface Value	Calypso Value
CLEARING FEE	Trade Date
CLEARING_FEE	EndOfMonth
EXCHANGE FEE	Settle Date

Under "TradeFees", create the above mappings.

These mapping is required to specific what date to capture on fee, Start & End date in trades.

Each Fee definition should be mapped as which fee date to be used.

Trade Date – it will directly pick the date from trade.



EndOfMonth – Date rules name.

Settle Date – In ETD, we have dedicated domain and configuration where the settlement date is used to calculate.

Value = ETDClearing.SettlementLag

The same domain is used to capture the settlement date for fees.

If Default value is true – setting it to true would be the same as being empty. Setting it to false would disable the lag. This domain allows to work with Settle Date = Trade Date + 1 Bus Day on Future & Options transactions and Clearing Transfer for statement purposes

Settle Date of transaction is computed by the system adding a default of 1 Business Day lag (using contract exchange calendar).

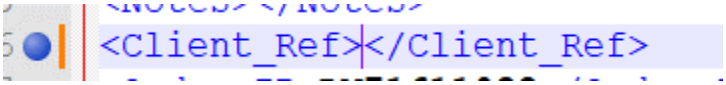
If a different lag must be used by currency, we refer to the currency attribute ClearingTransferSettleLag.

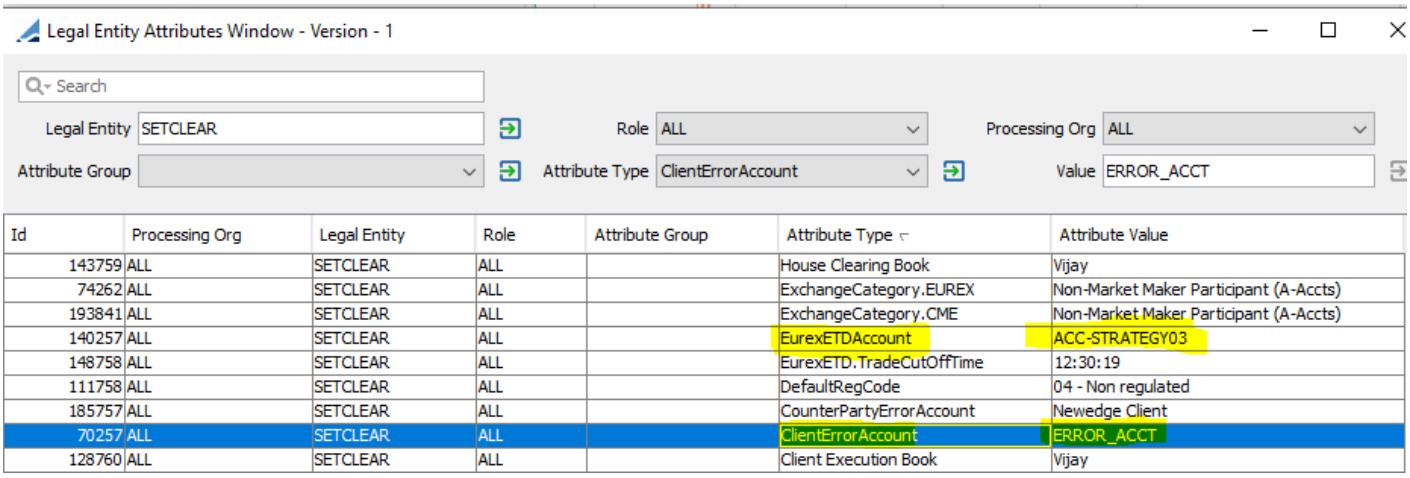
This attribute can also be defined per Counterparty (CCP or broker) by using the LE Short Name + ClearingTransferLag attribute.

For example, if you have to apply a 2D lag for PLN, except when you clear PLN with BROKER1 (BROKER1 being the shortname of your LE), you will define ClearingTransferSettleLag = 0 for PLN and another PLN attribute BROKER1ClearingTransferSettleLag that would be set to 0.

ClientErrorAccount

If Client_Ref is not present in the XML file, the PO attribute ClientErrorAccount is used for the account.

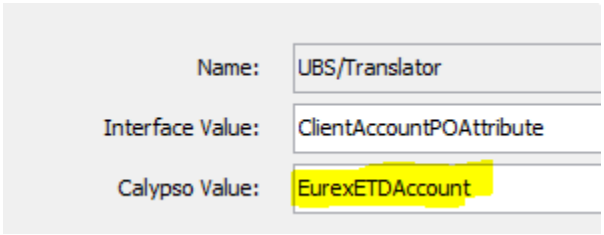




Id	Processing Org	Legal Entity	Role	Attribute Group	Attribute Type	Attribute Value
143759	ALL	SETCLEAR	ALL		House Clearing Book	Vijay
74262	ALL	SETCLEAR	ALL		ExchangeCategory.EUREX	Non-Market Maker Participant (A-Accts)
193841	ALL	SETCLEAR	ALL		ExchangeCategory.CME	Non-Market Maker Participant (A-Accts)
140257	ALL	SETCLEAR	ALL		EurexETDAccount	ACC-STRATEGY03
148758	ALL	SETCLEAR	ALL		EurexETD.TradeCutOffTime	12:30:19
111758	ALL	SETCLEAR	ALL		DefaultRegCode	04 - Non regulated
185757	ALL	SETCLEAR	ALL		CounterPartyErrorAccount	Newedge Client
70257	ALL	SETCLEAR	ALL		ClientErrorAccount	ERROR_ACCT
128760	ALL	SETCLEAR	ALL		Client Execution Book	Vijay

For exercise / assignment - When actual Trades are retrieved from TOQ then internally, the fees are split by each actual client account.

You can also configure the following mapping to retrieve the error account from a different PO attribute:



Calypso Value = <PO attribute> - The default PO attribute is ClientErrorAccount.

NOTE: If the workflow rule AutomaticFees is specified in the Trade workflow, the New/Amend trade doesnot capture fees from XML file.

3.15 Timezone

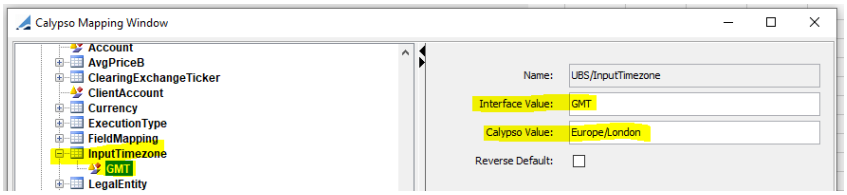
Import timezone from FCM files using the following mapping:

UBS/InputTimezone

Interface Value = GMT

Calypso Value = Europe/London

If not set, the contract timezone is used.



Supported timezone formats are +00:00, GMT and -00:00

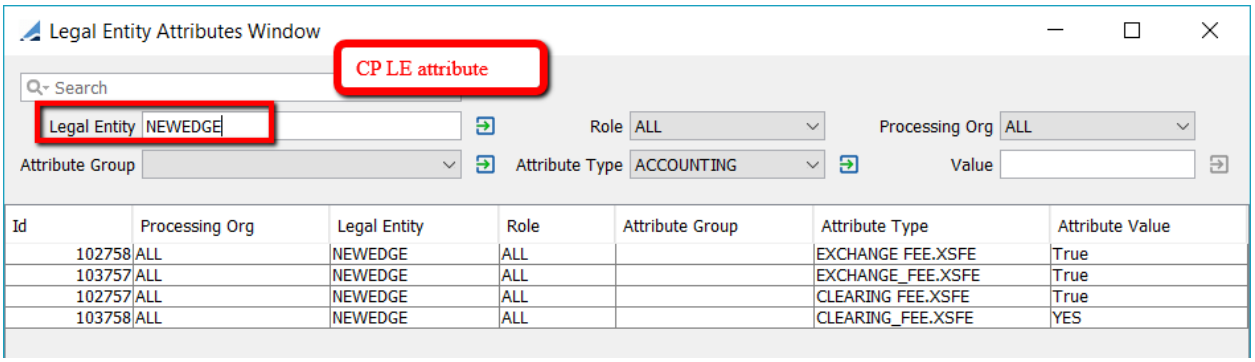
Timezones considered while implementing this feature are:

- UBS <Allocation_Time>12:41:55.204000000 GMT</Allocation_Time>
- CME TxnTm="2020-02-20T00:00:00-06:00"
- EUREX <TrdRegTS TS="2021-04-13T13:32:53.449+00:00" Typ="2"/>

NOTE: The above FCM and CCP files timezones are tested extensively. Any new timezone format will be treated as an additional requirement.

3.16 Counterparty LE attribute for Fees Mapping

The objective to obtain LE level mapping is only to consider the fee definition & exchange which are mapped to compute fees from CP file or else calc calypso value.



Id	Processing Org	Legal Entity	Role	Attribute Group	Attribute Type	Attribute Value
102758	ALL	NEWEDGE	ALL		EXCHANGE FEE.XSFE	True
103757	ALL	NEWEDGE	ALL		EXCHANGE_FEE.XSFE	True
102757	ALL	NEWEDGE	ALL		CLEARING FEE.XSFE	True
103758	ALL	NEWEDGE	ALL		CLEARING_FEE.XSFE	YES

Attribute Type should be the combination of Fee definition Type value and dot (.) and followed by Exchange MIC code. Refer the above screen sort.

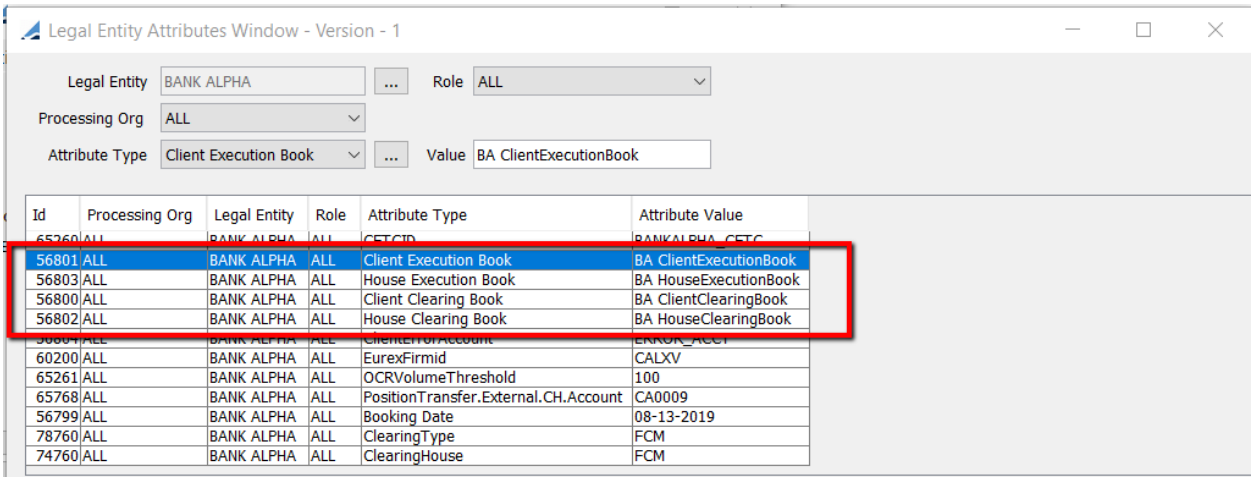
The attribute values are expressed in “True” or “False”.

‘True’ is to consider the fees from CP XML file and ‘False’ not to consider the fees from CP XML for particular exchange.

3.17 Book Mapping

The book is selected based on the configuration of the clearing account attribute setup “Clearing Book”. This is the default behavior. If an Account has a valid Book in the ‘Clearing Book’ value, then trades should be captured into that book.

If no value is present in the ‘Clearing Book’ attribute of the account, the book will be chosen from PO LE attributes by looking at the Subtype and Origin Code of the Client Account and following the logic below:

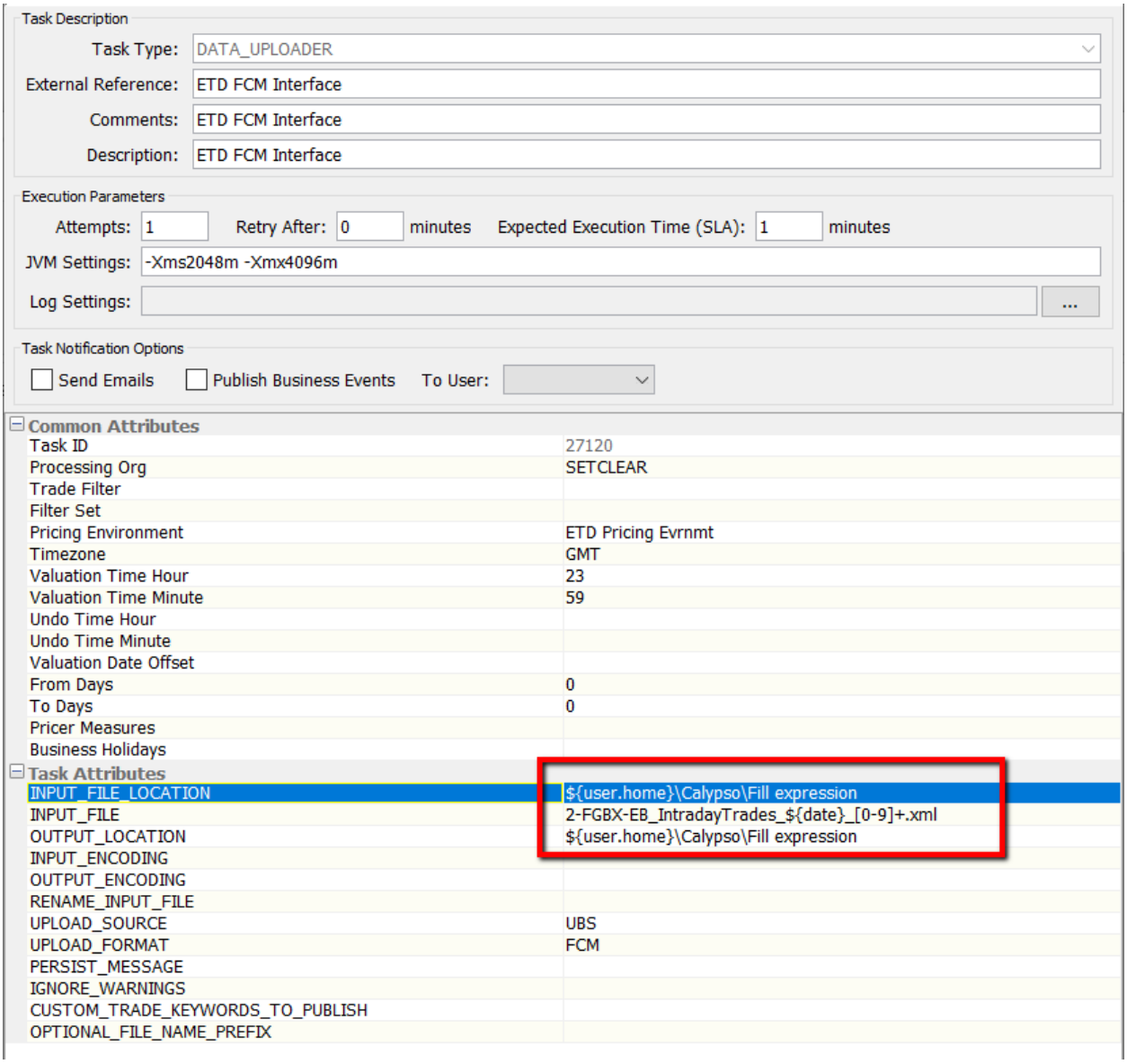


Id	Processing Org	Legal Entity	Role	Attribute Type	Attribute Value
56801	ALL	BANK ALPHA	ALL	Client Execution Book	BA ClientExecutionBook
56803	ALL	BANK ALPHA	ALL	House Execution Book	BA HouseExecutionBook
56800	ALL	BANK ALPHA	ALL	Client Clearing Book	BA ClientClearingBook
56802	ALL	BANK ALPHA	ALL	House Clearing Book	BA HouseClearingBook
60200	ALL	BANK ALPHA	ALL	EurexFirmid	CALXV
65261	ALL	BANK ALPHA	ALL	OCRVolumeThreshold	100
65768	ALL	BANK ALPHA	ALL	PositionTransfer.External.CH.Account	CA0009
56799	ALL	BANK ALPHA	ALL	Booking Date	08-13-2019
78760	ALL	BANK ALPHA	ALL	ClearingType	FCM
74760	ALL	BANK ALPHA	ALL	ClearingHouse	FCM

- Client Execution Book – Client account definition subtype (Execution), Account attribute “AccountType” (Client)
- House Execution Book – Client account definition subtype (Execution), Account attribute “AccountType” (House)
- Client Clearing Book – Client account definition subtype (Clearing), clearing tab Origin code (Client)
- House Clearing Book – Client account definition subtype (Clearing), clearing tab Origin code (House)

Process

4.1 Scheduled Task Configuration



One DATA_UPLOADER scheduled task instance must be configured for each reportable PO.

The following file expressions are supported:

${User.home} – <user home> folder.

${date} – Each CP file may be identified by date and it can be defined through this configuration.

If the scheduled task's valuation date is equal to CP file date, then it will directly pick those files to process and no need to configuration dates manually each time.

[0-9]+ or [0-9]{9} – Each CP file may be identified by time stamp or by sequence number when trade file is generated and it can be defined through this configuration.

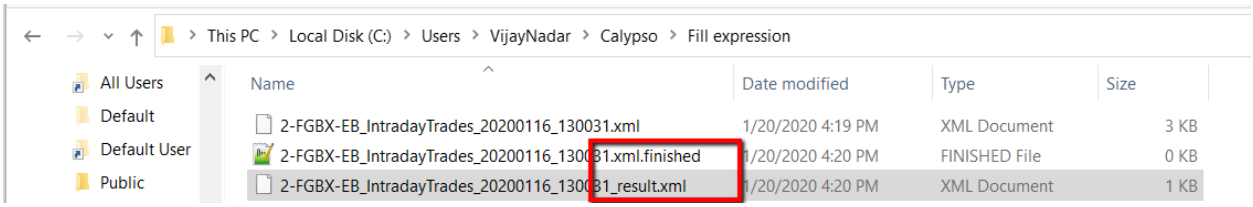
+ sign is to call infinite digits if it exceeds above 9 digits.

{9} to consider files expression only with nine digits and if it exceeds above nine digits then it will not consider those files to process.

For example:

If CP provides file with timestamp up to 6 digits or sequence number up to 6 digits then the scheduled task should be configured as [0-6]{6} and if timestamp or sequence number exceeds 6 digits then it should be configured as [0-9]+.

4.2 Scheduled Task Output



The scheduled task generates two XML representation which ends with 'finished' and '_result.xml'

'.finished' this XML file created with blank content to indicate the specific CP file is processed or not.

'_result.xml' this XML file is created with status of the trades and this file also recorded with exceptions if any.

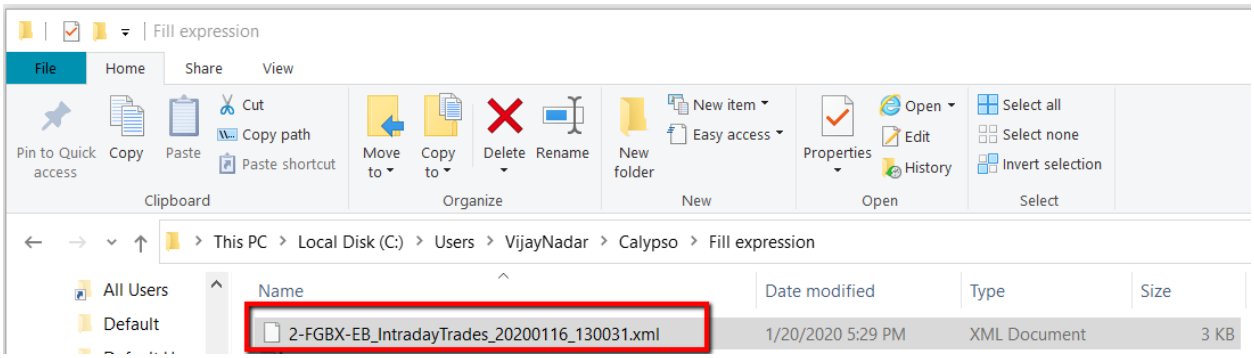
4.3 Use Case

To illustrate, we have created the below sample trade file and stored it in the local drive.

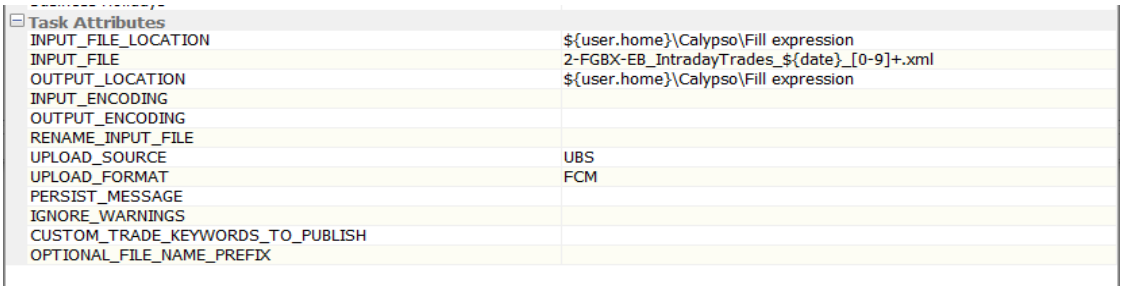
```
<data>
<row>
<Account>UB32000</Account>
<Account_Description>E70W80</Account_Description>
<Market>CBOT</Market>
<Market_Description>New York Mercantile Exchange</Market_Description>
<Product>17</Product>
<Product_Description> 30 year US Treasury Bonds </Product_Description>
<Prompt_Date>20200221</Prompt_Date>
<Trade_Type>C</Trade_Type>
```

```
<Contracts>111.000000</Contracts>
<Strike_Price>100.000000</Strike_Price>
<Price_Premium>100.23456789052</Price_Premium>
<Execution_Broker>UBS</Execution_Broker>
<EFP>N</EFP>
<Run_date>20200116</Run_date>
<Trade_date>20200116</Trade_date>
<Open_Close>O</Open_Close>
<Execution_Type>A1</Execution_Type>
<Product_Series>0</Product_Series>
<Expiry_Date></Expiry_Date>
<Serial_Number>EPN-9BES  kLv0</Serial_Number>
<Sequence_Number>635913765</Sequence_Number>
<ID>78328500</ID>
<EFFECTIVE_STATUS>N</EFFECTIVE_STATUS>
<Commission_Execution>-2.000000</Commission_Execution>
<Commission_Clearing>-3.500000</Commission_Clearing>
<First_Notice_Date></First_Notice_Date>
<GMI_Product>37</GMI_Product>
<GMI_Market>04</GMI_Market>
<GMI_Library>EPN</GMI_Library>
<GMI_Firm>Z</GMI_Firm>
<GMI_Office>00L</GMI_Office>
<GMI_Account>32638</GMI_Account>
<Pit_Broker>1H7L</Pit_Broker>
<Decimal_Price_Premium>105.7599999</Decimal_Price_Premium>
<Traded_Currency_Code>USD</Traded_Currency_Code>
<Last_Trade_Date>20190426</Last_Trade_Date>
<ISIN></ISIN>
<BBG_Yellow_Key>Comdty</BBG_Yellow_Key>
<BBG_Back_Office_Ticker>GCJ9</BBG_Back_Office_Ticker>
<BBG_Front_Office_Ticker>GCJ9 Comdty</BBG_Front_Office_Ticker>
<BBG_Exchange_Code>CMX</BBG_Exchange_Code>
<RIC>GCJ9</RIC>
<Root_RIC>GC</Root_RIC>
<SEDOL></SEDOL>
<Exchange_Product_Code>GC</Exchange_Product_Code>
<BBG_Unique_ID>IX35663804-0</BBG_Unique_ID>
```

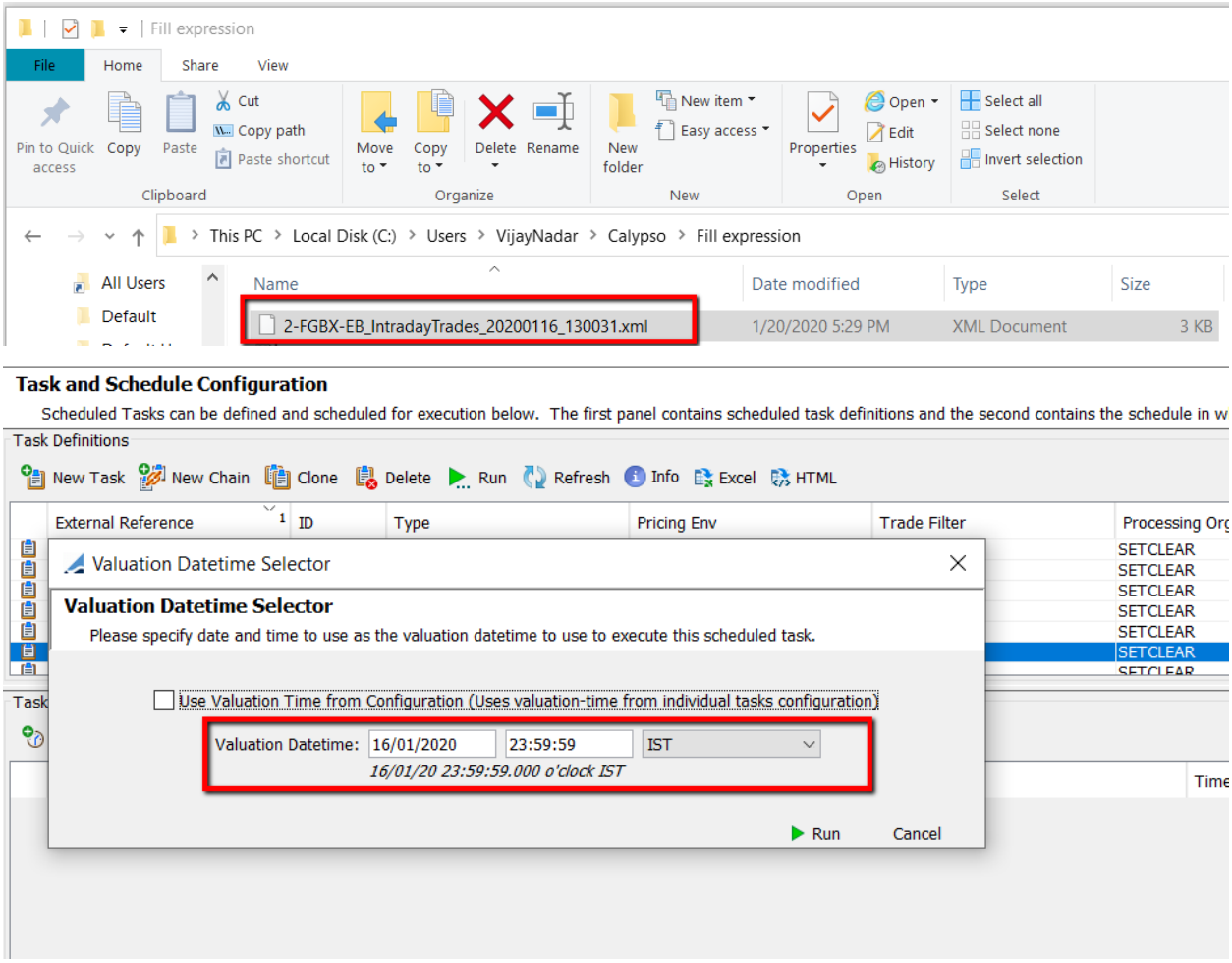
```
<Business_Series_Key></Business_Series_Key>
<GMI_Multiplier>100.000000</GMI_Multiplier>
<APS_Trade>Y</APS_Trade>
<Transaction_Type>NEWO</Transaction_Type>
<Legal_Entity_Id>PQOH26KWDF7CG10L6792</Legal_Entity_Id>
<Notes></Notes>
<Client_Ref>BA</Client_Ref>
<Order_ID>LN71611032</Order_ID>
<Allocation_Time>12:41:55.204000000 GMT</Allocation_Time>
<Trader>Testing 31</Trader>
<UTI>10303389609BESaakLv020190205Z00L32638</UTI>
<MiFID_Time></MiFID_Time>
<Order_Qty>1</Order_Qty>
</row>
</data>
```



Store the file in the folder.

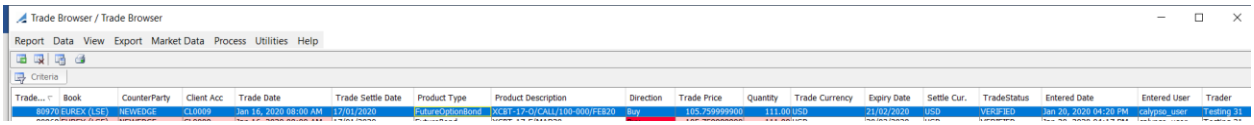


Configure the scheduled task with the following attributes ‘INPUT_FILE_LOCATION’, ‘INPUT_FILE’ and ‘OUTPUT_LOCATION’.

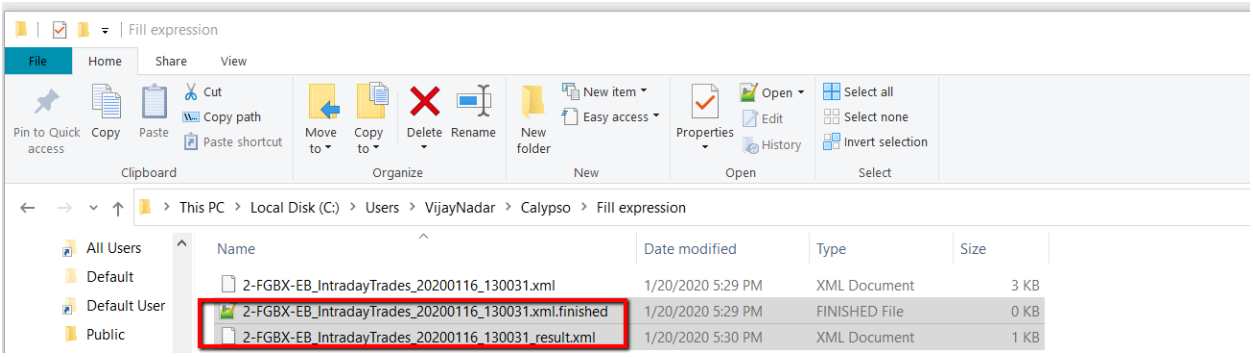


The scheduled task valuation Date should be equal to the file name's date expression.

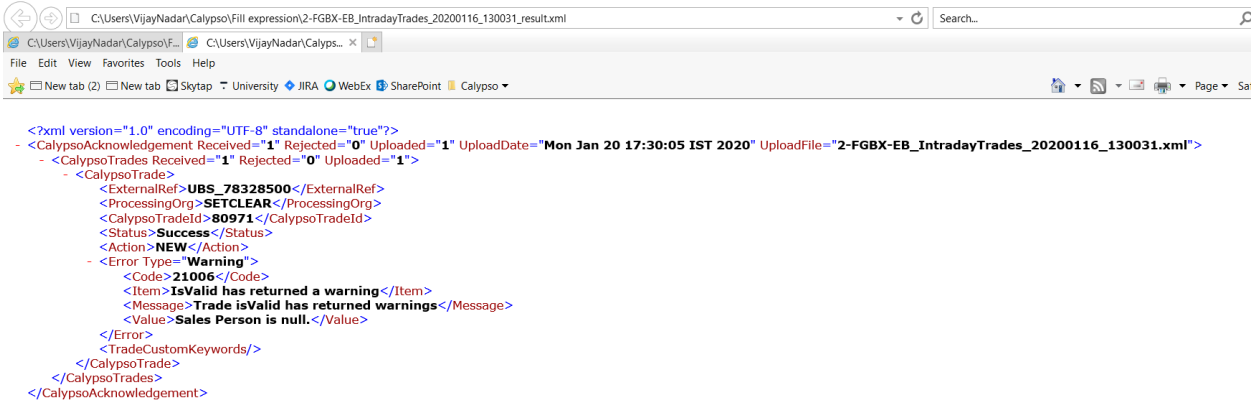
Trade Browser details once trade loaded into the Calypso.



Scheduled task output files:



'_result.xml' output file contents.



This XML file content will give brief details & trade status.