Nasdaq Calypso

CLS Integration Guide

Version 5.1.1

Revision 17.0
February 2024
Approved

Document History

Revision	Published	Summary of Changes
1.0	July 2013	First edition for version 14.0.
2.0	October 2014	Second edition – Updates for version 14.1.
3.0	March 2015	Third edition – Updates for version 14.2 – New CLS Star Program.
4.0	May 2015	Fourth edition – Updates for version 14.3 - FX MX messages.
5.0	August 2015	Fifth edition – Updates for version 2.1.0.
6.0	February 2020	Sixth edition – Updates for version 3.4.0. Note that this version integrates also the Third-Party Service which is described in its own user guide.
7.0	June 2020	Seventh edition – Updates for version 3.4.1. You can now use the JMS gateway instead of the MX Gateway and choose if the message type is binary or text.
8.0	February 2021	Eighth edition – Updates for version 3.4.5.
9.0	December 2021	Ninth edition – Updated for version 3.4.9 – Added CLS Now service.
10.0	February 2022	Tenth edition for version 4.0.0, 4.1.0.
11.0	September 2022	Edition 11 for version 4.3.0 – Added CLS Resilience.
12.0	November 2022	Edition 12 for version 4.5.0 – Improved CLS Resilience (added RESUBMIT action).
13.0	August 2023	Edition 13 for version 4.7.0 – CBPR+ SR 2023 – MX Messages changes.
14.0	October 2023	Edition 14 for version 4.7.1 – Improved CLS Resilience (added "Canceled" status).
15.0	December 2023	Edition 15 for version 4.8.0 - Added Use Cases for CLS Resilience.

Revision	Published	Summary of Changes
16.0	January 2024	Edition 16 for version 5.0.7 (compatibility with version 18) – Improved CLS Resilience RESUBMIT.
17.0	February 2024	Edition 17 for version 5.1.1 – Added CLS Resilience bulk action.

This document describes the setup and use of the CLS (Continuous Linked Settlement) integration for settling foreign exchange transactions finally and irrevocably.

Banks can be either directly members of CLS or being a third party to the CLS member. As such, when they execute a transaction through CLS, they need to send a MT304 to CLS Bank and they do not have to provide any settlement messages (usually) as CLS will take care of the full settlement process.

The Calypso CLS module supports the settlement of members' own trades in CLS.

Calypso does not explicitly support third-party functionality for settlement members but provides a recommended setup for POs that wish to use third-party services offered by CLS members.

The following products are currently settled through CLS:

- FX Spot
- FX Swap
- FX Forward

IMPORTANT NOTE: For Cloud deployments please contact your application management team as the deployment procedure for Cap Cloud is different.

Table of Contents

CLS Activity

In order to understand better at which points the global operational dilemma exists for CLS Members, below is a summary of the normal course of events that occurs for banks that use a CLS GUI to monitor their daily CLS obligations and to verify if their payments have been matched and settled.

Submission

At time of trade execution, members of CLS:

- Submit MT304 to CLS for counterparty using BIC codes
- MT300 to counterparties against which they did the trades
- Operations check CLS GUI for matching of messages

Settlement

00:00 CET

Once CLS updates all trades to Settled Mature and each member has produced its obligations in each Ccy, banks will extracts 'Initial' Pay-In Schedule (acronym IPIS and flagged as type 'official' in Calypso) manually from their CLS GUI and do their own excel or otherwise reconciliation vs. their expectations.

- If a discrepancy exists they investigate
- If no discrepancy then figure is confirmed and they send MT298 to their Nostro Agents
- Each sub-office (if clearing through their main branch) does what they need to do to prepare to exchange Interbank Obligations

00:30 CET

I/O Swaps are generated within CLS and made available to its members via Fax/Website.

Operations, retrieves this information manually and tries to reconcile between the amounts stated by CLS on the printed-out fax/website papers vs. its expectations. Operations, having confirmed with counterparties against whom these I/O Swaps are to be with, inform their relevant desks to book these trades.

- I/O Swaps are auctioned

06:30 CET

CLS produces 'Revised' Pay-In Schedules (acronym RPIS and flagged as type 'official' in Calypso) after collecting all trades in Status Settlement Mature and calculating them against each member's obligation.

Operations of Banks, when seeing their available RPIS on their CLS GUI extract them manually and again begin the process of manual reconciliation comparing figures of CLS vs. their excel or otherwise maintained figures.

- If a discrepancy exists, they investigate
- If no discrepancy or when verified, Operations begin the process of creating MT202 plus ledger entries
- MT298 and MT202 messages are checked and signed off before being sent out

The CLS GUI is monitored until 12:00 CET to check payments have been received and trades are settled, and for Pay-In Calls. Pay-In Calls are payment calls from CLS to its members in the event that another CLS member's account is short. These are not frequent but exist and can be asked on any of the 15 CLS currencies.

Filling the Gap

In view of the operational manual process explained above, Calypso as a result has created a CLS interface which allows clients to:

- Send Confirmation Messages to CLS
- Automate message matching process
- Monitor within Calypso the status of confirmations in CLS
- Extract Payment Schedules (initial and revised) directly from CLS on real-time basis
- Reconcile transfer amounts against the incoming CLS due amounts per Ccy
- Automate the generation of payment statements/messages of MT298 and MT202 at the desired times for the relevant amounts
- If required, sweep account transfers between Main Nostro Accounts and CLS Nostro Accounts
- Manually edit an Initial Pay-In Schedule into a Revised Pay-In Schedule
- Manually generate payment statements and messages
- Extract I/O Swaps and automate the booking of their trades
- If desired, extract and create the relevant transfers in the event of a Pay-In call
- Identify unknown/alleged trades in the event of their existence in CLS

All within the CLS regional cut-off times.

Installation and Setup

The CLS module is installed as part of the Calypso Installer when you select the “CLS Integration” interface.

During the installation, you must add the java libraries from your MQ installation.

▶ Refer to the Calypso Installation Guide for details.

The CLS2MQ gateway client now requires Oracle 11 32-bit client to be installed (previously it was Oracle 10).

2.1 Data Synchronization

Synchronize your database with the CLS data.

Run Execute SQL.

If you are upgrading from CLS version 1.0, you need to run the following upgrade script before synchronizing your data – They are located in `<calypso home>/client/resources/samples/dbscripts/util/downgrade`:

`upgrade_2.00.00_message.sql`

`upgrade_2.00.00_payInSchedule.sql`

`upgrade_2.00.00_tradeInfo.sql`

Then add the following files if not already present:

- `<calypso home>/bin/dbscripts/CLSSchemaBase.xml`
- `<calypso home>/bin/dbscripts/CLSSchemaData.xml`

This will create CLS tables and data.

You can now restart the Auth Server, Event Server, Data Server.

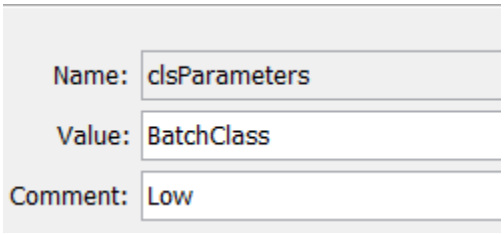
2.2 CLS Parameters

The following parameters need to be defined or reviewed in the domain “clsParameters”.

BatchClass

Specify the class of the MI Channel in the Comment field.

Example:



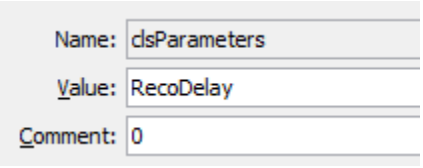
clsLECode

Short name of CLS BANK in case it is not “CLS BANK”.

- Value = clsLECode
- Comment = Short name of CLS member

RecoDelay

The system allows configuring the delay of the official reconciliation using the CLS parameter “RecoDelay”.



By default, RecoDelay=0, the reconciliation will not be delayed and will be run on-the-fly. If there are breaks, tasks will be created in the Task Station.

RecoDelay can be set to a number of minutes. For example, you can set RecoDelay=10, meaning that the official reconciliation will be delayed by 10 minutes.

In this case, at the reception of an IPIS/RPIS, a temporary reconciliation will be run and if there are breaks there will not be any task created in the Task Station as it is not considered as an official reconciliation.

After the specified number of minutes, a new official reconciliation is run and tasks are created as needed (please check AUDIT).

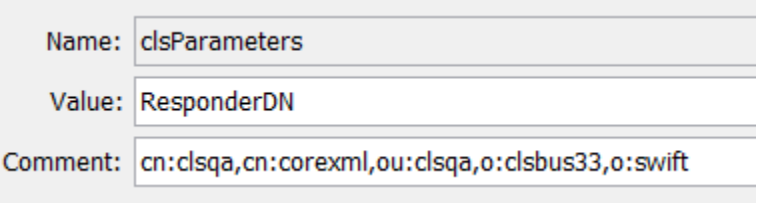
This enhancement will allow waiting for the delays in CLS Trade Notifications due to late process because of the channel (CLS connectivity/processing of the late received notifications).

[Important NOTE: This check is done for reconciliation purposes and does not impact the creation of the Transfer Agent trades at the reception of the RPIS]

ResponderDN

Specify the routing description in the Comment field.

Example:



SenderBIC

Specify the BIC to use to send a message to CLS in the Comment field.

Example:

2.3 CLS Message Engine

The CLS Message engine provides the connection with CLS.

The CLS Message engine is configured in the Engine Manager of Web Admin: event subscription and engine parameters.

You may need to add this engine if it is not available for configuration: Create a new engine called CLSMessageEngine with class name `com.calypso.engine.CLSMessageEngine`.

The "config" engine parameter is config = cls.engine.properties

▶ Please refer to Calypso Web Admin documentation for complete details.

2.3.1 Property File

A sample property file is provided under "`<calypso home>/client/resources/cls.engine.properties`".

```
#Prop for using over FILE SYSTEM Service Provider

#JMS DEF
jms.url=file:///home/bastien/Desktop/
jms.initial.context.factory=com.sun.jndi.fscontext.RefFSContextFactory
jms.queue.connection.factory=CLS_STAR
jms.queue.connection.username=administrator
jms.queue.connection.password=Calypso2012

#Engine def
gateways.names=swift,non_repudiation,admi_camt,dispatcher,fxtr1,fxtr2,fxtr3,fxtr4

#Proc def
swift.input.queue=CLS.OPDATA.SWIFT
swift.batchingSize=10
swift.msgTimeout=1500
swift.class=cls.api.messaging.connectivity.JMSMessageProcessor
swift.delegator=tk.messaging.executioner.CLSProcessor

non_repudiation.input.queue=nonrepudiation
non_repudiation.batchingSize=10
non_repudiation.msgTimeout=1500
non_repudiation.class=cls.api.messaging.connectivity.JMSMessageProcessor
non_repudiation.delegator=tk.messaging.executioner.CLSProcessor

admi_camt.input.queue=admin_camt
admi_camt.output.queue=ack_admin_camt
admi_camt.type=ACK
admi_camt.batchingSize=10
```

```
admi_camt.msgTimeout=1500
admi_camt.class=cls.api.messaging.connectivity.JMSMessageProcessor
admi_camt.delegator=tk.messaging.executioner.CLSProcessor

dispatcher.input.queue=bulk_in
dispatcher.output.queue=preprocessed
dispatcher.batchingSize=10
dispatcher.msgTimeout=1000
dispatcher.class=cls.api.messaging.connectivity.JMSMessagePreprocessor
dispatcher.delegator=preprocessor.messaging.executioner.CLSDispatcher

fxtr1.input.queue=preprocessed
fxtr1.msgSelector=HASH_KEY >= 0 AND HASH_KEY <= 64
fxtr1.batchingSize=10
fxtr1.msgTimeout=1500
fxtr1.class=cls.api.messaging.connectivity.JMSMessageProcessor
fxtr1.delegator=tk.messaging.executioner.CLSProcessor

fxtr2.input.queue=preprocessed
fxtr2.msgSelector=HASH_KEY > 64 AND HASH_KEY <= 128
fxtr2.batchingSize=10
fxtr2.msgTimeout=1500
fxtr2.class=cls.api.messaging.connectivity.JMSMessageProcessor
fxtr2.delegator=tk.messaging.executioner.CLSProcessor

fxtr3.input.queue=preprocessed
fxtr3.msgSelector=HASH_KEY > 128 AND HASH_KEY <= 192
fxtr3.batchingSize=10
fxtr3.msgTimeout=1500
fxtr3.class=cls.api.messaging.connectivity.JMSMessageProcessor
fxtr3.delegator=tk.messaging.executioner.CLSProcessor

fxtr4.input.queue=preprocessed
fxtr4.msgSelector=HASH_KEY > 192 AND HASH_KEY <= 256
fxtr4.batchingSize=10
fxtr4.msgTimeout=1500
fxtr4.class=cls.api.messaging.connectivity.JMSMessageProcessor
fxtr4.delegator=tk.messaging.executioner.CLSProcessor
```

You need to set the following properties in this file:

jms.url: location of your .binding file generated from MQ or jms jndi access

jms.initial.context.factory: your JMS initial context factory

jms.queue.conction.factory: name of your connection factory created into MQ

(optional) **jms.queue.connection.username**: username for the connection to your MQServer

(optional) **jms.queue.connection.password**: password for the connection to your MQServer

gateway.names: names of all gateways that you want to start. A gateway is identified by its name, so this is mandatory. A gateway can receive and send messages. Basically, in CLS case, a gateway is related to an input queue.

Then each gateway has different properties:

input.queue: name of your input queue, in order to receive messages from CLS (MQ related)

(optional) **output.queue**: name of your output queue, in order to send message to CLS (MQ related)

batchingSize: number of messages that you want to process in one JMS session (be careful, in case of error you will rollback this number of message)

msgTimeout: in milliseconds, if you don't receive a message during this time, your session will be committed. Useful when there is a number of messages which is not divisible by your batching size.

class: JMS implementation, here you can choose if your gateway is a 'Processor' or a 'Preprocessor' or something custom. Default implementation JMSMessageProcessor and JMSMessagePreprocessor (see example for complete package)

delegator: class that will process/preprocess your message. Default implementation CLSProcessor or CLSDispatcher

msgSelector: if you want to run several gateways for one input queue you need a message selector in order to forward your message. This will ensure that the queue order is not broken.

The property file must 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.

2.3.2 Communication with MQ Series through JMS

The CLS Message engine can listen to different queues. The pre-processor is not mandatory but significantly speeds-up your processing time. The pre-processor will add a JMSProperty in your JMSMessage called **HASH_KEY**, it is a load balancing key based on the CLS reference.

HASH_KEY = cls reference's hashcode % 256. If this value is negative we add 256, in order to have something between 0 and 256.

Then, you have to setup the gateways with a message selector in order to cover this range.

If your business does not need a pre-processor, just configure only one gateway without message selector.

The pre-processor will put back the message into MQ, so you have to configure a 'pre-processed' queue.

In your engine configuration, your pre-processor needs an output queue when an acknowledgment is needed.

The processor saves the CLS object into your database, generates transfers, and does the reconciliation.

2.4 Supported Incoming Messages

Please find below the list of supported incoming message sent by CLS:

Message Description	ISO/OUT of CLS	Calypso CLS Report	Task	Ack	ISO/IN CLS
Bulk status notification For ex: Settle	fxtr.030.001.02	CLSTradeInfo Report	No	No	N/A
Trade short notification For ex: Rescind	fxtr.008.001.04	CLSTradeInfo Report	No	No	N/A
Trade Long notification (Trade Status) For Ex: Split/ Matched/Settlement Mature	fxtr.017.001.02	CLSTradeInfo Report	No	No	N/A
Withdrawal notification	fxtr.013.001.03	Update on the CLSTradeInfo Report	No	No	N/A
Statements	camt.053.001.04	Not Supported	No	No	N/A
Pay out notification	camt.054.001.04	CLSAccountNotification	Yes if Exception	No	N/A
Pay in notification	camt.054.001.04	CLSAccountNotification Xfers Agent Xfers = SETTLED	Yes if Exception	No	N/A
Settlement notification	camt.054.001.04	Not Supported		No	N/A
Pay-in call	camt.061.001.02	CLSPayInSchedule Report	Yes	Yes	camt.063.001.02

Message Description	ISO/OUT of CLS	Calypso CLS Report	Task	Ack	ISO/IN CLS
Pay-in schedule	camt.062.001.03	CLSPayInSchedule Report	No	Yes	camt.063.001.02
Message Failure	admi.002.001.01	CLSMessage Report	Yes	No	N/A
System Event Other	admi.004.001.01	CLSMessage Report	Yes	No	N/A
System Event Operational Msg	admi.004.001.01	CLSMessage Report	Yes	Yes	admi.011.001.01
Member Suspension	admi.004.001.01	CLSMessage Report/ CLS Trade Info Update	Yes	No	N/A

2.5 Supported Status Codes and Sub Status Codes

Name	Status	Sub Status	Description
ALLEGED	UMTC	IURT	Unmatched alleged
ALLEGED_RESCIND	UMTC	SRST	Unmatched alleged due to rescind
ALLEGED_AMEND	UMTC	AMUI	Unmatched alleged due to amendment of matched trade
ALLEGED_REINSTATED	UMTC	NIRA	Unmatched alleged - Reinstated
ALLEGED_NEW_DATE	UMTC	CCAA	New Value Date and Unmatched Alleged
MATCHED	FMTC	IMAT	Matched status
MATCHED_REINSTATED	FMTC	NIRI	Matched reinstated. Previously suspended
MATCHED_NEW_DATE	FMTC	CCAM	New Value Date and Matched
INVALID	INVA	no code	Invalid status
INVALID_001	INVA	0001	Instruction contains a negative amount to be bought.
INVALID_002	INVA	0002	Instruction contains a negative amount to be sold
INVALID_003	INVA	0003	Instruction contains an unknown currency to be bought
INVALID_004	INVA	0004	Instruction contains an unknown currency to be sold

Name	Status	Sub Status	Description
INVALID_005	INVA	0005	Instruction contains a currency to be bought which is not eligible
INVALID_006	INVA	0006	Instruction contains a currency to be sold which is not eligible.
INVALID_007	INVA	0007	Instruction contains a currency to be bought that is deactivated.
INVALID_008	INVA	0008	Instruction contains a currency to be sold that is deactivated.
INVALID_009	INVA	0009	Instruction received with a value date beyond the calendar currently defined for one or both Designated currencies.
INVALID_010	INVA	0010	RTGS is not open on the given value date for the currency to be bought
INVALID_011	INVA	0011	RTGS is not open on the given value date for the currency to be sold
INVALID_012	INVA	0012	Originator BIC relates to a settlement member who is not eligible in the currency to be bought.
INVALID_013	INVA	0013	Originator BIC relates to a settlement member who is not eligible in the currency to be sold.
INVALID_021	INVA	0021	Originating member does not relate to a settlement member
INVALID_022	INVA	0022	Counterparty member does not relate to a settlement member
INVALID_026	INVA	0026	The Originating BIC is not defined in the CLS System as a BIC that has been included in the static data of a member
INVALID_027	INVA	002	The Counterparty BIC is not defined in the CLS System as a BIC that has been included in the static data of a member.
INVALID_030	INVA	003	Originator BIC relates to a member who is not eligible.
REJECTED	RJCT	no code	Rejected Status

Name	Status	Sub Status	Description
REJECTED_001	RJCT	0001	Generated when a Short Notice Bank Holiday is declared between the Initial Pay-In Schedule Deadline and the Start of Settlement Business Day for all unsettled split component Instructions which have a buy or sell currency equal to that of the currency declared in the Short Notice Bank Holiday.
REJECTED_002	RJCT	0002	Instruction is received which is a duplicate of another Instruction
REJECTED_003	RJCT	0003	Instruction received with value date in the past or Instruction received with value date today and Currency Close Deadline reached for one of the currencies of the Instruction.
REJECTED_004	RJCT	0004	Generated at Currency Close Deadline for all unsettled Instructions in that currency.
REJECTED_005	RJCT	0005	Generated at Currency Close Deadline for each Instruction that has been suspended by the Regulatory Filter and has not been re-instated by the Currency Close Deadline.
REJECTED_006	RJCT	0006	Same currency appears more than once in an Instruction.
REJECTED_007	RJCT	0007	Input contains a buy currency which is suspended for input.
REJECTED_008	RJCT	0008	Input contains a sold currency which is suspended for input.
REJECTED_009	RJCT	0009	Originator BIC relates to a member who is suspended for Input in the currency to be bought.
REJECTED_010	RJCT	0010	Originator BIC relates to a member who is suspended for Input in the currency to be sold.
REJECTED_011	RJCT	0011	Counterparty BIC relates to a member who is suspended for Input in the currency to be bought.
REJECTED_012	RJCT	0012	Counterparty BIC relates to a member who is suspended for Input in the currency to be sold.
REJECTED_013	RJCT	0013	The Originator BIC of the Instruction has been deleted from the CLS System.

Name	Status	Sub Status	Description
REJECTED_014	RJCT	0014	The Counterparty BIC of the Instruction has been deleted from the CLS System
REJECTED_016	RJCT	0016	Input Rejected as a Fund or an original party not pre-advised
REJECTED_017	RJCT	0017	The currency to be bought has been deactivated
REJECTED_018	RJCT	0018	The currency to be sold has been deactivated
RESCINDED	RSCD	no code	Rescinded
RESCINDED_ALLEGED	RSCD	SRST	Alleged rescinded due to counterparty rescind
RESCINDED_001	RSCD	0001	Unilateral rescind successful
RESCINDED_002	RSCD	0002	Bilateral rescind successful
SETTLED	STLD	no code	Settled
SPLIT	SPLI	no code	Split
STTL_MATURE	SMAT	no code	Settlement Mature
STTL_MATURE_PENDING	SMAP	no code	Settlement Mature Pending
STTL_MATURE_NO_PENDING	SMAT	NLPE	Settlement Mature – No Longer Pending
STTL_MATURE_ALL	SMAT	ASMA	All Settlement Mature
SUSPENDED_DEFAULT	SUSP	no code	Suspended
SUSPENDED	SUSP	NISP	Suspended
SUSPENDED_ALLEGED	SUSP	NISA	Alleged trade is now Suspended
SUSPENDED_NEW_DATE	SUSP	CCAM	New Value Date and Suspended
UNMATCHED_DEFAULT	UMTC	no code	Unmatched
UNMATCHED	UMTC	IURT	Unmatched
UNMATCHED_RESCIND	UMTC	SRST	Unmatched due to counterparty rescind
UNMATCHED_AMEND	UMTC	AMUI	Unmatched due to counterparty amendment of matched trade

Name	Status	Sub Status	Description
UNMATCHED_REINSTATED	UMTC	NIRI	Unmatched - Reinstated
UNMATCHED_NEW_DATE	UMTC	CCAM	New Value Date and Unmatched
WITHDRAW	WTDN	AMUI	Withdrawn due to counterparty amendment

Reference Data Setup

3.1 Legal Entities

Members of CLS can either be “Control Branches” or “Submission POs”. The Control Branch is the PO that makes all payments directly to CLS. They are also the ones who make payments to CLS on behalf of the Submission POs.

Thus, as a result, for certain Processing Organizations where Submission POs exist as well as Control Branches, they both need to be set up in Calypso with the Submission PO linked (via an attribute) to its relevant Control Branch. Should only a Control Branch exist then you only set up the Control Branch.

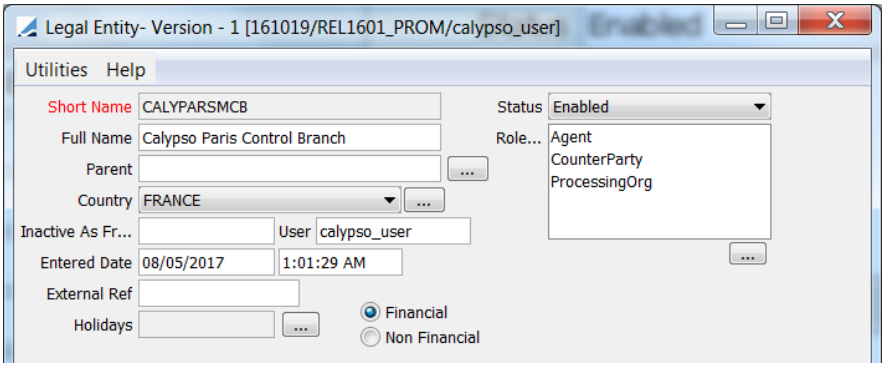
To insure that the CLS Interface functionality within Calypso works as it should for each Legal Entity desired to clear through CLS (whether be it a Control Branch, Submission PO or a Counterparty), you need to set up the correct Legal Entity details: Attributes, Legal Agreements, Contacts and Settlement Instructions.

3.1.1 Control Branch / Settlement Member

These are recipients of Pay-In Schedules and Pay-In Calls. These are the Control Branches for all Submission POs. You need to set up the Control Branch as follows:

- Role = PO

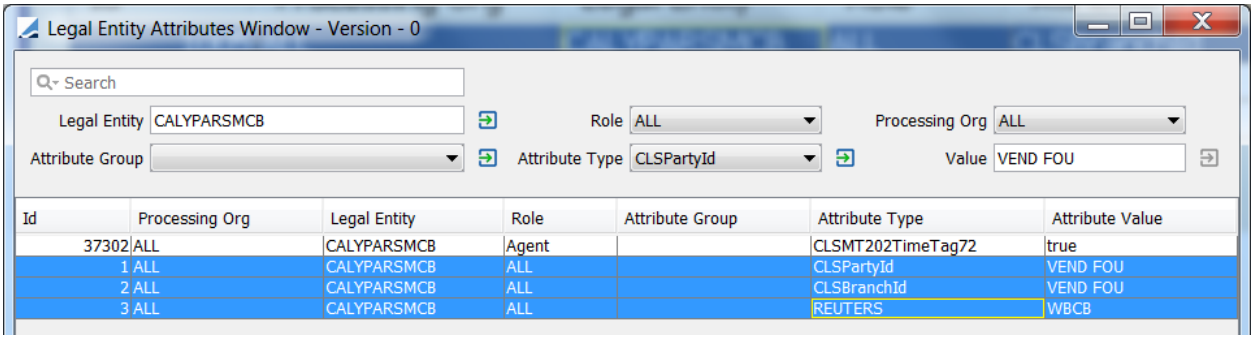
From the Calypso Navigator, navigate to Configuration > Legal Data > Entities.



Attributes

For the Control Branch the following Legal Entity Attributes need to be defined.

- CLSPartyId (This serves as Party key and an identifier i.e., nick name for associating Pay-In Schedules and Pay-In Calls with this legal Entity). This PartyId will be dependent on what CLS defines your particular PartyId to be.
- CLSBranchId (This serves to associate Messages for the branch with the Legal Entity). This BranchId will be dependent on what CLS defines your particular BranchId to be.
- REUTERS (This serves to identify the Legal Entity when I/O Swaps are fed in from CLS)

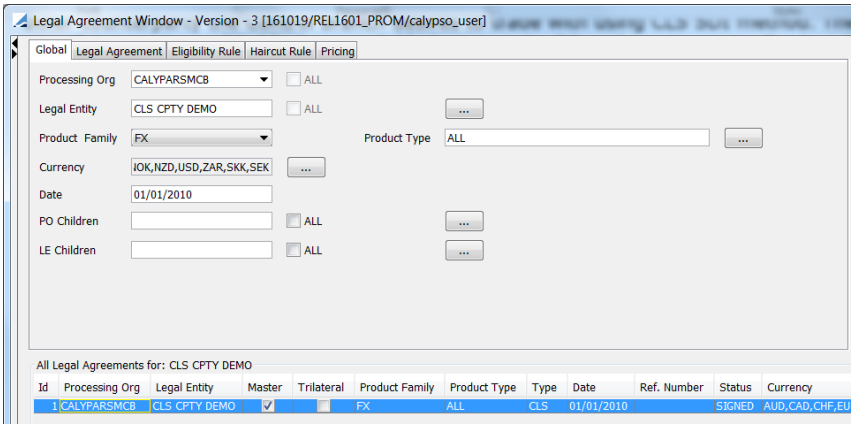


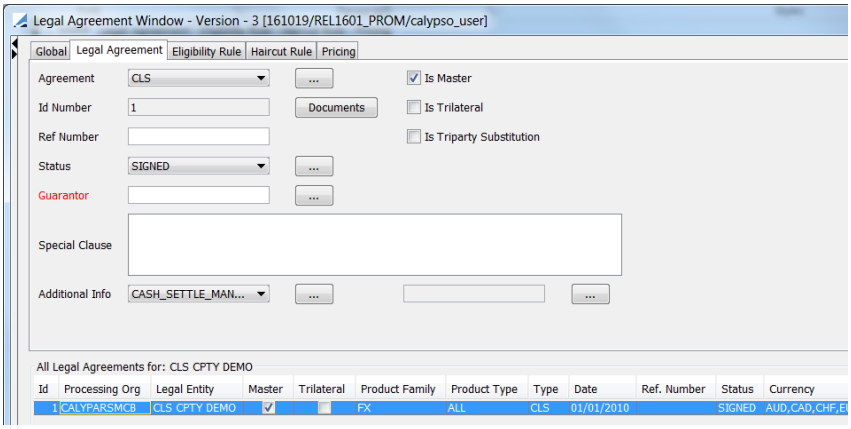
Legal Agreement

A legal agreement with the CLS contract parameters needs to be defined between the Control Branch PO and every other counterparty the Control branch desires to trade via CLS.

In the FX Trade window, the visibility of the CLS checkbox is dependent on the existence of a Legal Agreement between PO and Counterparty for a specific product type and specified Currencies. In this case the checkbox being checked or not is dependent on correct SDIs.

A legal agreement with the CLS contract parameters needs to be defined between the Control Branch/PO and every other counterparty the Control branch desires to trade with using CLS SDI method. The legal agreement allows identifying trades as “CLS” when available between the PO and counterparty. Product Family has to be defined and Currency has to be set to either exactly as in the Ccy of the SDIs for PO and Cpty or left to ‘Any.’





Contact

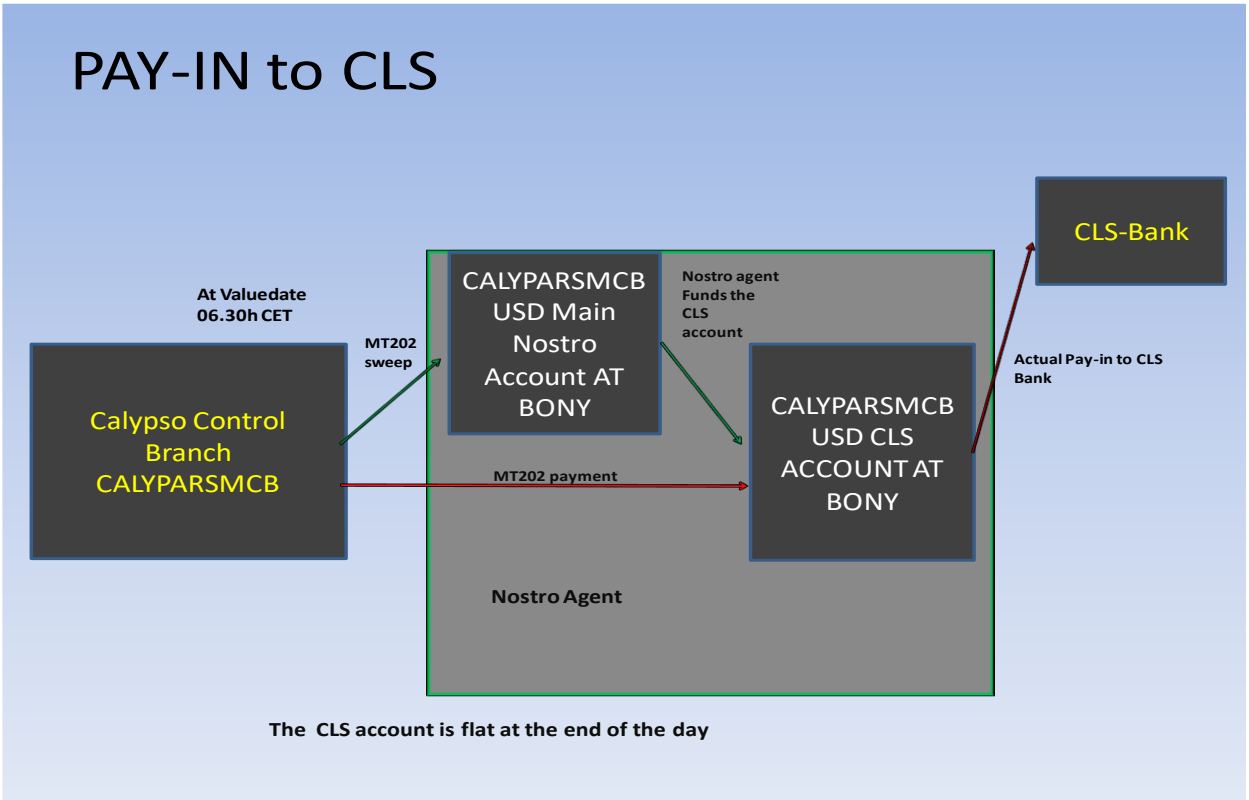
The contact for the Control branch should have a Swift BIC Address to facilitate the correct MT298 and MT202 messages generations and for MT300 and MT304 messages for FX trades done with Counterparties. Message setup is described later in the document.

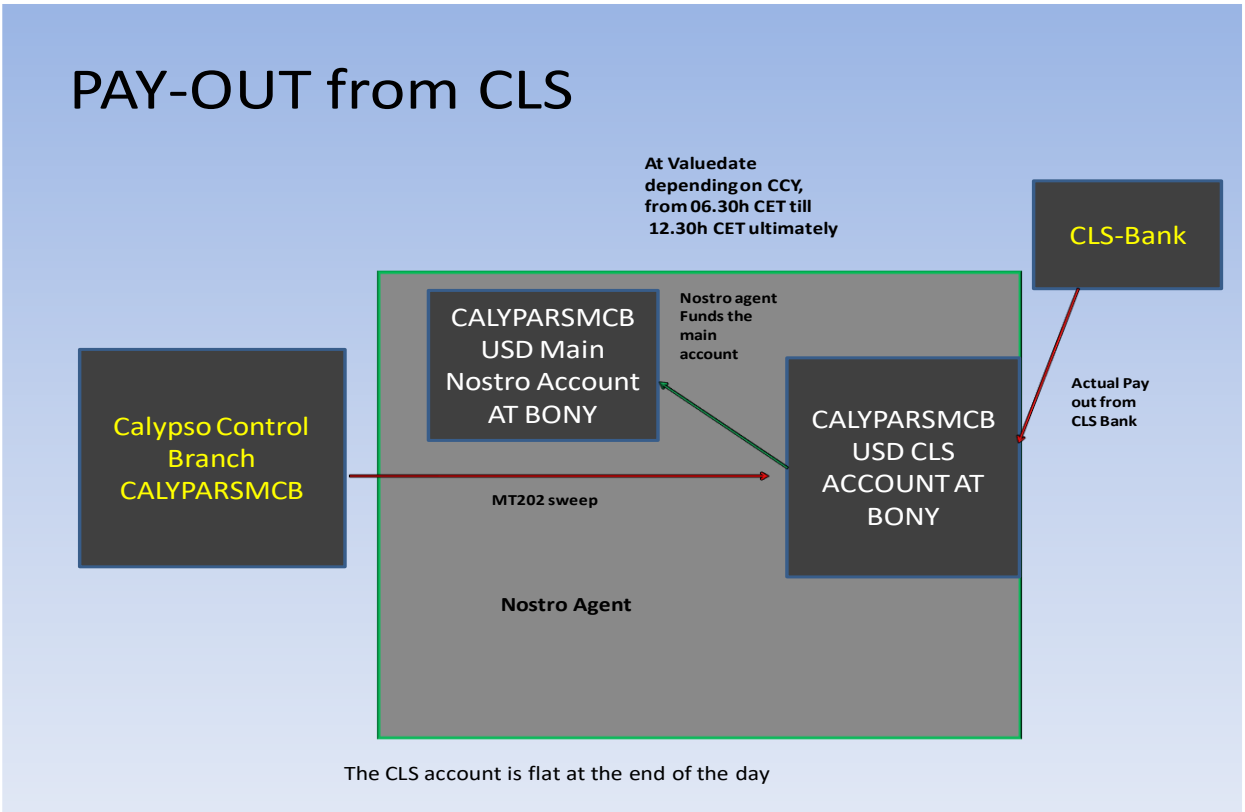
Settlement Instructions

For the Control Branch, as mentioned above part of your SDI setup, will be dependent on whether you have sweep accounts from your Main Nostro Account to your CLS Accounts at your Nostro before transferring from your CLS Accounts at your Nostro to your Accounts at CLS or you just have accounts that make payments from your CLS Accounts at your Nostro to your accounts at CLS.

Calypso handles these settlement instructions via Transfer Agent trades which generate the necessary payment messages, but we'll cover this at a later stage. The rest of the SDI set up will be to support any FX trades/transactions which are in or out of CLS.

Should sweep accounts be involved, the diagram below explains the flow of payment messages for Pay-In and Pay-Out calls.


PAY-IN to CLS
At Valuedate
06.30h CET
MT202
sweep
Calypso Control
Branch
CALYPARSMCB
CALYPARSMCB
USD Main
Nostro
Account AT
BONY
Nostro agent
Funds the
CLS
account
MT202 payment
CALYPARSMCB
USD CLS
ACCOUNT AT
BONY
Nostro Agent
CLS-Bank
Actual Pay-in to CLS
Bank
The CLS account is flat at the end of the day


PAY-OUT from CLS
At Valuedate
depending on CCY,
from 06.30h CET till
12.30h CET ultimately
Calypso Control
Branch
CALYPARSMCB
CALYPARSMCB
USD Main
Nostro Account
AT BONY
Nostro agent
Funds the
main
account
MT202 sweep
CALYPARSMCB
USD CLS
ACCOUNT AT
BONY
Nostro Agent
CLS-Bank
Actual Pay
out from
CLS Bank
The CLS account is flat at the end of the day

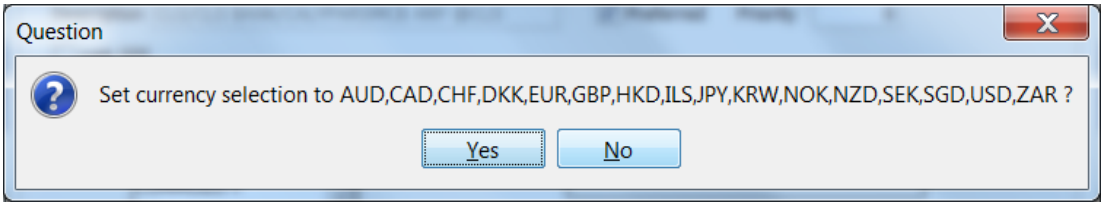
The following types of SDIs can be defined:

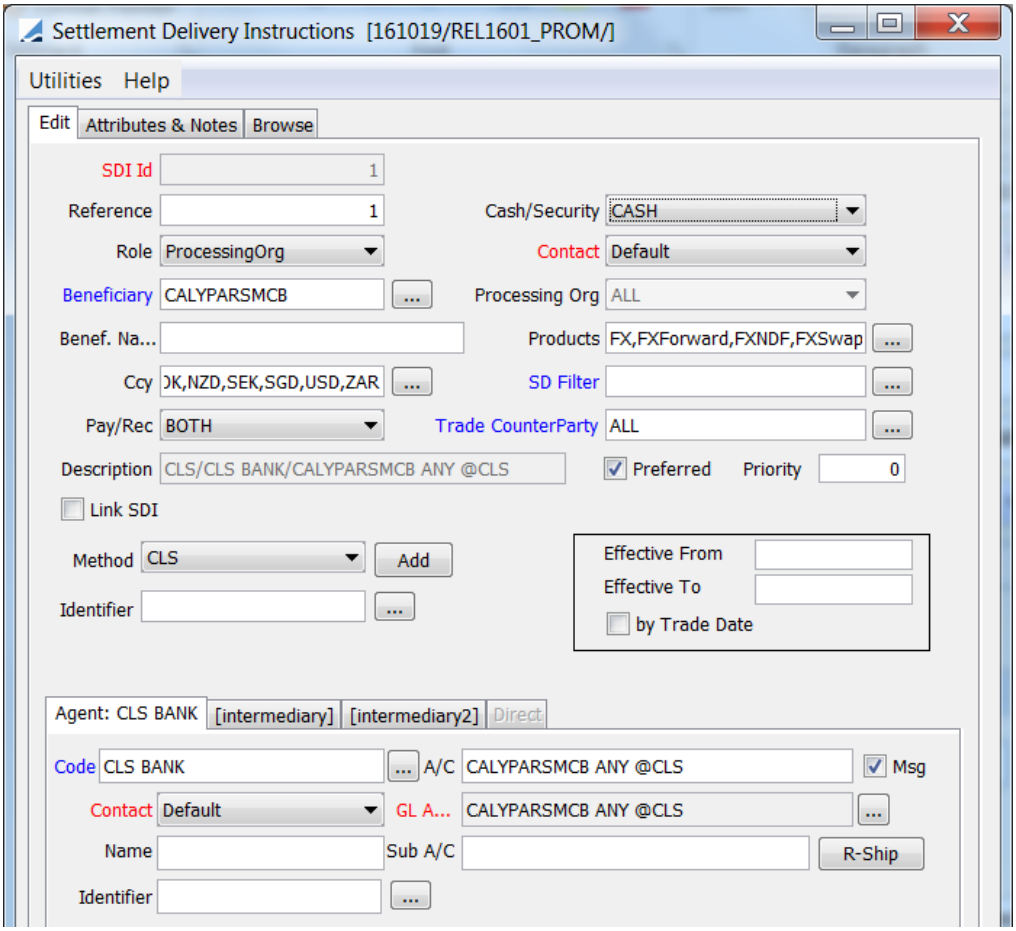
- 1st type: 'Method' = CLS - 'Agent' = CLS Bank
- 2nd type: 'Method' =SWIFT - 'Agent' = Your Nostro2
- 3rd type: 'Method' = SWIFT - 'Agent' = CLS Bank
- 4th type: 'Method' = SWIFT – 'Agent' = Your Nostro (SDIAttribute:CLSUsage=sweep)

1st type: 'Method' CLS - 'Agent' CLS Bank

Create a Pay and a Receive (or both) for Control Branch for Security 'Cash.' This SDI will be used for FX trades done with Counterparties using CLS settlement Method. The method to use here is CLS and the Agent is CLS Bank. Again, currencies in which payments would be made and the product for which it will be used HAVE to be defined here.

[NOTE: As soon as you define CLS as Settlement Method you will get a pop up which will ask you if you want to set your currency selection to the 17 CLS currencies in your default. If you say 'Yes' then it will automatically populate the 17 currencies into your Ccy section]





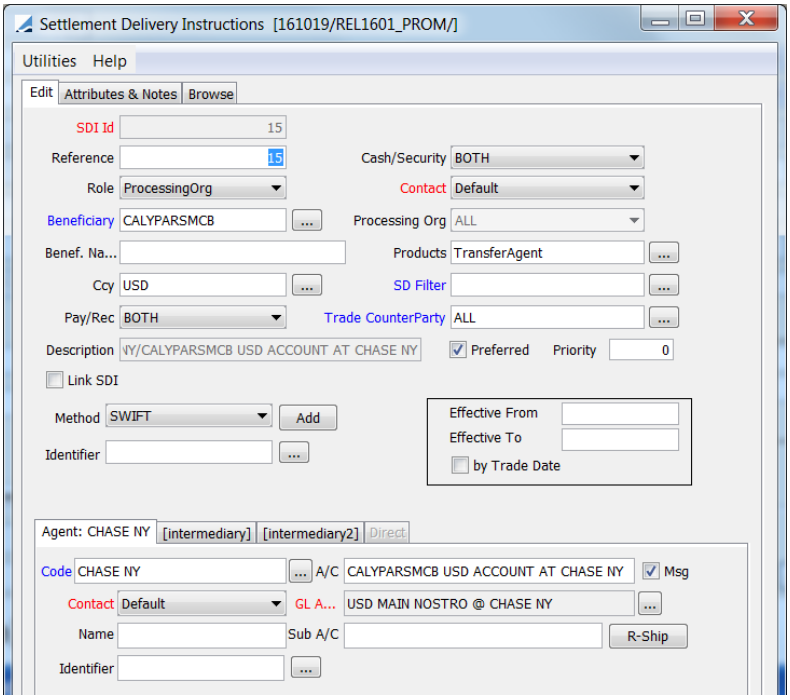
2nd type: 'Method' = SWIFT - 'Agent' = Your Nostro

The use of this SDI set up could be twofold: (a) you can use this SDI in the event that you do not want to set up a separate SDI for settling all other FX transactions that are not in CLS, such as the far leg of an I/O Swap or other FX transactions. Or (b) this SDI can be used by the TransferAgent trades (generated after the Pay-In Schedules are reconciled and in the event you activate the Pay-In call trade generation) to send payment instructions from PO to Nostro Agent in order to eventually instruct payments to be made from your CLS accounts at your Nostro to your accounts at CLS.

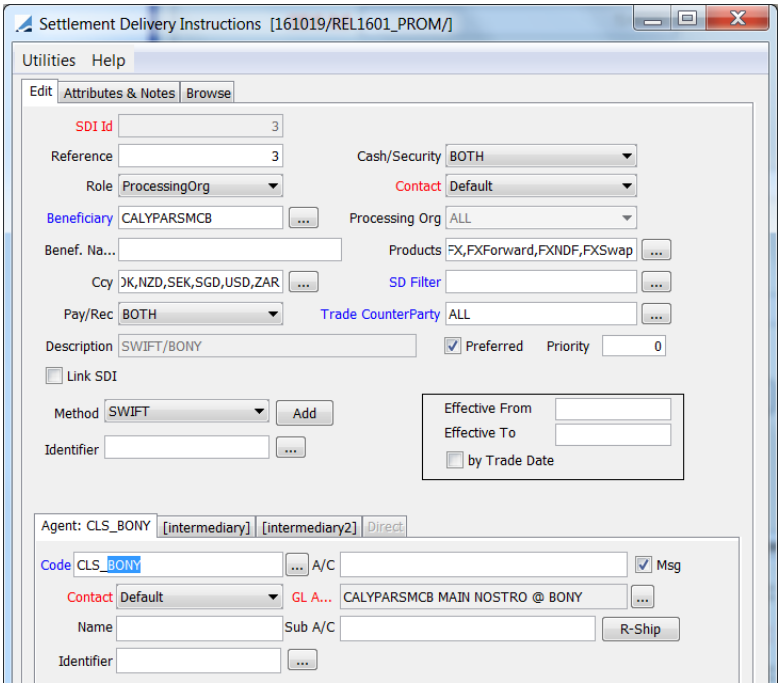
Create a Pay and a Receive (or both) for Control Branch for Security 'Cash.' The 'Method' here does not have to be SWIFT but is recommended unless of course your Nostro Agent has ability to transfer method to Swift to deliver to CLS. 'Agent' used should be your Nostro Agent. NOT CLS.

[NOTE: In my example, I have configured an SDI to be used for scenario (b), but for scenario (a) I have configured a separate SDI. If you intend to use the same SDI for scenario (a) as well, please make sure the products in your SDI reflect the FX products]

SDI's per currency to my Nostro Agent.



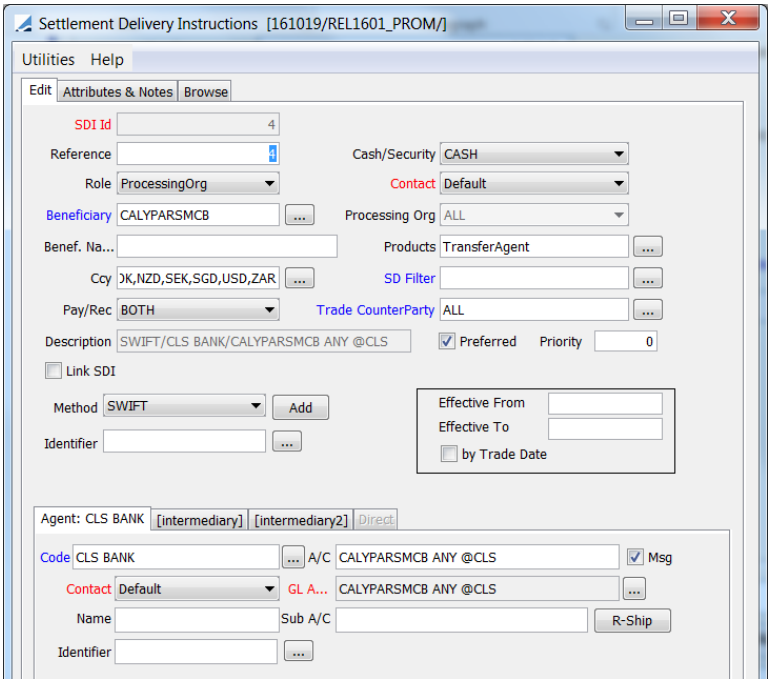
My separate SDI for scenario (a)



3nd type: 'Method' = SWIFT / 'Agent' = CLS Bank

Create a Pay and a Receive (or both) for Control Branch for Security 'Cash.' This SDI will be used only by the TransferAgent trades generated after the Pay-In Schedules are reconciled and in the event you activate the generation of Pay-In call trades.

Control Branch will use this SDI in order to pay its own account at CLS. The method to use here is Swift and Agent IS CLS. Currencies in which payments to CLS account would be made should be defined as part of the SDI as well as the product it will be used for.



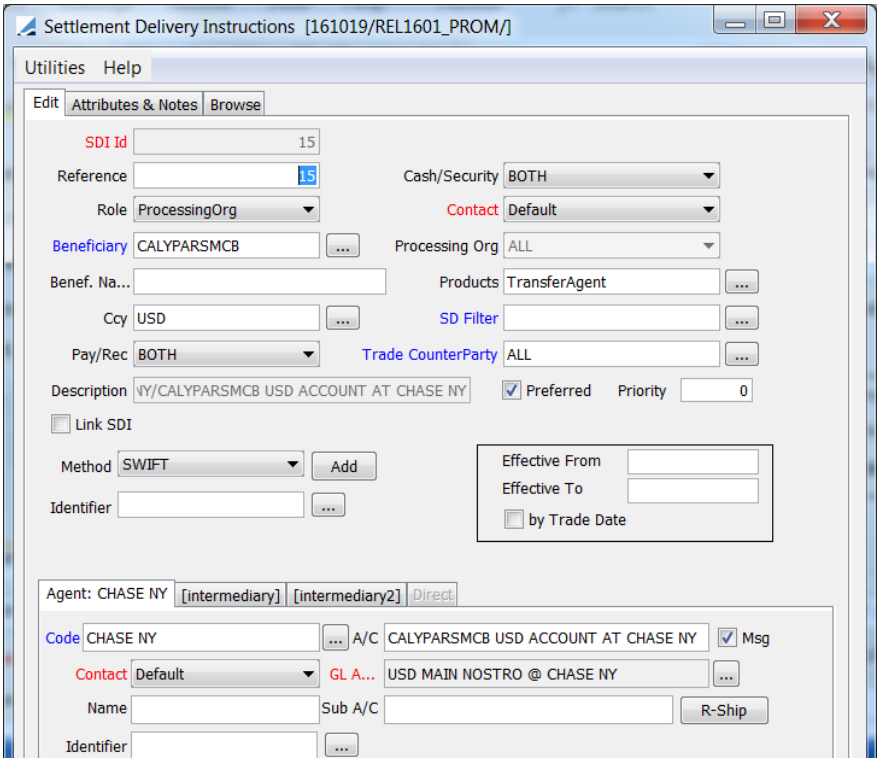
4th type: 'Method = 'SWIFT/ 'Agent' = Your Nostro (SDIAttribute:CLSUsage=sweep)

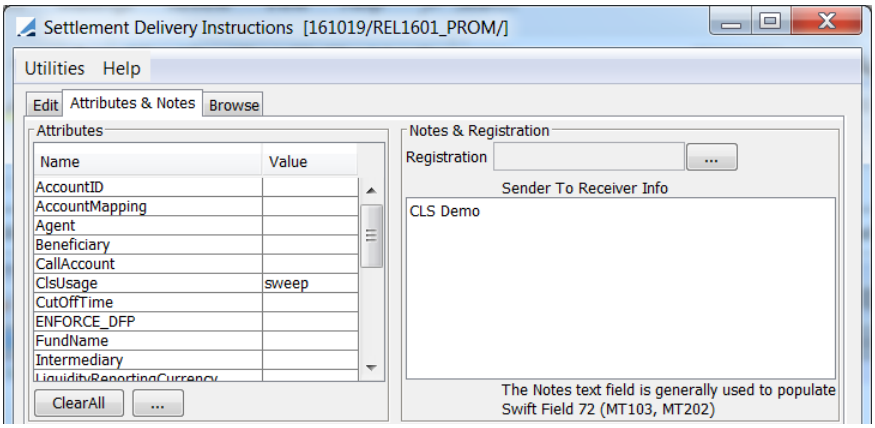
In the case where you may have sweep accounts from your Main Nostro to your CLS accounts at your Nostro you first need to set up an SDI as the following:

Create a Pay and a Receive (or both) for Control Branch for Security 'Cash.' After the Pay-In Schedules are reconciled and in the event, you activate the Pay-In call trade generation, this SDI will be used by the TransferAgent trades generated to sweep payment instructions between your Main Nostro and your CLS accounts at your Nostro.

The 'Method' here does not have to be SWIFT but is recommended unless of course your Nostro Agent has ability to transfer method to Swift to deliver to your CLS accounts at your Nostro.

'Agent' used should be your Nostro Agent. NOT CLS. In order for the Sweep to work you need to set up an SDI attribute CLSUsage, with the value Sweep.

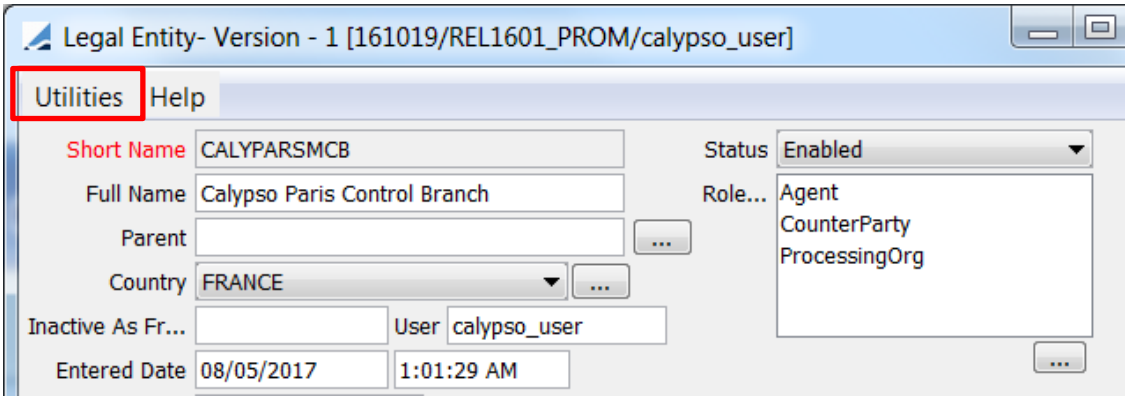




Once you have finished all the set-up for your Control Branch Legal Entity, the 'CLS Setup Check' in 'Utilities' menu of your Legal Entity window allows you to double check if your configuration is complete for that Legal Entity.

[NOTE: This check also double checks on the related books defined for the PO for the CLS use against what has been configured in your Domain Values. Therefore, once you have configured your books correctly, you may want to come back and double check your CLS Setup again]

From the Calypso Navigator, navigate to Configuration > Legal Data > Entities > Utilities.



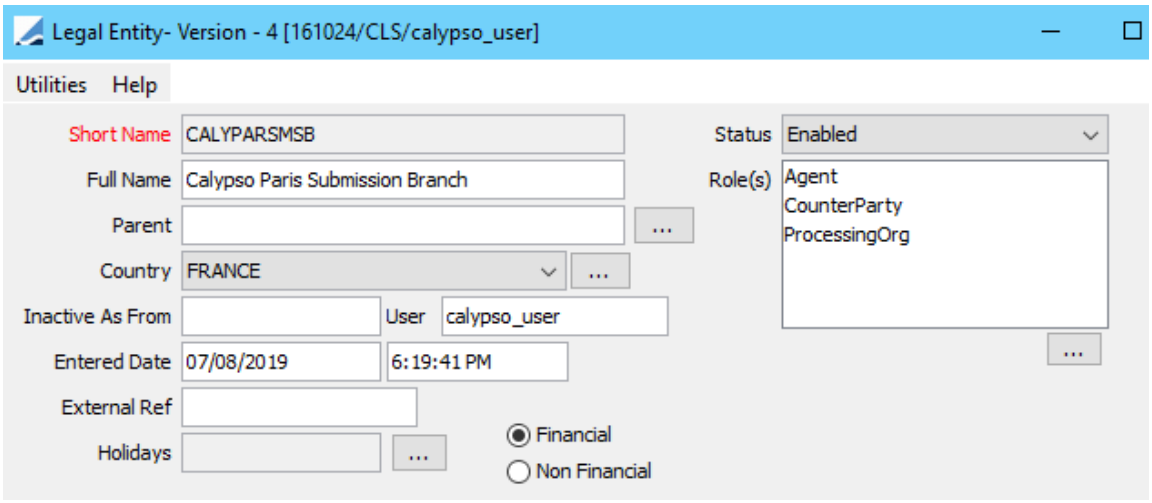
3.1.2 Settlement Members / Settlement POs

These settings are recommendations as Calypso does not explicitly support third-party functionality for settlement members.

Settlement POs (or Submissions Branches) are Processing Organizations that although trade with counterparties using CLS settlement methods do not make any payments directly to CLS but go through their Control Branches. The Control Branch will clear their payments. You need to set up the Settlement POs as follows:

- Role = PO

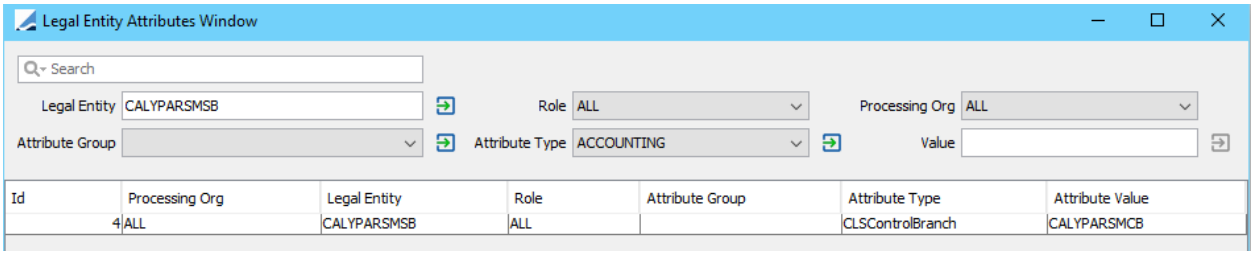
From the Calypso Navigator, navigate to Configuration > Legal Data > Entities.



Attributes

For the Submission PO the following attribute needs to be defined.

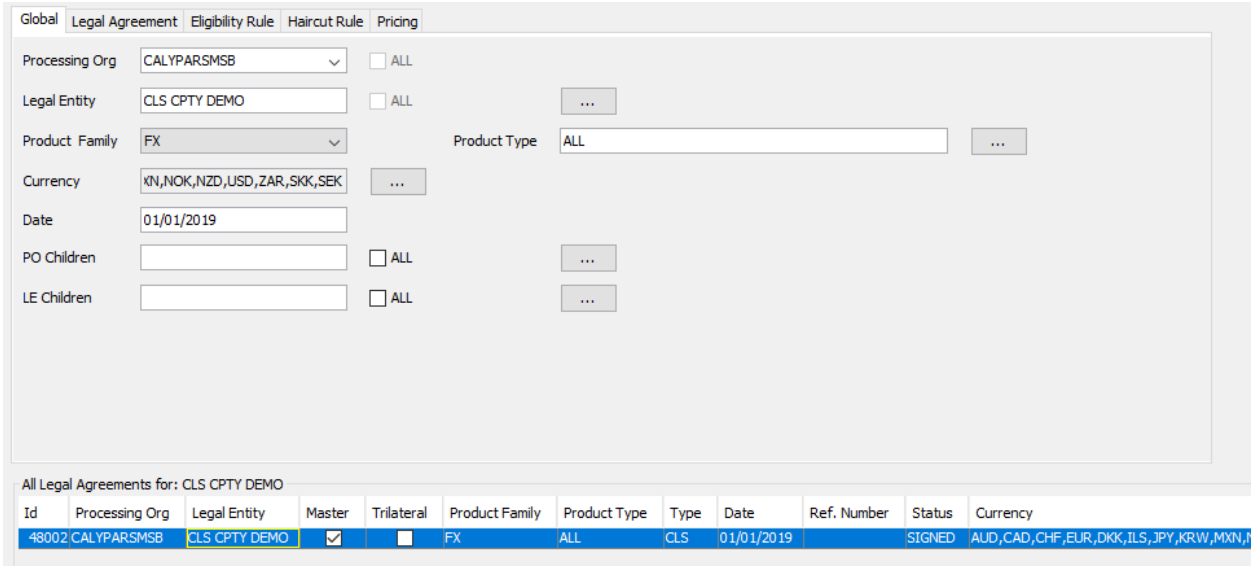
- CLSControlBranch (This is the short name of your Control Branch. This will identify the Control Branch that will be submitting for this Submission PO and thus will include the Submission POs trades in the reconciliation process).

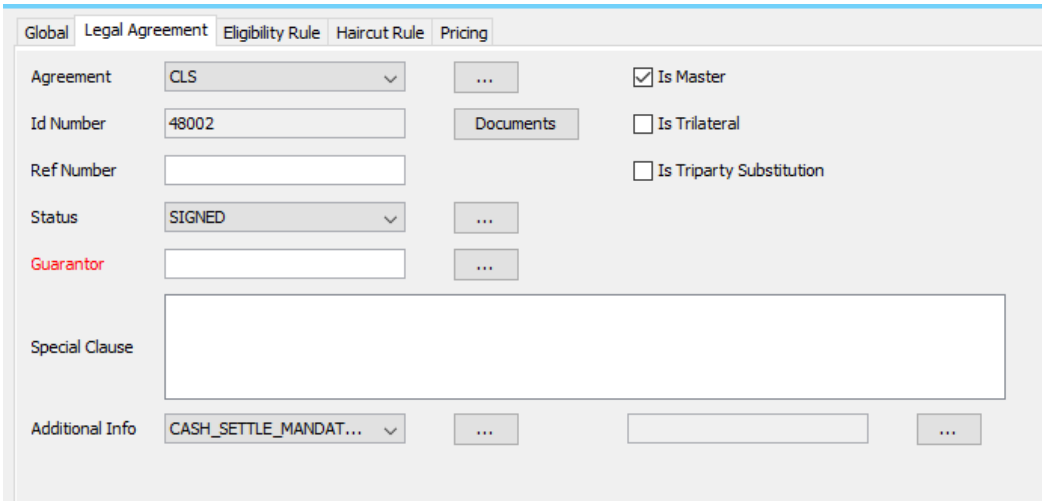


Legal Agreement

A legal agreement with the CLS contract parameters needs to be defined between the Submission PO and every other counterparty the Submission PO desires to trade via CLS.

In the FX Trade window, the visibility of the CLS checkbox is dependent on the existence of a Legal Agreement between Submission PO and Counterparty for a specific product type and specified Currencies. In this case the checkbox being checked or not is dependent on correct SDIs.





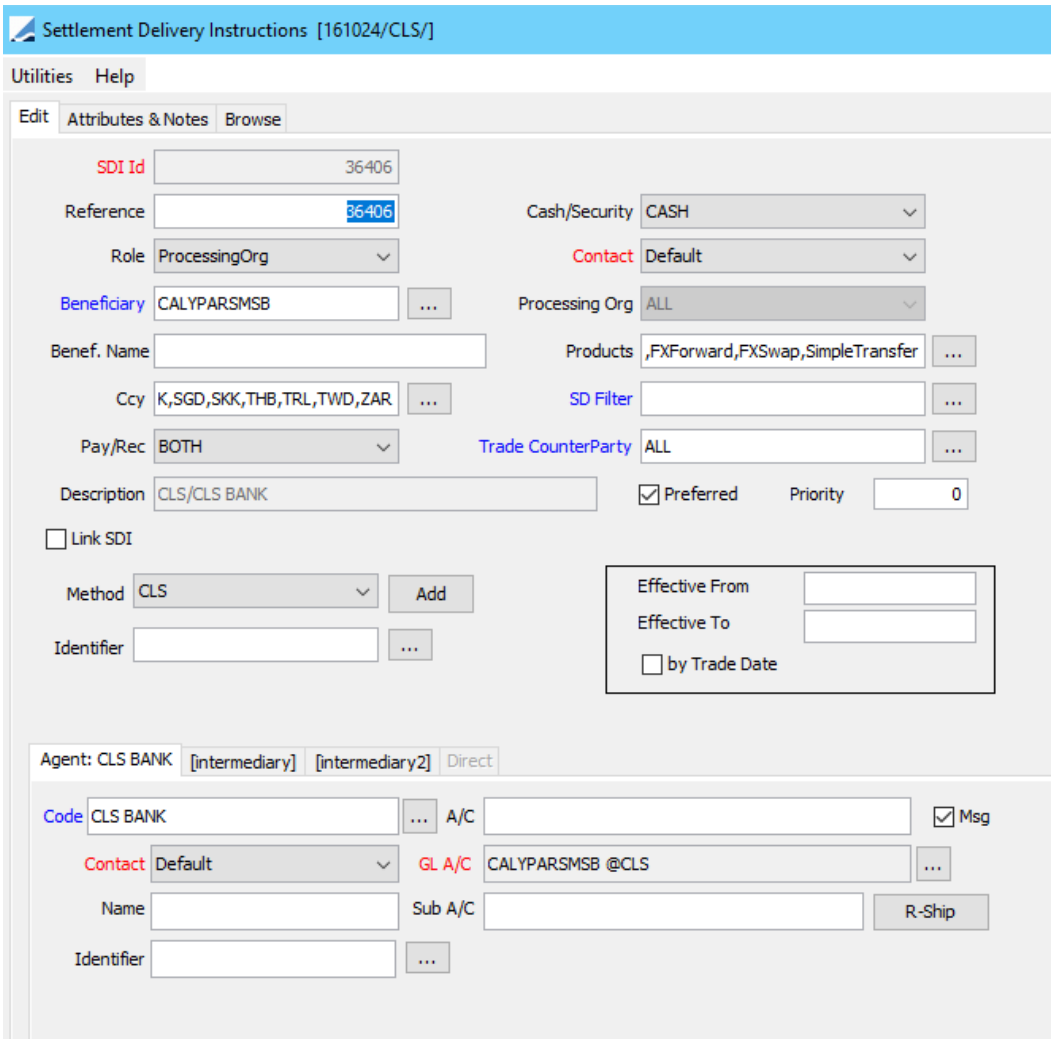
Contact

For the regular CLS FX transactions there needs to be a Swift address for contact. Message setup is described later in the document.

Settlement Instructions

For the Submission PO you only need to set up one type of SDIs. The last three mentioned for the Control Branch above do not apply as 'payments' for Pay-In Schedules are not done with the Submission PO and neither are I/O swaps.

Again, exactly like the SDI created for the Control Branch, using CLS Settlement Method, create a Pay and a Receive (or both) for Submission PO for Security 'Cash.' This SDI will be used for trades done with Counterparties using CLS settlement Method. The method to use here is CLS and the Agent also CLS Bank. Again, currencies in which payments would be made and the product for which it will be used HAVE to be defined here as per examples below.

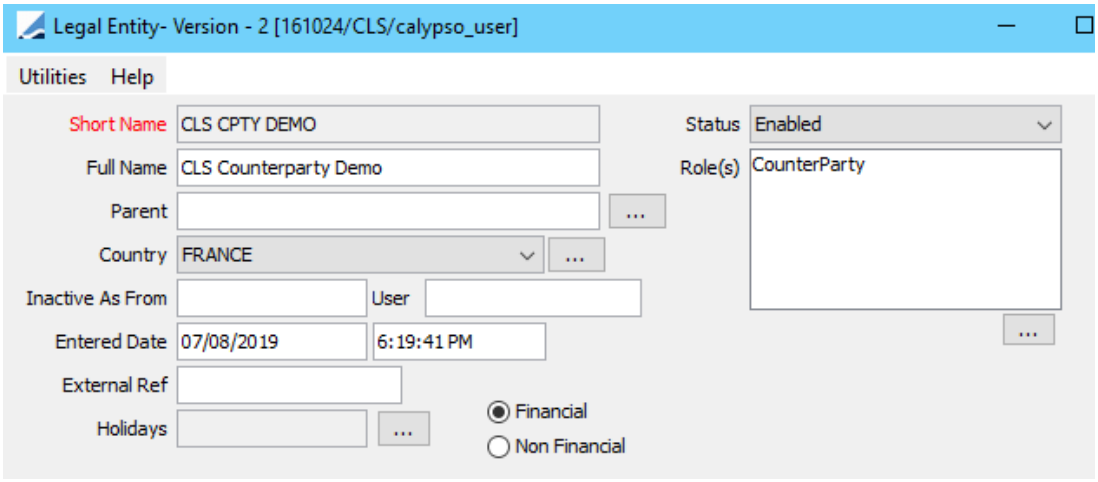


3.1.3 Counterparty

For each counterparty that may settle via CLS or that may be designated an I/O Swap with Near leg in CLS and Far leg out of CLS, the set up should be the following:

- Role = Counterparty

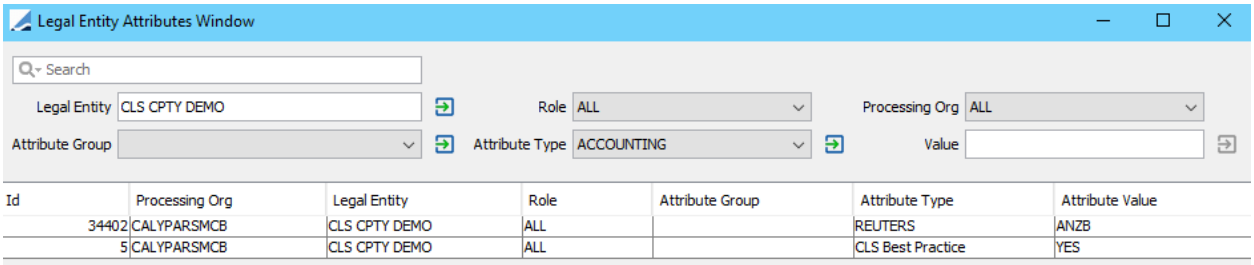
From the Calypso Navigator, navigate to Configuration > Legal Data > Entities.



Attributes

You need to set up the following Legal Entity Attributes:

- CLS Best Practice - YES/NO
- REUTERS (This serves to identify the Legal Entity when I/O swaps are fed in)



Contact

Both for CLS FX trades and I/O swaps where the leg is in CLS contact type will need to have a Swift address. Message setup is described later in the document for the relevant MT300 and MT304.

Settlement Instructions

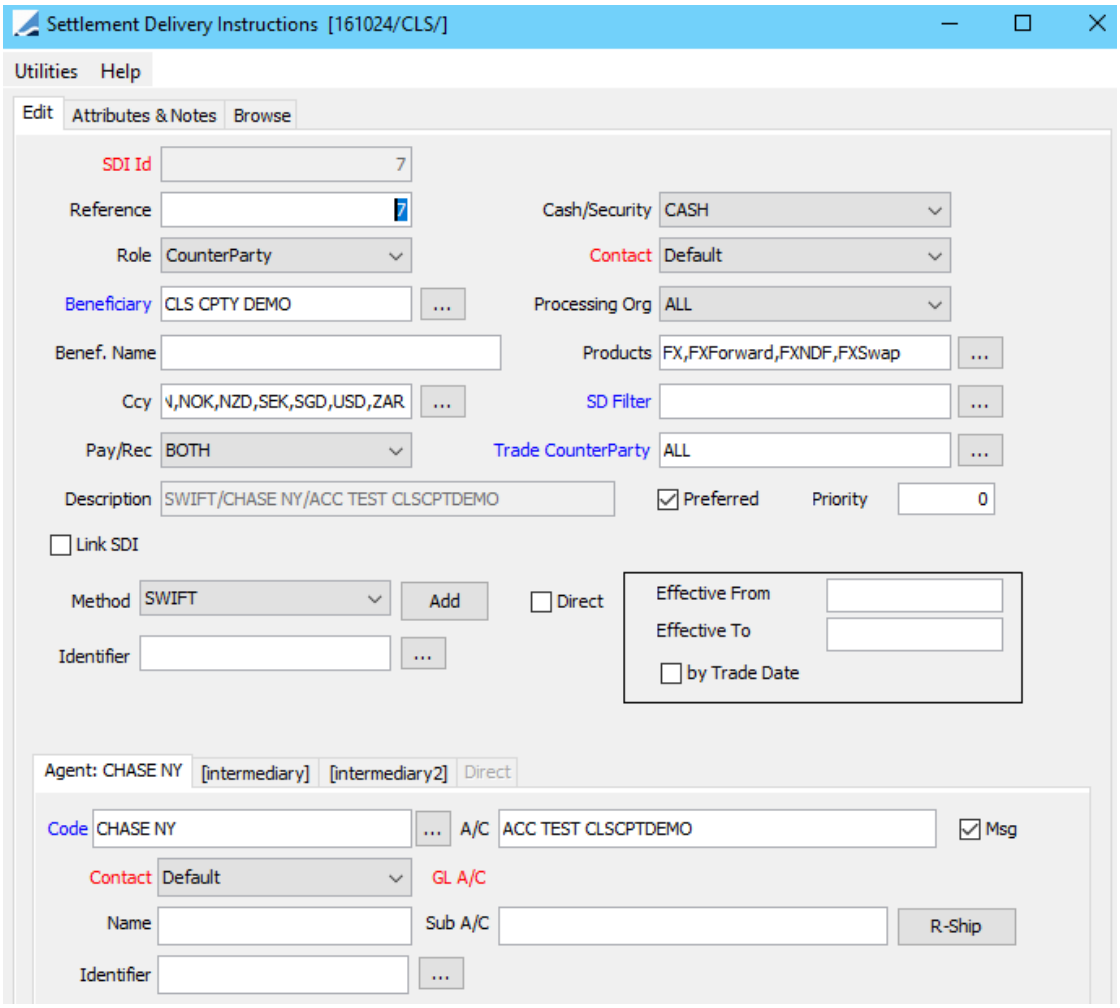
As the Counterparties will be used to either trade trades using CLS settlement method or to clear a leg of an I/O Swap extracted from CLS two types of SDIs need to be defined.

- 1st type: 'Method' as desired / 'Agent' as desired
- 2nd type: 'Method' CLS / 'Agent' CLS Bank

1st Type - 'Method' as desired / 'Agent' as desired

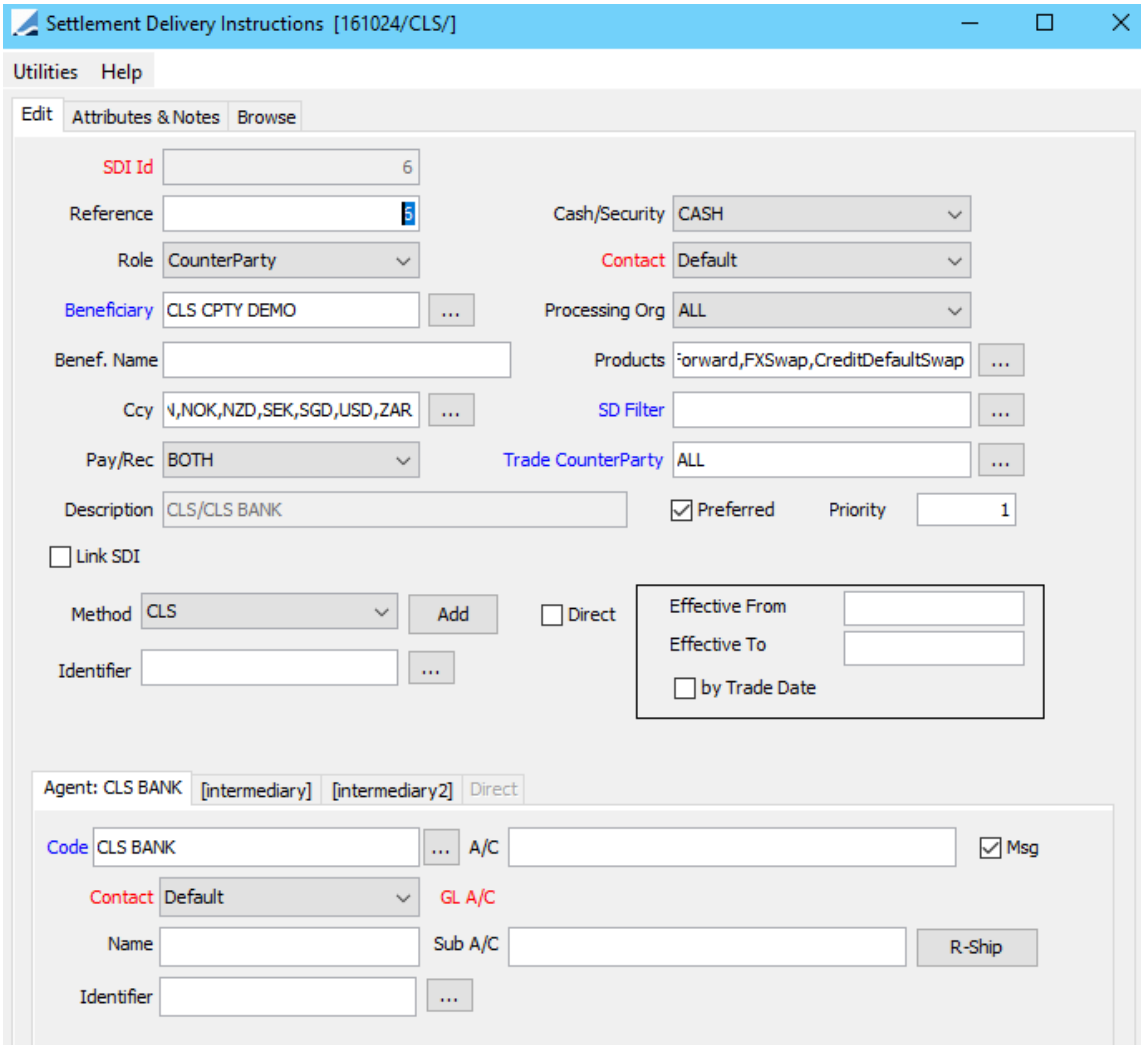
Create a Pay and a Receive (or both) for Counterparty for Security 'Cash.' This SDI will be used to settling the Far leg outside of CLS for the I/O Swaps. The method to use here is anything but CLS and the Agent again anything but CLS.

Currencies for which this SDI applies should be defined as part of the SDI and the product for which it will be used for. This SDI should have the lowest priority.



2nd Type - 'Method' CLS / 'Agent' CLS Bank

Create a Pay and a Receive (or both) for Counterparty for Security 'Cash.' This SDI will be used for trades done against POs using CLS settlement Method. The method to use here is CLS and the Agent also CLS Bank. Again, currencies in which payments would be made and the product for which it will be used HAVE to be defined here.

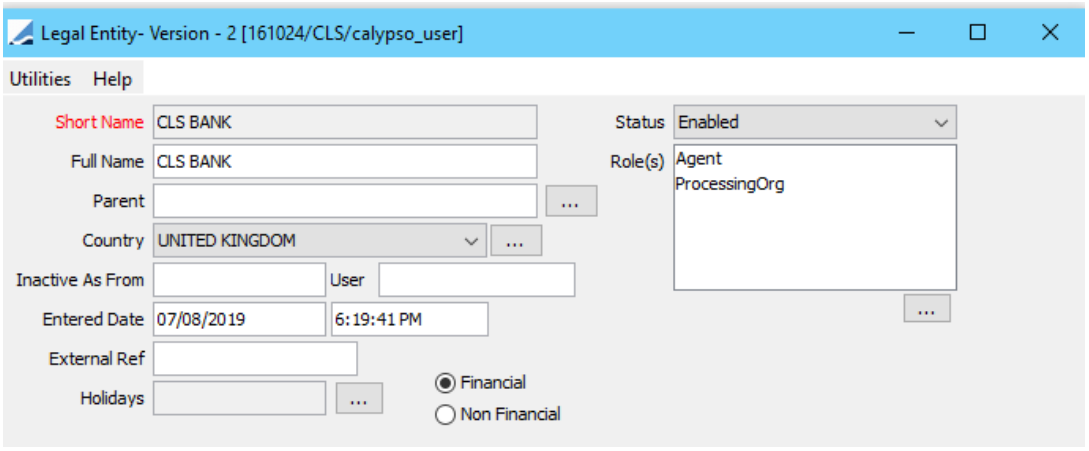


3.1.4 CLS Bank

A Legal Entity must be set up for the CLS Bank as follows:

- Role = 'Agent'
- Short name = 'CLS BANK'

The role Agent is needed for all the trades done with other Counterparties that will be settling via CLS.

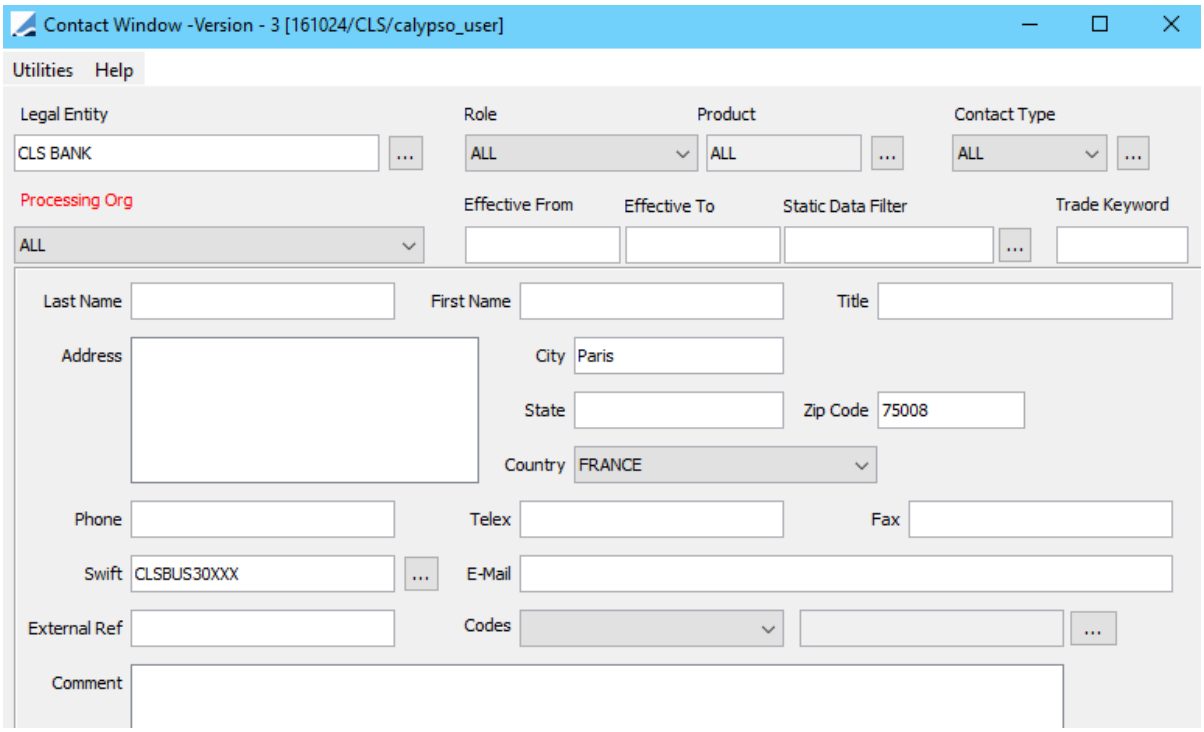


The short name can be different from CLS BANK, in which case, it must be specified in domain “clsParameters”:

- Value = clsLECode
- Comment = Short name of CLS BANK.

Contact

Contact for CLS Bank needs to be set up with Swift BIC address for MT202 messages and acknowledgement messages. Message setup is described later in the document.



The logic for the generation of the ACK message by the CLS Message Engine referring to the BIC Code to use will be as follow:

1) CLS BANK Contact Type = CLS

2) CLS BANK Contact Type = ALL

3) CLS BANK Contact Type = default

Settlement Instructions

As CLS Bank is only used as an Agent, an SDI definition is not necessary.

3.1.5 SDI Selector

The environment property CLS_SDI_SELECTION (default is false) can be set to true to treat the CLS keywords on a trade as if the SettleMethod keyword was set. This means that if the keywords are present, only CLS SDIs will be automatically assigned.

For the normal SDISelector, a domain "settlemethodKWRestricted" now exists which holds a list of settlement methods that are only (automatically) assigned if the SettleMethod keyword has the corresponding value. "CLS" should be in this domain if previously the FX*SDISelector classes from calypsox were used.

For the "Alternate" SDI selector, this aspect can be defined as part of the settlement method in the SettlementMethodWindow.

Example

If the CLS instruction has priority = 0 and the SWIFT instruction has a priority = 1, then the system selects by default the CLS instruction. Otherwise, the SWIFT instruction is chosen regardless the legal agreement configuration.

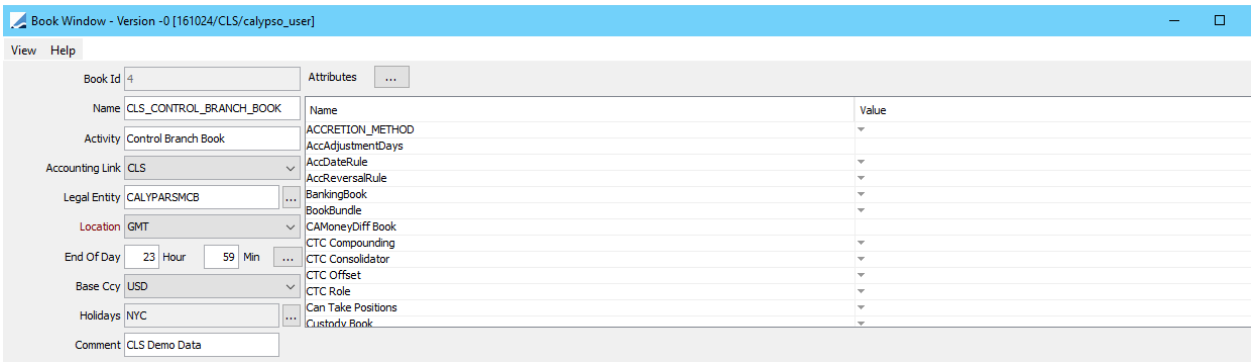
If the CLS instruction has a lower priority than the SWIFT instruction BUT the environment property CLS_SDI_SELECTION is set to true, then the system selects by default the CLS instruction.

3.2 Books

The payments for the CLS due amounts in Calypso are automatically generated when RPIS is loaded or when Pay-In call is loaded. They are saved in the form of Transfer Agent trades. As a result you will need to setup a book for this, which will belong to the Control Branch PO.

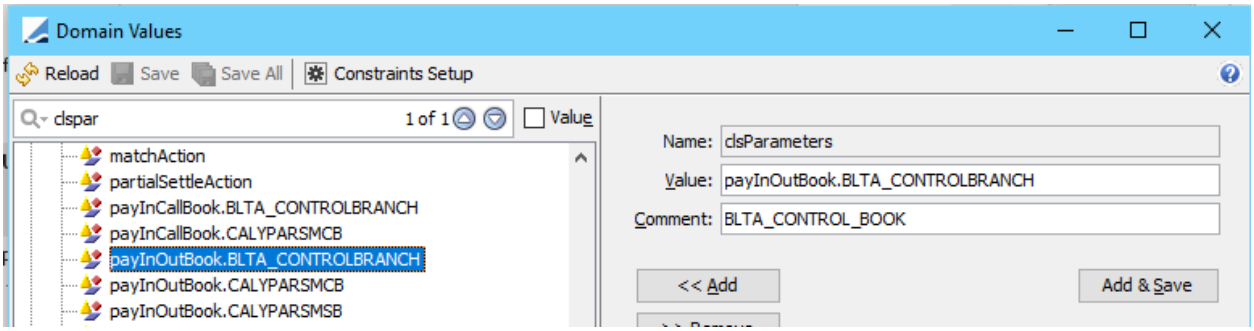
For the trades which will be generated once a RPIS has been loaded and reconciled, you will need to have the following Set-up.

From the Calypso Navigator, navigate to Configuration > Books & Bundles > Trading Books.



This book then needs to be defined as a comment in domain “clsParameters”:

- Value = payInOutBook.<legal entity short name of control branch>
- Comment = <book>



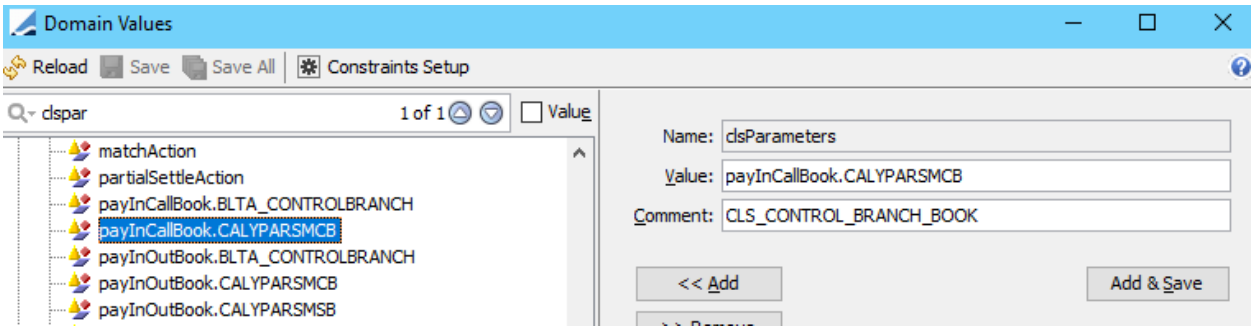
Once the Revised Pay-In Schedule has been reconciled Calypso will look up the book defined in the above domain and automatically generate the necessary Transfer Agent trades.

Once the Pay-In Call information is available from CLS, Calypso can download this information and automatically generate the relevant Pay-In call trades. If desired, the same book as for the Pay-In Schedules can be used to.

However if you would like to use a different book than the above, then you can specify by going to domain values clsParameters and adding the value “payInCallBook.<legal entity short name of control branch>” and under comment specifying the exact book.

Otherwise, should you wish to restrict the automatic generation of Pay-In calls altogether, then in the domain values “payInCallBook.<legal entity short name of control branch>”, the comment needs to be NONE (not case sensitive) as shown below.

Restricted or not, when Pay-In call information is extracted from CLS calypso updates the CLS Message Report logging the detailed information for the Pay-In and also creates an exception of type EX_CLS_PAY_IN_CALL in the task station with the relevant information.



3.3 Messages

For CLS eligible trades, Trade confirmations, Payment Messages and Payment Statements need to be defined. In addition to defining a way to hold back the generation of Settlement Messages for FX, FX Forward and FX Swap

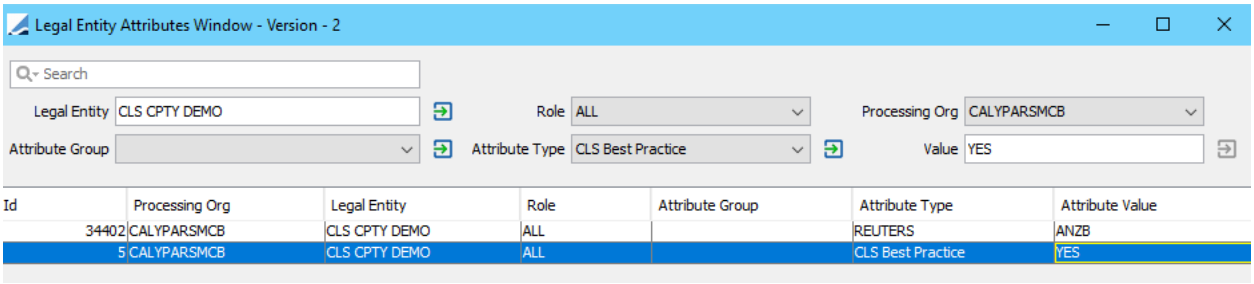
trades that are CLS eligible for those using Calypso as a CLS Interface the ability to configure Payment/Statement messages for Pay-In Schedule trades and Pay-In calls has also been introduced.

3.3.1 CLS Best Practice

Within Calypso, Counterparties that are settling through CLS i.e., have CLS defined as settlement method need also to define a Legal Entity Attribute “CLS Best Practice” YES or NO in order to comply with CLS regulations for Best Practice.

The regulations require that if a trade is done with a counterparty that is defined as “CLS Best Practice =YES” the only message to send to CLS is MT304. However, if a trade is done with a counterparty that is defined as “CLS Best Practice = NO” then in addition to sending an MT304 to CLS an MT300 needs to also be sent to those counterparties.

From the Calypso Navigator, navigate to Configuration > Legal Data > Attributes.



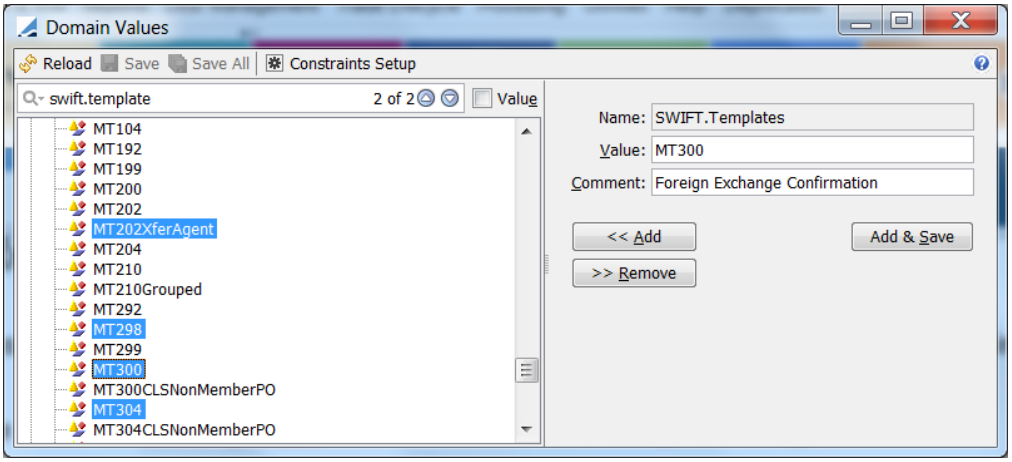
3.3.2 Message Related Domain Values

In order to configure your Messages correctly you first need to add message types and Swift templates in your domain Values.

Add the following values to the domain “messageType”: CLSCONFIRM and CLS_NOSTRO_ADVICE.

In domain “Swift.Templates” make sure that MT202xferAgent, MT300, MT304, MT298 exist.

From the Calypso Navigator, navigate to Configuration > System > Domain Values.



3.3.3 Static Data Filters

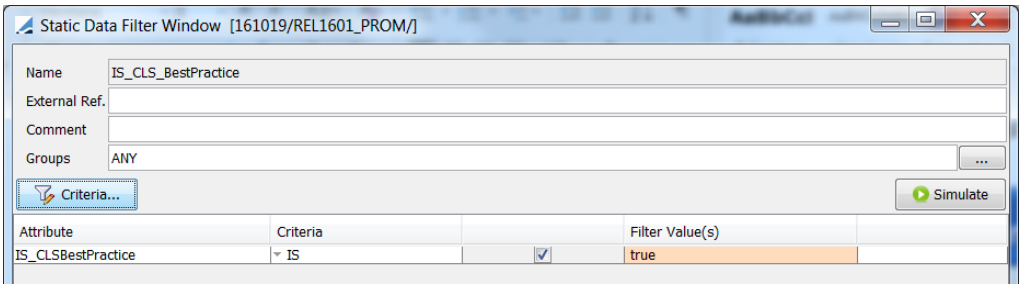
Define Static Data to identify CLS transactions as shown below.

From the Calypso Navigator, navigate to Configuration > Filters > Static Data Filter.

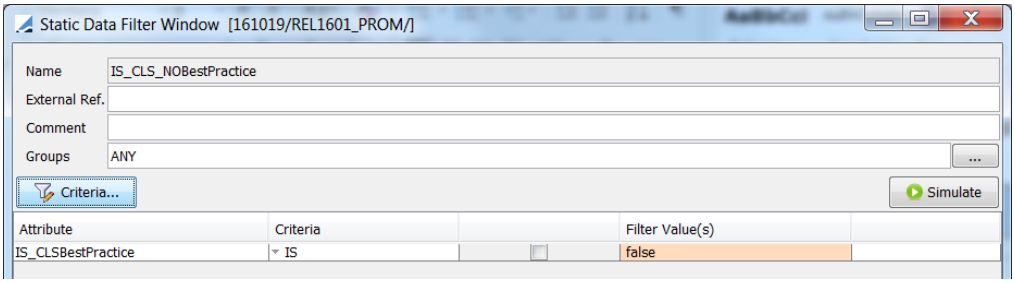
IS_CLS attribute - This SD Filter attribute checks for 'true' in the keyword 'CLS' for the FX Spot and Forward trades, and 'NEAR_CLS' and 'FAR_CLS' for each leg of the FX Swaps.

IS_CLSBestPractice

This SD filter checks trade keywords then the Legal Entity Attribute CLS Best Practice if the counterparty is defined with a Legal Entity attribute CLS Best Practice YES or NO.

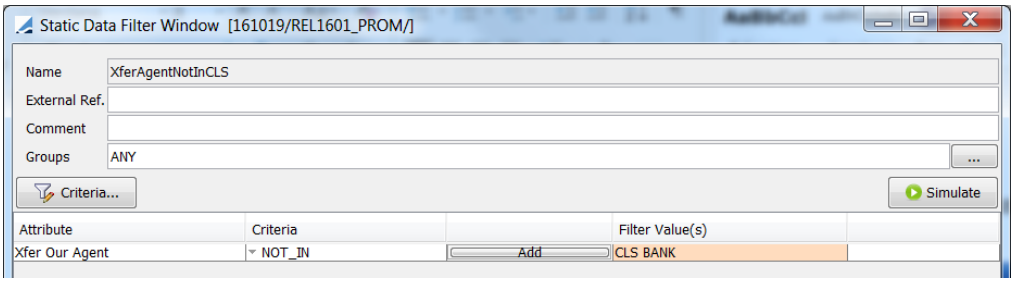


Not_CLS_BestPractice



Xfer Our Agent – NOT_IN

This SD Filter needs to be set up to avoid the generation of Settlement messages for your FX, FX Forward and FX Swap trades when your trade is CLS eligible.



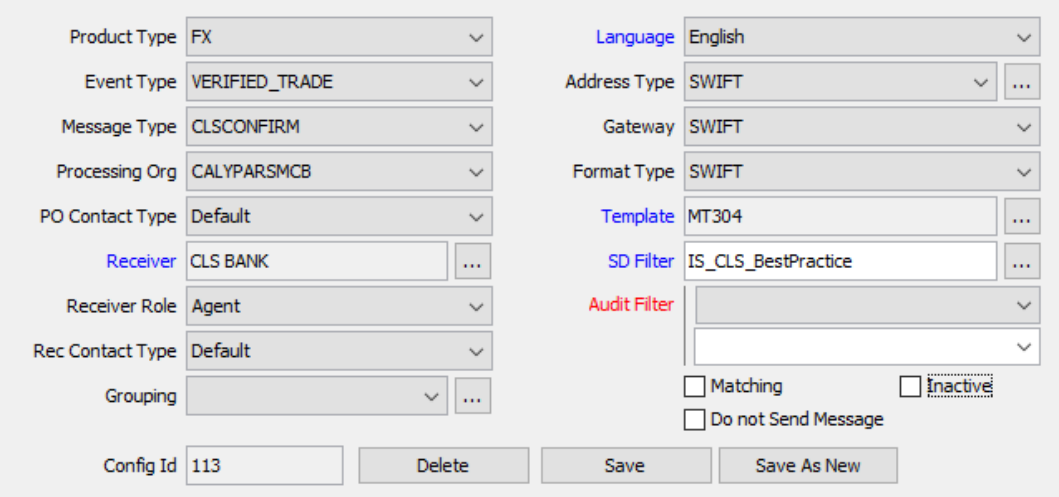
3.3.4 Trade Confirmation

Having defined your Legal Entity Attribute and set up your Static Data Filters you can now define your Message Configuration setup for CLS trades where Counterparties are either Best Practice or not.

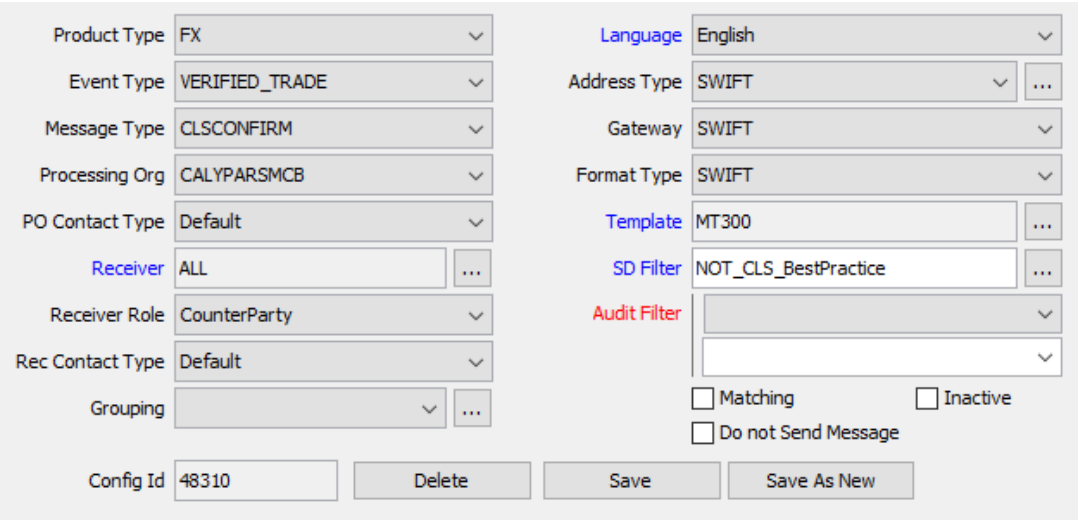
For each product type FX, FX Forward and FX Swap define two message configurations for Event Type: VERIFIED_TRADE attaching the relevant SD filter in order to generate an MT304 or an MT304 and an MT300. Respectably you should also have a CANCELED_TRADE per VERIFIED_TRADE set-up.

From the Calypso Navigator, navigate to Messages & Matching > Message Set-up.

MT304 Message Type CLSCONFIRM



MT300 Message Type CONFIRM – Member PO



FX MX Messages

Domain Values

Add the following domain values.

Domain “gateway”:

- MX

Domain “formatType”:

- MX

Domain “MX.Templates”:

- FXTradeInstruction.selector
- fxtr.014.001
- fxtr.015.001
- fxtr.016.001

CBPR+ SR 2023

If domain “USE_SR_2023” contains Value = TRUE and domain "MXUseCBPR2023" contains Value = TRUE, the following changes apply:

Changes to FXTradeInstructionDetails:

Removed:

TradAmts/TradgSdBuyAmt/@Ccy

TradAmts/TradgSdSellAmt/@Ccy

Added:

TradAmts/TradgSdBuyIAmt/Amt

TradAmts/TradgSdBuyAmt/Amt/@Ccy

TradAmts/TradgSdSellAmt/Amt

TradAmts/TradgSdSellAmt/Amt/@Ccy

The following message templates are impacted:

fxtr.014.001.05

fxtr.015.001.05

fxtr.016.001.05

Changes to fxtr.017.001.05 integration:

“Amounts.Buy Amount” is taken from TradAmts/TradgSdBuyAmt/Amt

“Amounts.Buy Currency “ is taken from TradAmts/TradgSdBuyAmt/Amt/@Ccy

“Amounts.Sell Amount” is taken from TradAmts/TradgSdSellAmt/Amt

“Amounts.Sell Currency” is taken from TradAmts/TradgSdSellAmt/Amt/@Ccy

Changes to camt.054.001.08 integration:

Ntfctn/Ntry/NtryDtls/TxDtls/Refs/UETR is stored in message attribute LinkedUETR

Ntfctn/Ntry/NtryDtls/TxDtls/RltdAgts/InstgAgt/FinInstnId/BICFI is stored in message attribute InstructingAgentBIC

Ntfctn/Ntry/NtryDtls/TxDtls/RltdAgts/InstdAgt/FinInstnId/BICFI is stored in message attribute InstructedAgentBIC

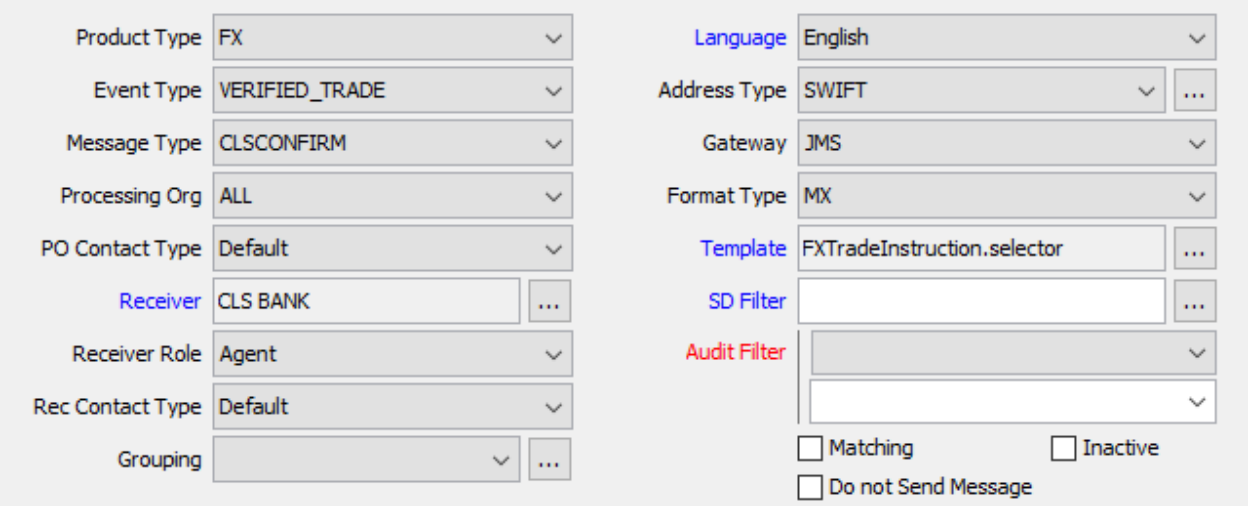
Message Setup

If you want to use the FX MX messages instead for CLSCONFIRM configurations, you need to replace the MT304 template with the “FXTradeInstruction.selector” template. It is part of the domain “MX.Templates”.

It generates the following messages:

- New Trade – “fxtr.014.001”
- Amend Trade – “fxtr.015.001”
- Cancel Trade – “fxtr.016.001”

Example for FX:



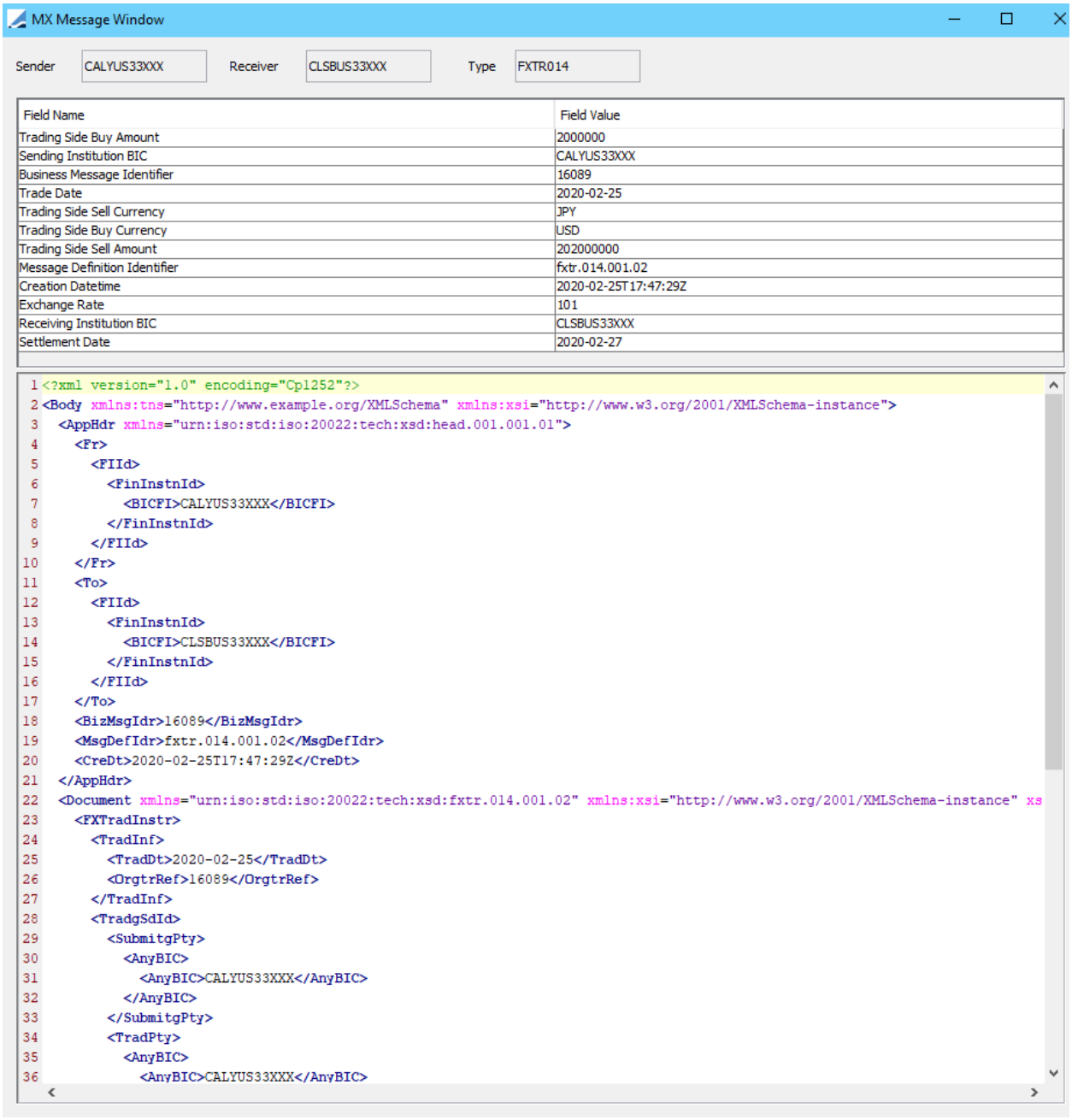
We recommend using the JMS Gateway in order to choose the output file format: CLS requires a binary file.

To use the JMS Gateway in binary mode, you need to add isBinary=True in the properties file of the gateway.

For example: calypso_JMS_config.properties

```
# Import Message Engine (MQ)

input.queue.name=dynamicQueues/input1
dynamicQueues/input.queue.setContext=true
isBinary=True


# Sender Engine (JMS)

#For ActiveMQ
jms.url=tcp://localhost:61616
jms.modetypeclass=org.apache.activemq.jndi.ActiveMQInitialContextFactory
jms.queue.connectionFactory=ConnectionFactory

#For IBMMQ
#messagingPlatform=IBMMQ
#jms.url=file://localhost/c:/tools/ibm/mqs/binding
#jms.modetypeclass=com.sun.jndi.fscontext.RefFSContextFactory
#jms.queue.connectionFactory=QueueConnectionFactory

output.queue.name=dynamicQueues/output1
dynamicQueues/output.queue.ackType=auto
dynamicQueues/output.queue.persist=true
dynamicQueues/output.queue.transacted=false

#IBM SSL Support
#sslVersion=SSLV3
#sslKeystore=<KeyStore certificate>
#sslKeystorePassword=<KeyStore Password>
#sslTrustStore=<TrustStore Certificate>
#sslTrustStorePassword=<TrustStore Password>
```

Sample message “fxtr.014.001”:

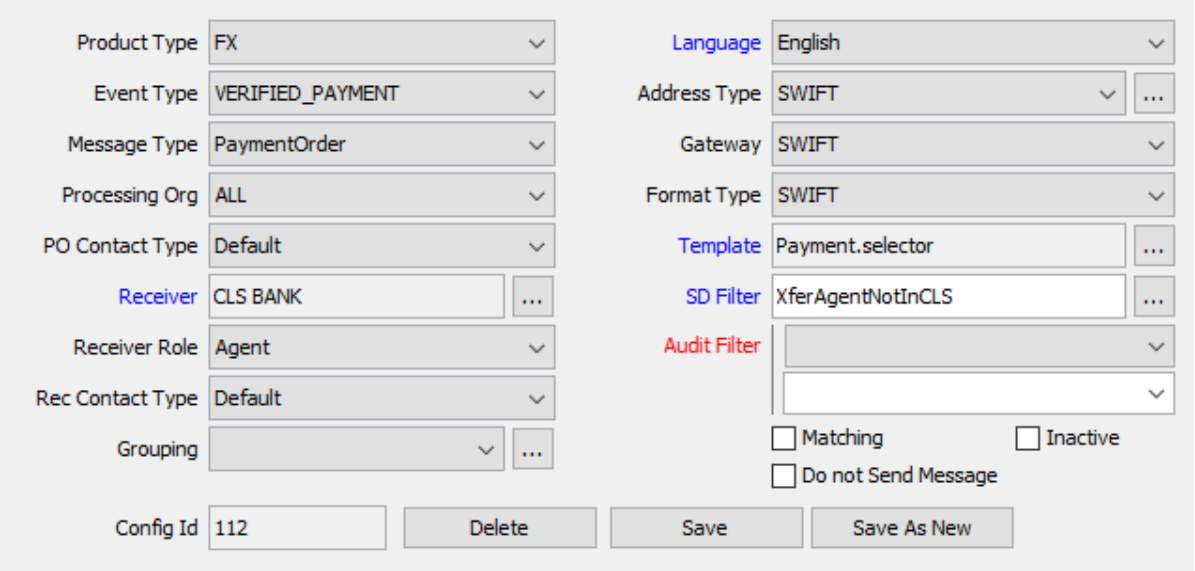

MX Message Window
Sender
CALYUS33XXX
Receiver
CLSBUS33XXX
Type
FXTR014
Field Name
Field Value
Trading Side Buy Amount
2000000
Sending Institution BIC
CALYUS33XXX
Business Message Identifier
16089
Trade Date
2020-02-25
Trading Side Sell Currency
JPY
Trading Side Buy Currency
USD
Trading Side Sell Amount
202000000
Message Definition Identifier
fxtr.014.001.02
Creation Datetime
2020-02-25T17:47:29Z
Exchange Rate
101
Receiving Institution BIC
CLSBUS33XXX
Settlement Date
2020-02-27
<?xml version="1.0" encoding="Cp1252"?>
<Body xmlns:tns="http://www.example.org/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<AppHdr xmlns="urn:iso:std:iso:20022:tech:xsd:head.001.001.01">
<Fr>
<FIId>
<FinInstnId>
<BICFI>CALYUS33XXX</BICFI>
</FinInstnId>
</FIId>
</Fr>
<To>
<FIId>
<FinInstnId>
<BICFI>CLSBUS33XXX</BICFI>
</FinInstnId>
</FIId>
</To>
<BizMsgIdr>16089</BizMsgIdr>
<MsgDefIdr>fxtr.014.001.02</MsgDefIdr>
<CreDt>2020-02-25T17:47:29Z</CreDt>
</AppHdr>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:fxtr.014.001.02" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xs
<FXTradInstr>
<TradInf>
<TradDt>2020-02-25</TradDt>
<OrgtrRef>16089</OrgtrRef>
</TradInf>
<TradgSdId>
<SubmitgPty>
<AnyBIC>
<AnyBIC>CALYUS33XXX</AnyBIC>
</AnyBIC>
</SubmitgPty>
<TradPty>
<AnyBIC>
<AnyBIC>CALYUS33XXX</AnyBIC>

3.3.5 Matching of Trade Confirmation

In the event that tag 20 of the MT304 is customized, then the CustomCLSMessageFinder needs to be implemented. This will find the correct corresponding message.

3.3.6 Payment Confirmation

You can avoid generating your Settlement Messages if your trade is CLS eligible for FX, FXForward and FXSwap products. This is set up as a Static Data filter as mentioned above (xfer Our Agent NOT_IN value set to false) applied to the VERIFIED_PAYMENT messages per product FX, FXForward and FXSwap.

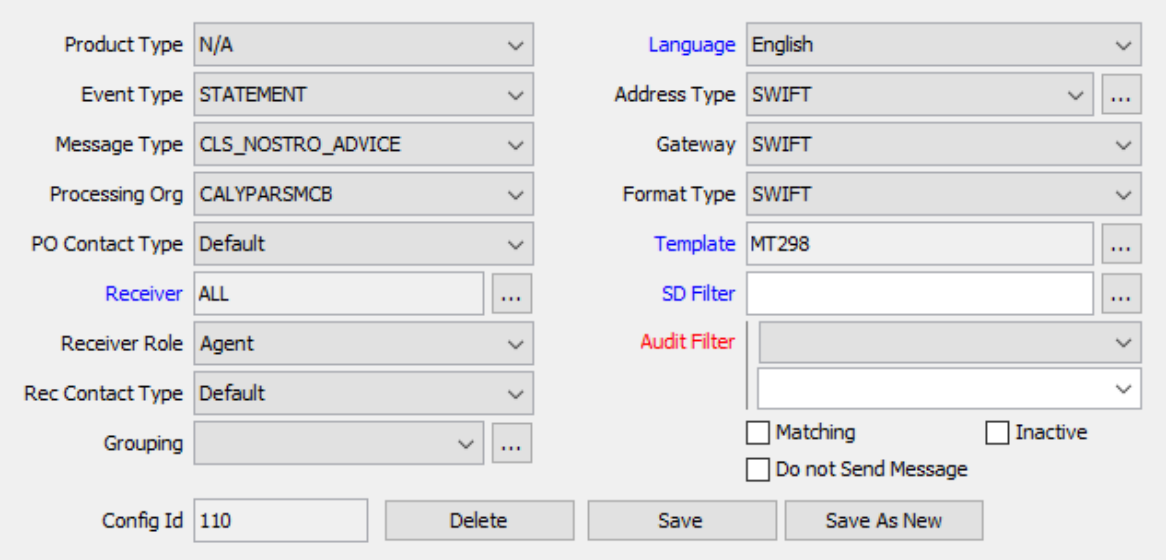


3.3.7 MT298 Payment Statement

To accommodate the CLS Interface development you now need to enhance your Message set-up to include a set up for payment statement MT298.

Having added in your message type domain name CLS_NOSTRO_ADVICE as explained above you can now set up your CLS_NOSTRO_ADVICE message type as below.

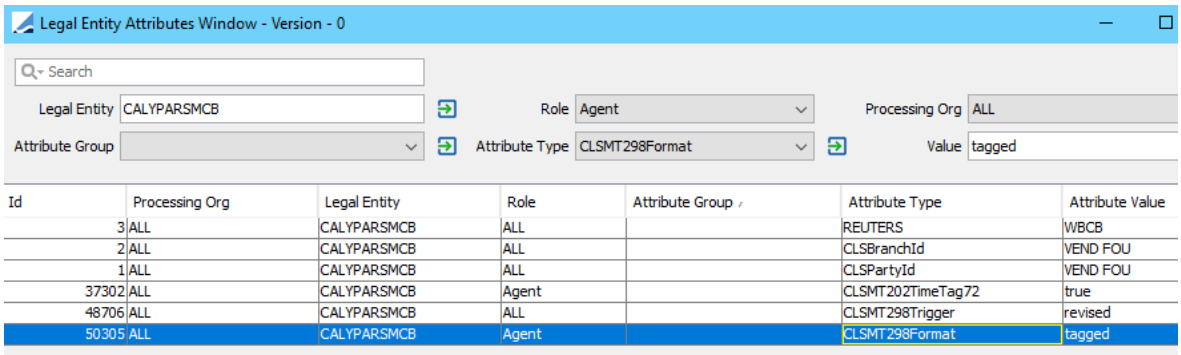
- Product Type – 'N/A'
- Event Type – STATEMENT
- Message Type – CLS_NOSTRO_ADVICE
- PO – Control Branch
- Format Type – SWIFT
- Receiver – 'ALL' for normal cases
- Receiver Role – Agent
- Template – MT298 - You must make sure this template is available.



Generally, MT298 Statements are generated after each Pay-In Schedules are extracted from CLS. However, via the Legal Entity Attributes CLSMT298Format and CLSMT298Trigger, Calypso allows you to define more specifically the conditions in which this message is generated according to your business needs.

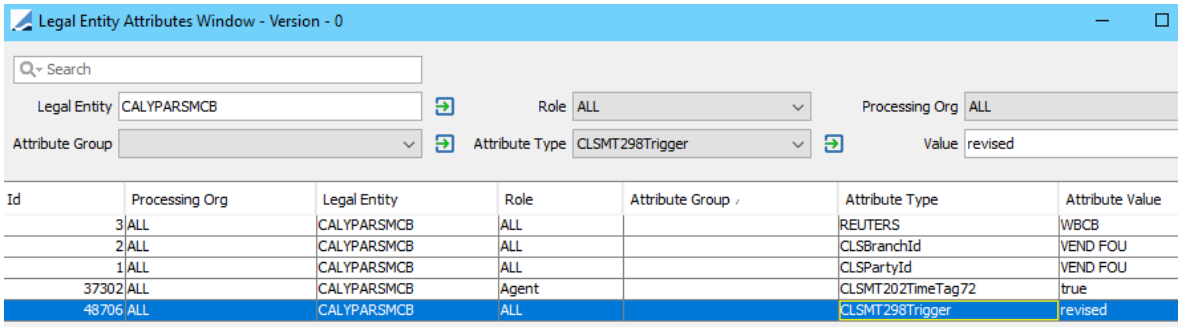
If you would like to control the MT298 message generation (to define if and for which Pay-In Schedules an MT298 is generated and also if generated, then what the contents of the message are) then for your Nostro 'Agent' define legal entity attributes CLSMT298Format and CLSMT298Trigger as needed:

- CLSMT298Format: If value is set to "tagged", then an MT298 with tags ":20:", ":21:", ":13C:", ":32B:" will be generated for Pay-Ins and tags ":20:", ":21:", ":32B:", "56A:" will be generated for Pay-Outs. If value is left blank or LEAttribute for Agent is not set then a free-text format is generated



Id	Processing Org	Legal Entity	Role	Attribute Group	Attribute Type	Attribute Value
3	ALL	CALYPARSMCB	ALL		REUTERS	WBCB
2	ALL	CALYPARSMCB	ALL		CLSBranchId	VEND FOU
1	ALL	CALYPARSMCB	ALL		CLSPartyId	VEND FOU
37302	ALL	CALYPARSMCB	Agent		CLSMT202TimeTag72	true
48706	ALL	CALYPARSMCB	ALL		CLSMT298Trigger	revised
50305	ALL	CALYPARSMCB	Agent		CLSMT298Format	tagged

- CLSMT298Trigger: Value should be left blank if MT298 is desired for all Types of Pay-In Schedules. If no MT298 is desired at all you can use a value like "none". If you would like to eliminate generation for some types and only have them generated for other types, input name of types and separate list by comma for example 'requested, revised' which will only generate MT298 for requested Pay-In Schedule and revised Pay-In Schedule or 'initial, revised' (as below) for messages for only initial and revised.

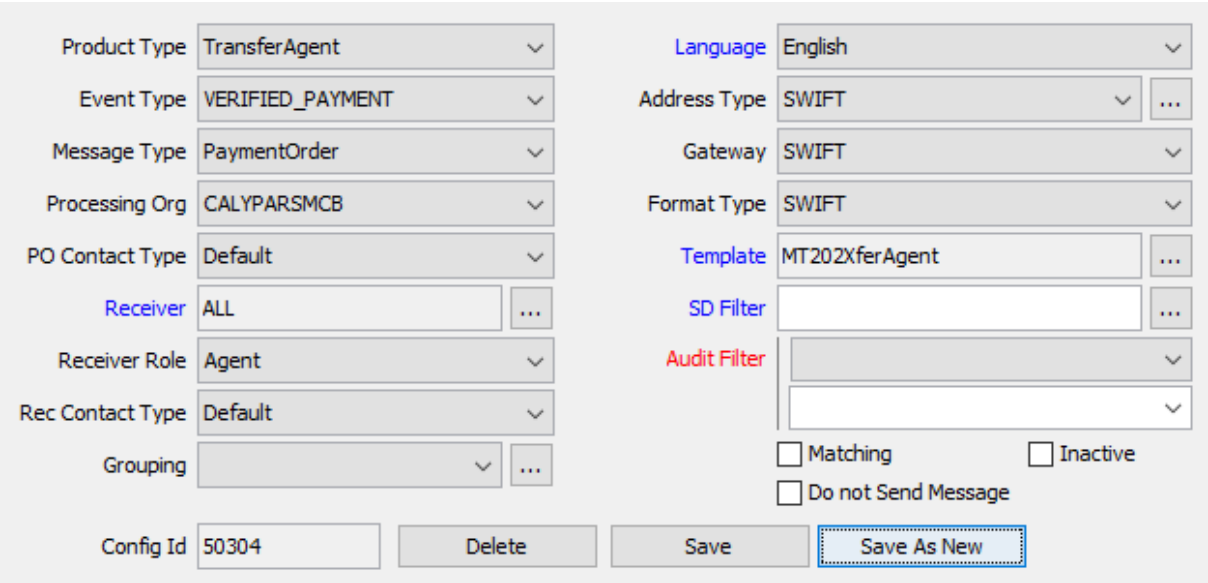


3.3.8 MT202 Payment Message

In the same way your message set up now needs to include a set-up for MT202. When transfers are generated within Calypso to pay Control Branch CLS account after Pay-In Schedules and Pay-In calls have been reconciled (via Transfer Agent trades), an MT202 payment message is generated and sent to Nostro Agents.

Format to follow should be:

- Product Type – 'TransferAgent'
- Event Type – VERIFIED_PAYMENT
- Message Type – PAYMENTMSG
- PO – Control Branch
- Format Type – 'SWIFT'
- Receiver – Agent against which the Simple transfer payments are going through
- Receiver Role – Agent
- Template – MT202XferAgent



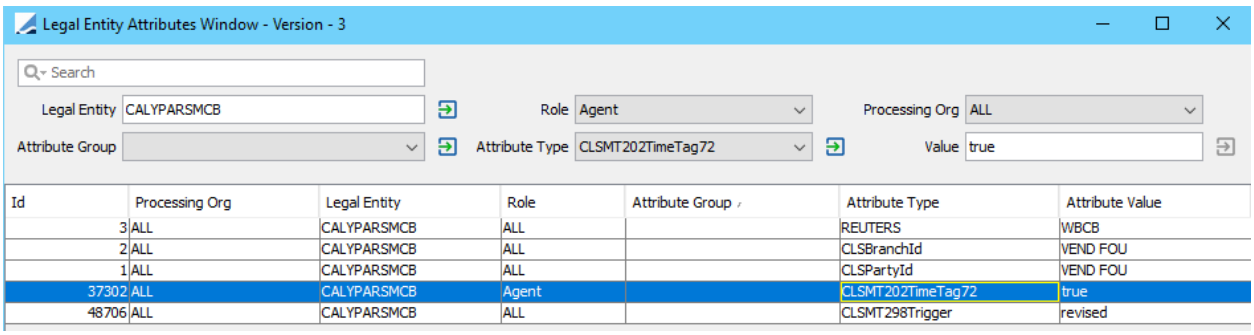
[NOTE: The template to use here is MT202XferAgent]

3.3.9 MT202 tag 13C vs. tag 72

For Pay-Ins, the Transfer Agent trades generated contain a trade keyword '13CTimeIndication' with the time when the amount should be credited on CLS accounts at the relevant central bank. The format is the same as for the MT202 message, e.g. /CLSTIME/0800+0200 indicating 8 am in the CLS timezone (during DST). This Trade keyword is copied to a transfer attribute with the same name and the MT202.xml template picks up this value (if present) and fills it into a 13C tag for the generated MT202.

MT202s for the Pay-Ins will as a result by default include 13c tag. However, in the event this information is required for tag 72 instead of tag 13C then you must indicate this at the Legal Entity Attribute level, for role 'Agent.'

Legal entity attribute CLSMT202TimeTag72.



If defined and "true", the "/CLSTIME/" information is placed in Tag 72 instead of tag 13C.

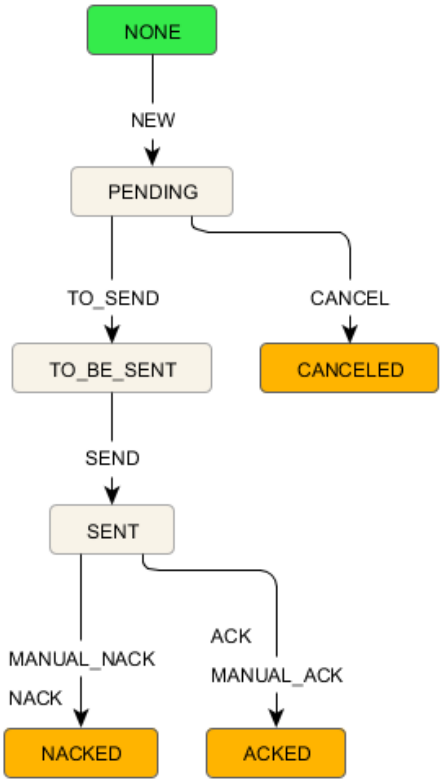
3.4 Workflow Configuration

The status of the message sent to CLS won't be updated by the status received from CLS. The status has to be monitored using CLSTradeInfo Report. The only need to have a message workflow is to send it to CLS and receive the ACK/NACK from SWIFT.

3.4.1 Message Workflow

For the products FX, FX Forward and FX Swap, you can configure a message workflow with subtype CLSCONFIRM

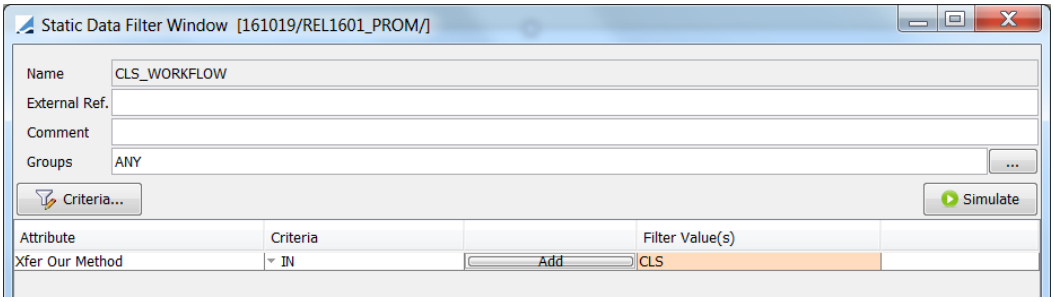
Example of CLSCONFIRM message WF:

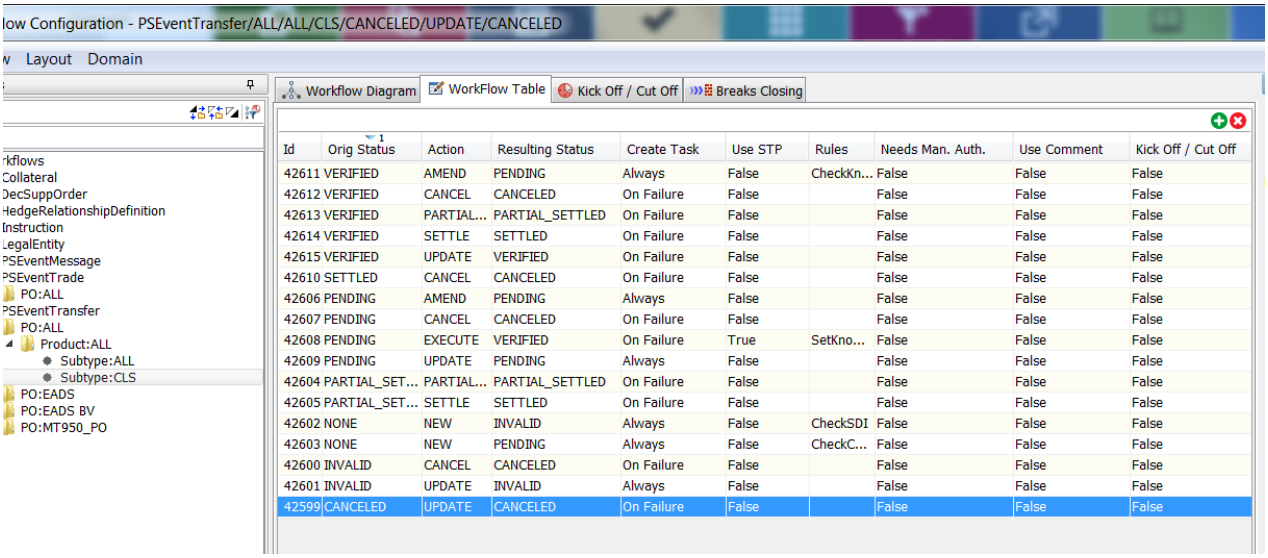


3.4.2 Transfer Workflow

For the Transfer Workflow a new xferWorkflowType for subtype CLS has been added. The workflow can be loaded from `calypso_home$/cls/src/main/resources/CLS_Transfer_Workflow.wf`.

This workflow needs to be used for CLS transfers thus a SD Filter needs to exist to detect the transfer is a CLS workflow type. As a result, as shown below add SD Filter 'xfer Our Method' value CLS on any transition.

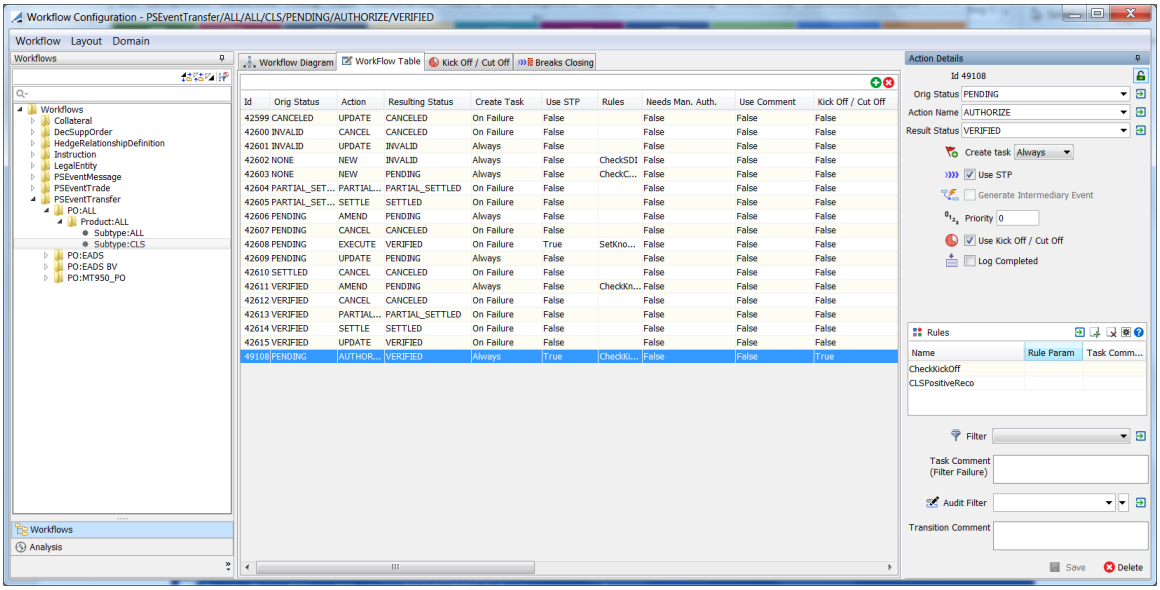


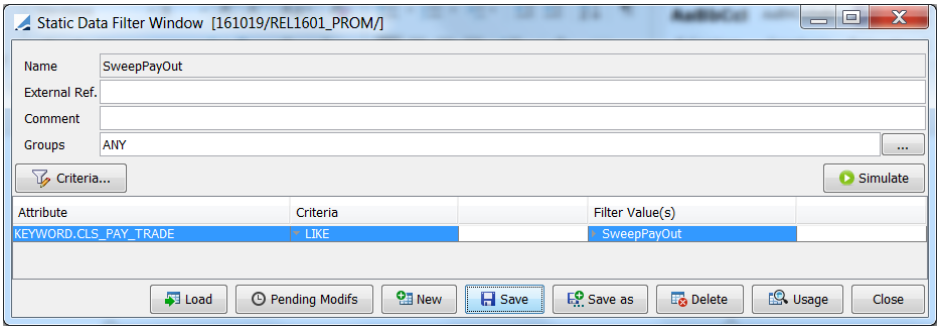


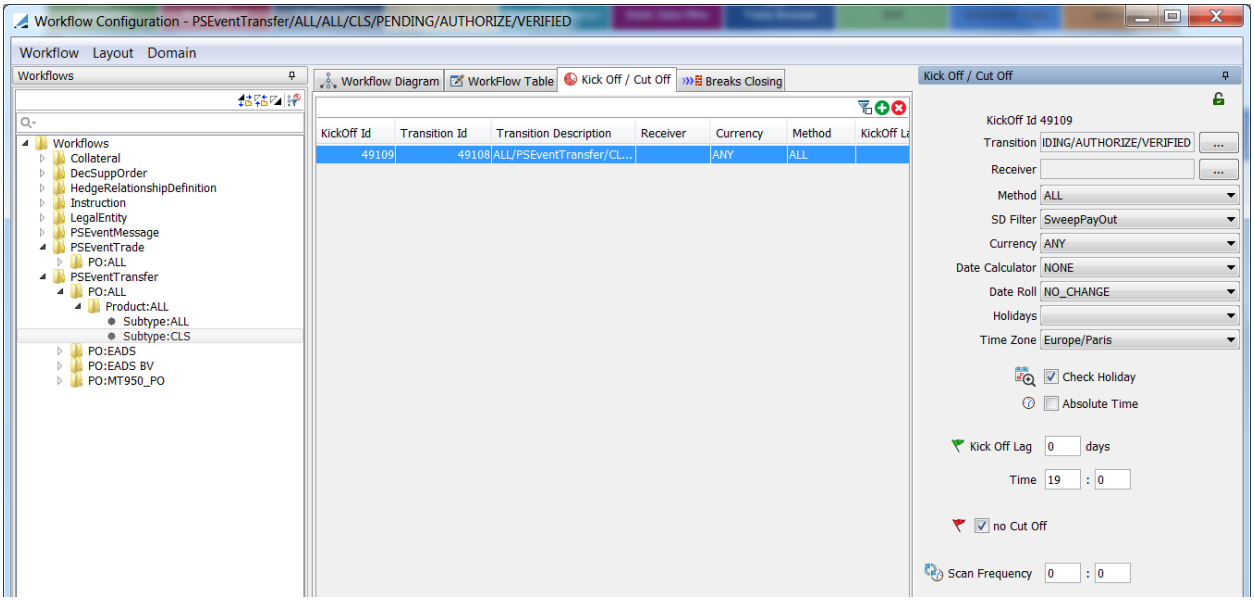
3.4.3 Kick Off/Cut Off

In the case where you have sweep accounts, if you do not want the Pay-Out call payment messages to be generated to the Nostro Agent before CLS has input payments into your Nostro CLS account then you can use the Kickoff/Cut off functionality on the transfer workflow for the Pay-Out trades. Your setup for the Kick Off/Cut Off should be set between Pending – Authorize – Verified. That way if the kick off/Cut Off time has not yet been reached, when the automatic Transfer Agent trades get generated, the transfer of the Pay Out trades will be held back in status pending.

Below is an example of setup.







3.4.4 Workflow Rules

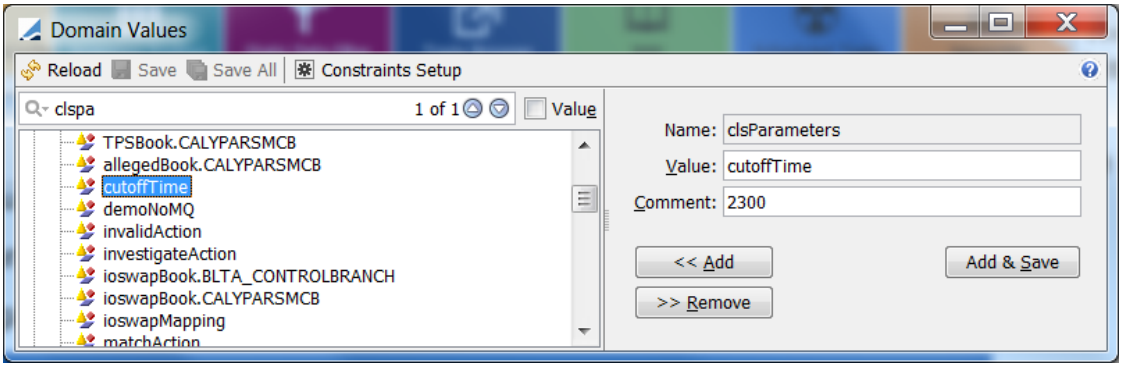
CLSCutoff

Best Practice is to cease submission of trades to CLS at CET00:00 on the day of value. Therefore, we have added the cutoff time logic in order to control submission of FX, FX Forward and FX Swap trades after that time. There is an exception to this in the event that the trade is an I/O Swap. The CLS cutoff time functionality permits this trade capture exception.

The workflow rule CLSCutoffTradeRule should be set on the first transitions of the workflow (when the trade is created) and also on all the Amend transitions (for financial amendments).

The rule works based on a configurable cutoff time which is defined in the domain “clsParameters”.

From the Calypso Navigator, navigate to System > Domain Value.



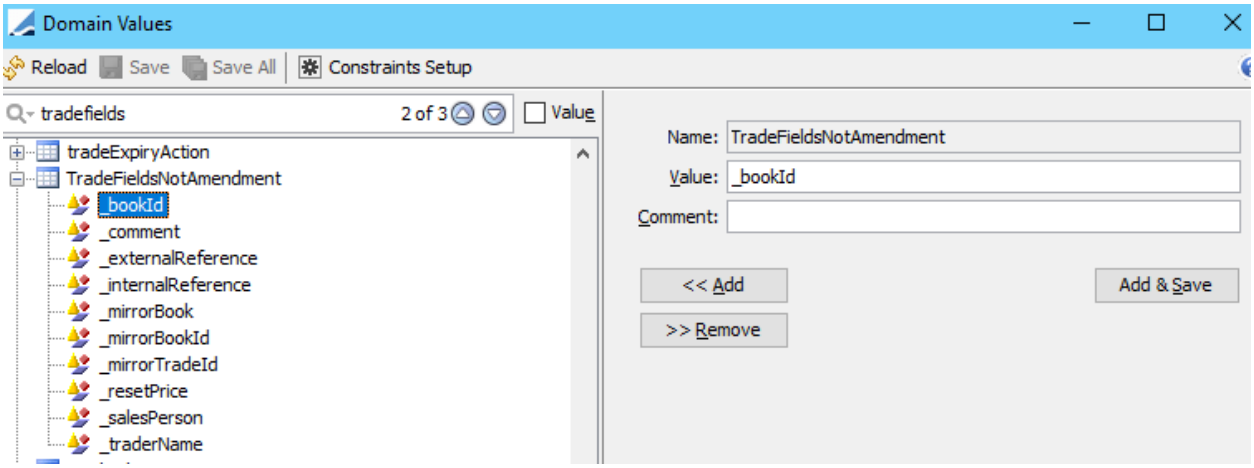
In “clsParameters” domain, define value to be ‘cutoffTime’ and set the Comment to the time at which you would like the cut-off to occur.

The format of the comment is “hhmm”. Values between “0000” and “0629” are interpreted as cutoff on the settlement date; values between “0630” and “2359” are interpreted as cutoff on the business day before settlement date.

Thus, in the above example value of “2300” prevents changes made (to the named FX product trades above) after 23:00 CET (in the CLS time zone) the business day before settlement day.

If using the old FX trade screens, then changes made after the cut-off time will generate an error warning (like the below) and in the new FX trade screens changes made will not generate warning but will move trade to non STP alternative state with a comment generated in the Task Station.

[NOTE: By default, every change causes a new message to CLS to be generated. The domain TradeFieldsNotAmendment can be filled with field names (as in the Audit of the trade) that should not cause a regeneration of confirmations for the regular FX, FX Forward and FX Swap trades]



To have this domain value automatically filled in check the CLSmsg checkbox when running Execute SQL.

This logic is shared between the MessageHandler classes and this CLSCutoff workflow rule. If you customize the MessageHandler you may have to customize this workflow rule as well.

AdvanceSettleDate/AdvanceForwardDate

In the event that a Short Notice Bank holiday occurs, there are two Trade Workflow Rules that allow you to either adjust your settle date or forward date. These rules should not be part of the main workflow but as a transitional workflow.

The rule will adjust the dates using the holidays defined in the book, if none found then holidays of the PO and if none found holidays of the trade Ccy.

[NOTE: Short notice bank holidays are not announced anywhere in CLS for any Ccy. Thus we do not feed in any information from CLS to update you on such occasions. CLS presumably contacts its relevant clients by external means]

3.5 Additional Information

3.5.1 Transfer Attributes

The following xferAttributes are available:

- CLS_SUSP_INFO – To indicate reasons for not including the transfer in settlement i.e., suspension, pending flag
- CLS_PART_STL_IDS – To indicate the id of Trade Info entries.

3.5.2 Transfer Netting

For CLS Interface we do not recommend netting transfers.

3.5.3 Single Payment Tolerance

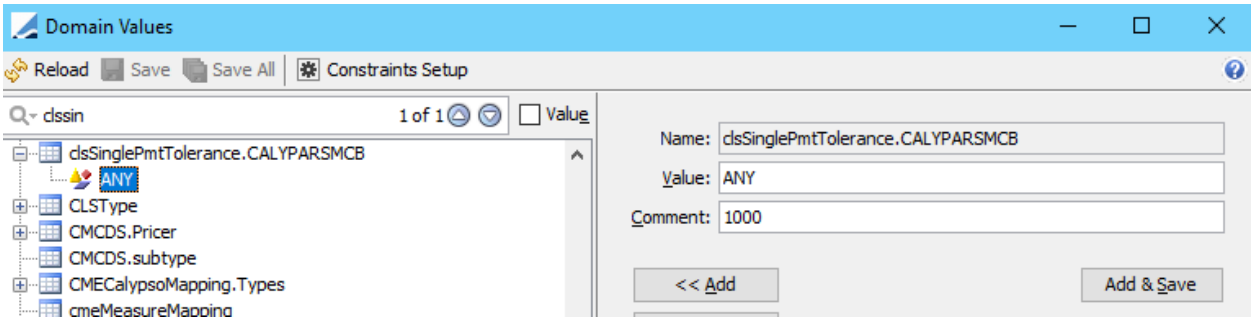
You can define if you would like to generate a single payment to CLS if the overall obligation is less than a certain amount. This is even if the Pay-In schedule contains payments in several portions.

From the Calypso Navigator, navigate to System > Domain Values.

Add the domain “clsSinglePmtTolerance.<legal entity short name of the Control branch>”.

As a value you can input “Any” or a 3 letter Ccy code if you want to define payment tolerance per Ccy. In the comments of this domain name input the amount cutoff you will begin to allow the generation of the single payment

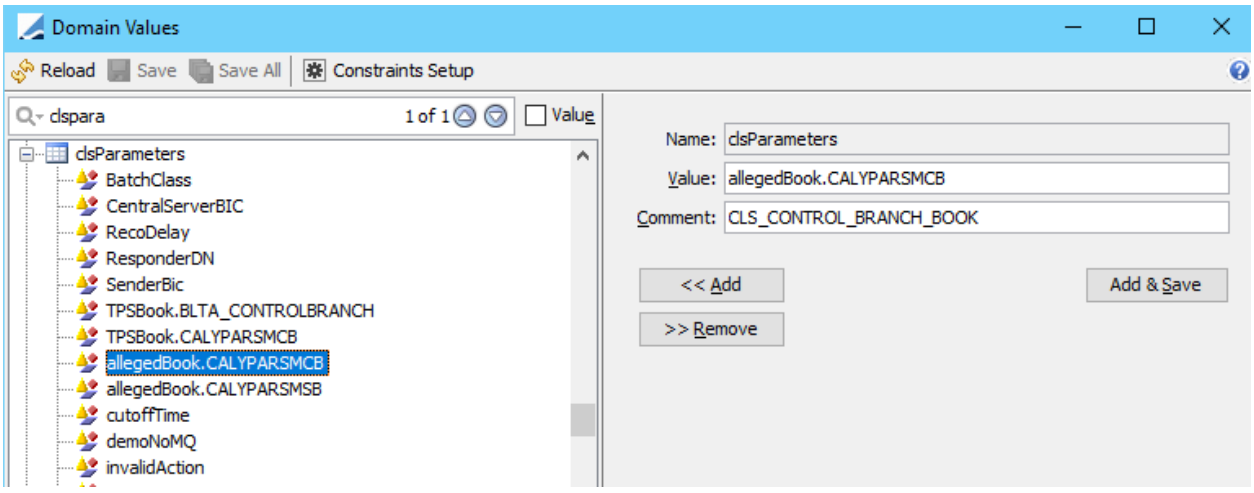
Example: If overall obligation to CLS is less than 500 then I will generate a single payment.



3.5.4 Alleged Trades

In the event that an external party recognizes to have done a CLS eligible trade against you and you do not yet recognize it, or in the event there is a mismatch between what you have submitted as confirmation and what your counterpart has submitted, you can be warned via the CLS Trade Info Report with a Trade Notification sent by CLS.

From the Calypso Navigator, navigate to System > Domain Values.



In domain “clsParameters”, add value “allegedBook.<legal entity short name of the Control branch or submission PO>”.

Where for the comments of this domain name you will need to input the name of the default book you would want the alleged trades to go into.

CLS Exception Event Types	Description	Example
EX_CLS_ACK_FAILURE	Message sent to CLS was not successfully acked	Ex: SWIFT answer equals Failed Storage
EX_CLS_INFORMATION	CLS General Information	Ex1: Payments generated for CLS Pay-In Schedule: 08/13/2014 Ex2: CLS Message successfully acknowledged Ex3: CLS Message: Suspended for Settlement
EX_CLS_MESSAGE_FAILURE	CLS Message content validation failed due to business errors.	Ex: CLS Message Failure Reason: Quoted settlement session is invalid
EX_CLS_PAYMENT_FAILURE	Problem while processing pay-in schedule or pay-in call	Ex: Official 08/06/2014 Could not save Pay-In Schedule. Pay manually. Could not save schedule; (SQL Exception)
EX_CLS_PAY_IN_CALL	Pay-in call has been received/processed.	Ex: CLS Pay-In Call received, 2 trades generated: CFST /06/10/2014
EX_CLS_RECONCILIATION	Reconciliation issue for a currency.	Ex: Currency USD does not reconcile. CLS -343000.0 calypso null
EX_CLS_SWIFT_ERROR	Message failed after reception from SWIFT due to errors	Ex: Message Ref: G123456790 / ErrorCode/ErrorText: ErrorText
EX_CLS_NOTIFICATION!	Notification message failed	Ex. Could not find TransferAgent Trade for CLS Pay-In Notification. USD 2000.0. [CLSPayIn 07/27/2020 value date: 07/27/2020 ccy: USD amount: 2000.0 orderingBankRef: 16135 centralBankRef: V201507020178901]

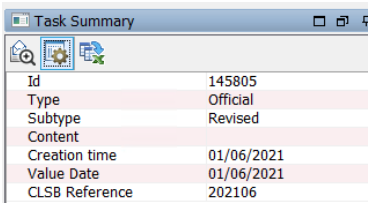
3.5.5 Exception Types

The below exception types can be monitored in the Task Station:

You can right-click an exception and bring up the Investigate menu to quickly access information related to the task.

The Task Summary panel also shows details about the task.

Example:



The table below details the Investigate menu options provided and Task Summary information:

CLS Exception Event Types	Investigate menu options	Task Summary
EX_CLS_ACK_FAILURE	External message: displays the xml message that was received	Not available as ACK failed.
EX_CLS_INFORMATION	XML message: displays message information Payment Trades: shows payment trades created Reco Result: displays the reconciliation results Schedule: displays the CLS pay-in schedule	Depends on the information given. Ex: If payments are generated based on pay-in schedule then the CLSPayInSchedule report is seen. Ex: If created because of a CLS Message then the CLS message report is given
EX_CLS_MESSAGE_FAILURE	XML message: displays message information CLSMessage Report: opens the CLS message report Window	Displays the CLS Message report
EX_CLS_PAYMENT_FAILURE	External message: displays the xml message that was received	Not available as pay in schedule is not created/saved
EX_CLS_PAY_IN_CALL	XML message: displays message information Reco Result: displays the reconciliation results Schedule: displays the CLS pay-in schedule CLS Pay-In Report: opens the CLSPayInSchedule Report window	Displays the CLSPayInSchedule report columns
EX_CLS_RECONCILIATION	XML message: displays message information Schedule displays the CLS pay-in schedule	Displays the CLSPayInSchedule report columns
EX_CLS_SWIFT_ERROR	Not available. Standard Bo message is seen in the task summary	Uses the standard BO message report
EX_CLS_NOTIFICATION	CLSAccountNotification: displays a view of the failed notification. XML message: displays the notification message	Not available as notification failed

3.5.6 Access Permissions

The following access permission functions are added to the system.

- CLSMT298Generation – To generate MT298 messages from the CLSPayInSchedule Report
- CLSPayTradeGeneration – To generate TransferAgent trades from the CLSPayInSchedule Report
- CLSRunReco – To run the reconciliation

The environment properties MAX_TRADES_PER_USER and MAX_MESSAGES_PER_USER have been activated for the CLS Trade Info report and the CLS Message report in order to limit the maximum number of trades to load.

You can also define the user attributes “Max.CLSTradeInfo” and “Max.CLSMessage” to define the limits per user. The limits per user have priority over the environment properties.

I/O Swaps

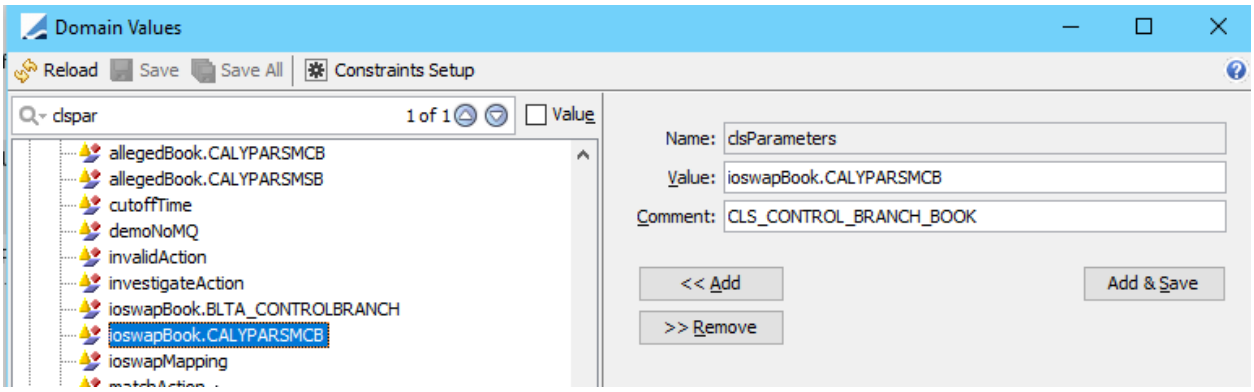
I/O Swaps are generated within CLS and made available to members via fax/website. To load the TOF files (which contain the full information of the I/O Swap details), parts of the Reuters DTS/TOF interface are used. Refer to the Calypso Reuters TOF Integration Guide.

For two of the fields errors (length exceeded) are “normal”.

I/O Swaps are FX Swap trades where both legs are value today, but one leg is settled inside CLS, the other outside CLS. For any potential Counterparty in an IO-Swap, you must define a REUTERS Legal Entity Attribute for the Counterparty using its Reuters „Bank Dealing Code“ (used in TOF field 508) E.g. “CIYL” for Citibank London.

The Book in which the I/O Swaps are fed in, is determined using the logic described in the above document.

You need to define a domain value “ioswapBook.<legal entity short Code>” with the book name in which you would like the I/O Swaps to be fed-in as comment in the “clsParameters” domain.

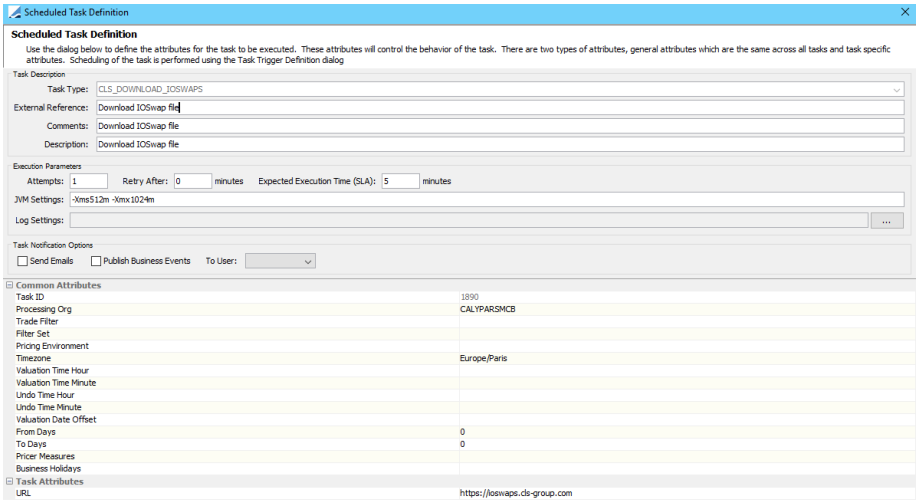


[NOTE: You can re-implement CustomTOFBookMapper if the current logic for the book default does not meet your requirements]

Once imported, I/O Swaps are tagged with a trade keyword CLS_IOSWAP = true. In the default setup this is a system keyword, so the user cannot edit it.

I/O Swap Download

Use the scheduled task CLS_DOWNLOAD_IOSWAPS



You need to set in the calypsouser.properties file the property:

- CLS_IOSWAP_USER: user to log into the ioswap website
- CLS_IOSWAP_PASSWORD: password
- CLS_DOWNLOAD_DIR: directory where you want to save this file
- SSL

While recent browsers can access the IO-Swap Website without problems, the JDKs provided by Sun are not currently functioning. This seems to be caused by an expired root certificate in the certificate chain sent by the IO-Swap web server. It expired on Jan 08, 2004.

If you are affected by this, you can create a jssecacerts file containing sufficient certification for the IO-Swap website and place it in the resources directory or update your JDK “cacerts” file.

On Windows, the “Security” tab on the Java Control Panel may help as well.

It may also be that CLS will fix this issue.

I/O Swap Integration

Use the scheduled task CLS_IMPORT_IOSWAPS described below.

Position Management

As explained, the trades created for the Pay-In Schedule and Pay-In calls use Transfer Agent products. These are not position-based, so they will not affect the position computed by the Liquidation engine and Position engine.

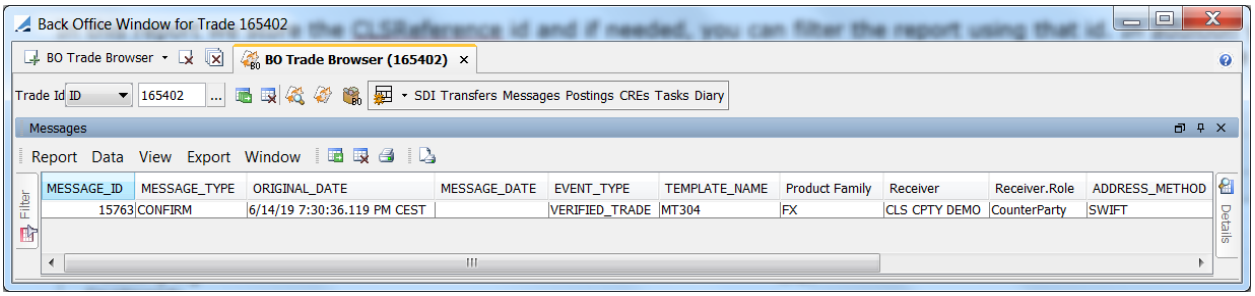
The Transfer Agent trades used for the Payments, transfer your position between CLS and your CLS nostro. Your CLS position in the Inventory Position report should be zero at the end of the settlement / pay-out process.

If you use Liquidation engine you can monitor your CLS position in real time.

Sample CLS Integration Lifecycle

Throughout the day, FX trades are input against a Counterparty that has a Legal Agreement with CLS.

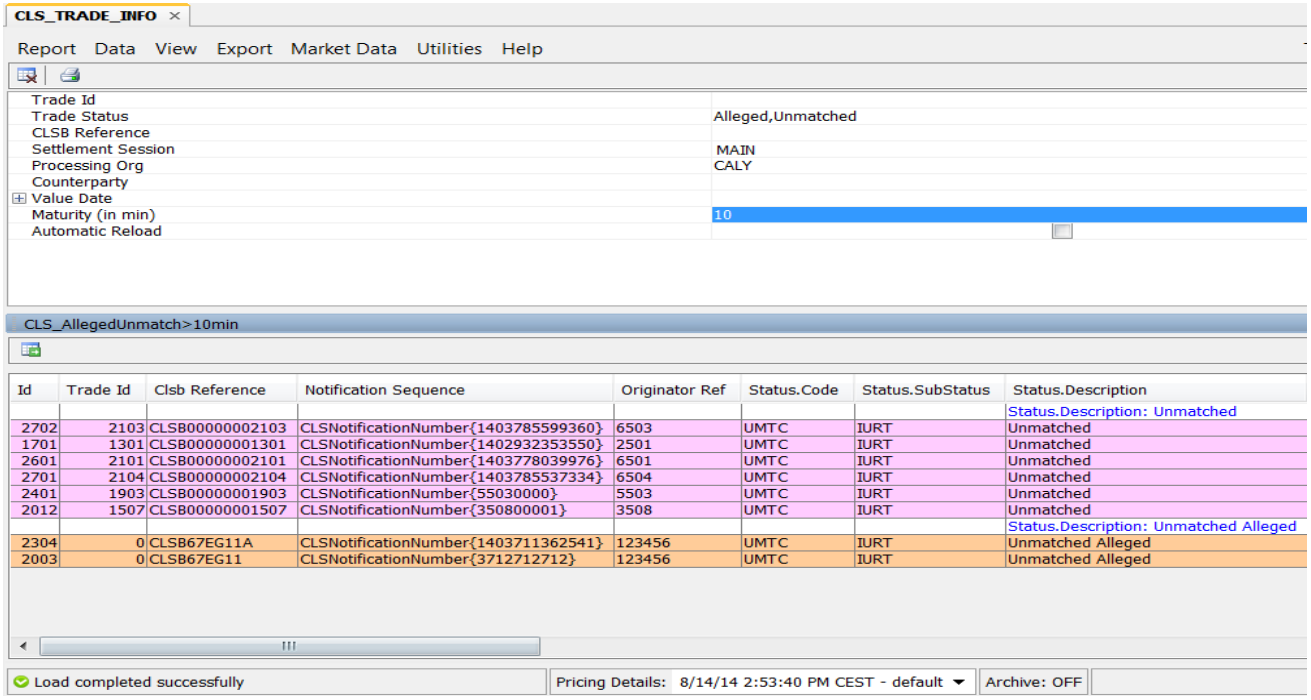
Dependent on whether the counterparty is Best Practice or not, an MT304 or an MT304 and an MT300 will be generated.



After the message has been SENT, confirmation messages can be imported from CLS.

You can monitor the status of confirmations in CLS (trade notifications), using the CLS Trade Info report (menu action `reporting.ReportWindow$CLSTradeInfo`).

In this report we store the CLSReference id and if needed, you can filter the report using that id. In addition, you can add the Calypso Msg_Attribute.CLS_REF which shows the CLS internal reference id. This attribute can be used in any report.



» Enter selection criteria as needed (they are described below), and click  to load the trades.

You can right-click a trade and drill-down to Audit, Trade and Transfer information from the Show menu.

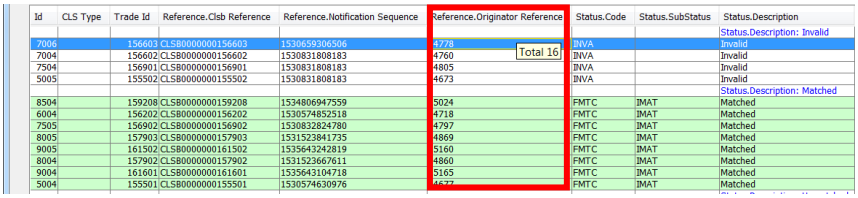
Selection Criteria

Fields	Description
Settlement Session	Values can be MAIN/AMER. It will check at the column General.Settlement.Session. Calypso only supports MAIN session.
Processing Org	Originator.Settlement.Member BIC Code defined in the Contact of the Legal Entity. Type a few letters of the BIC code and the filter will bring up the results for all the originators that start with those letters.
Counterparty	Counterparty.Settlement.Member BIC Code defined in the Contact of the Legal Entity. Type a few letters of the BIC code and the filter will bring up the results for all the counterparties that start with those letters.
Value Date	Trade Settle Date.
Start Date	The Start and End times can be set in the form HH:MM:SS AM or HH:MM:SS PM.
Start Tenor	Same functionality as for Start Date but set in TENOR -1D/-1Y etc.
End Date	The Start and End times can be set in the form HH:MM:SS AM or HH:MM:SS PM.
End Tenor	Same functionality as for End Date but set in TENOR -1D/-1Y etc.
Maturity (in min)	This parameter allows defining time limits of notifications to be filtered out of the report. Notification timestamp < Maturity (in min). Exampldiscrepancy,o exclude trades that are new, and have been unmatched or alleged for less than 10 minutes.
Automatic Reload	If the flag is check; the user will need to define the frequency to reload the report in seconds.
Timer (in sec)	Frequency in seconds to be defined. Example: 30.

CLS Trade Info Results

Should there be an incoming message with tag 20 customized, we would still find the relevant message id within calypso and match as shown below. As you will see, the originator reference of Message id 2461 has been customized but calypso is able to find and match it.

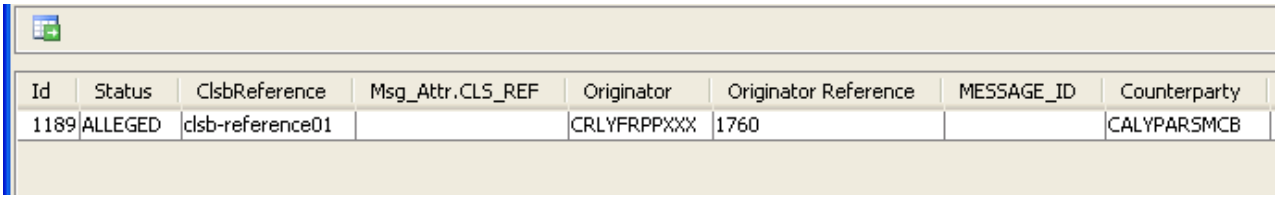
Id	Status	ClsbReference	Msg_Attr.CLS_REF	Originator	Originator Reference
1187	Matched	CLS00000000007702	CLS00000000007702	CALYPARSMCB	abcde000002461
1188	Matched	CLS00000000007701	CLS00000000007701	CALYPARSMCB	2464



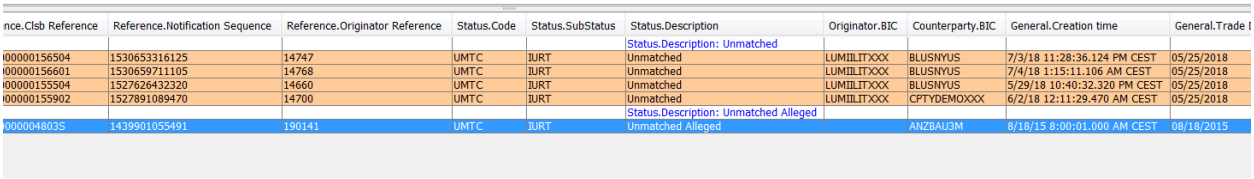
Id	CLS Type	Trade Id	Reference.Clsb Reference	Reference.Notification Sequence	Reference.Originator Reference	Status.Code	Status.SubStatus	Status.Description
								Status.Description: Invalid
7006		156603	CLSB0000000156603	1530659306506	4778	INVA		Invalid
7004		156602	CLSB0000000156602	1530831808183	4760	INVA		Invalid
7504		156901	CLSB0000000156901	1530831808183	4805	INVA		Invalid
5005		155502	CLSB0000000155502	1530831808183	4673	INVA		Invalid
								Status.Description: Matched
8504		159208	CLSB0000000159208	1534806947559	5024	FMTC	IMAT	Matched
6004		156202	CLSB0000000156202	1530574852518	4718	FMTC	IMAT	Matched
7505		156902	CLSB0000000156902	1530832824780	4797	FMTC	IMAT	Matched
8005		157903	CLSB0000000157903	1531523841735	4869	FMTC	IMAT	Matched
9005		161502	CLSB0000000161502	1535643242819	5160	FMTC	IMAT	Matched
8004		157902	CLSB0000000157902	1531523667611	4860	FMTC	IMAT	Matched
9004		161601	CLSB0000000161601	1535643104718	5165	FMTC	IMAT	Matched
5004		155501	CLSB0000000155501	1530574630976	4672	FMTC	IMAT	Matched

Total 16

In the event that an external counterparty submits trade confirmation to CLS you will get an alleged message in the CLS Trade Info report.



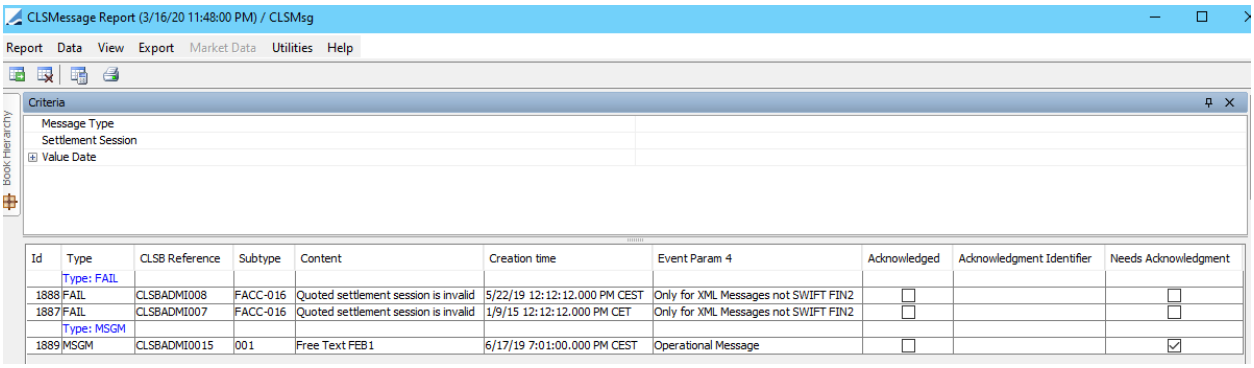
Id	Status	ClsbReference	Msg_Attr.CLS_REF	Originator	Originator Reference	MESSAGE_ID	Counterparty
1189	ALLEGED	clsb-reference01		CRLYFRPPXXX	1760		CALYPARSMCB



nce.Clsb Reference	Reference.Notification Sequence	Reference.Originator Reference	Status.Code	Status.SubStatus	Status.Description	Originator.BIC	Counterparty.BIC	General.Creation time	General.Trade I
					Status.Description: Unmatched				
00000156504	1530653316125	14747	UMTC	IURT	Unmatched	LUMILITXXX	BLUSNYUS	7/3/18 11:28:36.124 PM CEST	05/25/2018
00000156601	1530659711105	14768	UMTC	IURT	Unmatched	LUMILITXXX	BLUSNYUS	7/4/18 1:15:11.106 AM CEST	05/25/2018
00000155504	1527626432320	14660	UMTC	IURT	Unmatched	LUMILITXXX	BLUSNYUS	5/29/18 10:40:32.320 PM CEST	05/25/2018
00000155902	1527891089470	14700	UMTC	IURT	Unmatched	LUMILITXXX	CPTYDEMOXXX	6/2/18 12:11:29.470 AM CEST	05/25/2018
					Status.Description: Unmatched Alleged				
000004803S	1439901055491	190141	UMTC	IURT	Unmatched Alleged		ANZBAU3M	8/18/15 8:00:01.000 AM CEST	08/18/2015

6.1 CLS Message Report

To view any messages CLS may want to communicate to its clients for information purposes, you can use the CLS Message report (menu action `reporting.ReportWindow$CLSMessage`).

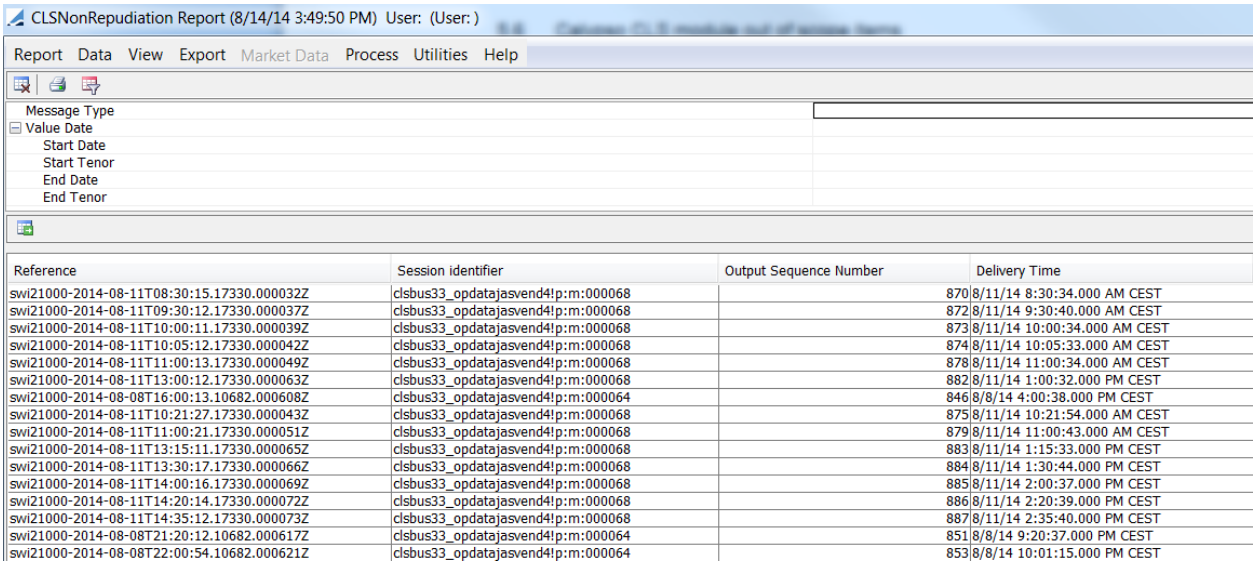


Id	Type	CLSB Reference	Subtype	Content	Creation time	Event Param 4	Acknowledged	Acknowledgment Identifier	Needs Acknowledgment
	Type: FAIL								
1888	FAIL	CLSBADMI008	FACC-016	Quoted settlement session is invalid	5/22/19 12:12:12.000 PM CEST	Only for XML Messages not SWIFT FIN2	☐		☐
1887	FAIL	CLSBADMI007	FACC-016	Quoted settlement session is invalid	1/9/15 12:12:12.000 PM CET	Only for XML Messages not SWIFT FIN2	☐		☐
	Type: MSGM								
1889	MSGM	CLSBADMI0015	001	Free Text FEB1	6/17/19 7:01:00.000 PM CEST	Operational Message	☐		☑

6.2 Non-Repudiation Report

To facilitate the access to the Non Repudiation messages sent by CLS, you can use the Non Repudiation report (menu action `reporting.ReportWindow$CLSNonRepudiation`).

You can check the attached XML message sent by CLS using Process > Show XML.



Reference	Session identifier	Output Sequence Number	Delivery Time
swi21000-2014-08-11T08:30:15.17330.000032Z	clsbus33_opdatajasvend4!p:m:000068	870	8/11/14 8:30:34.000 AM CEST
swi21000-2014-08-11T09:30:12.17330.000037Z	clsbus33_opdatajasvend4!p:m:000068	872	8/11/14 9:30:40.000 AM CEST
swi21000-2014-08-11T10:00:11.17330.000039Z	clsbus33_opdatajasvend4!p:m:000068	873	8/11/14 10:00:34.000 AM CEST
swi21000-2014-08-11T10:05:12.17330.000042Z	clsbus33_opdatajasvend4!p:m:000068	874	8/11/14 10:05:33.000 AM CEST
swi21000-2014-08-11T11:00:13.17330.000049Z	clsbus33_opdatajasvend4!p:m:000068	878	8/11/14 11:00:34.000 AM CEST
swi21000-2014-08-11T13:00:12.17330.000063Z	clsbus33_opdatajasvend4!p:m:000068	882	8/11/14 1:00:32.000 PM CEST
swi21000-2014-08-08T16:00:13.10682.000608Z	clsbus33_opdatajasvend4!p:m:000064	846	8/8/14 4:00:38.000 PM CEST
swi21000-2014-08-11T10:21:27.17330.000043Z	clsbus33_opdatajasvend4!p:m:000068	875	8/11/14 10:21:54.000 AM CEST
swi21000-2014-08-11T11:00:21.17330.000051Z	clsbus33_opdatajasvend4!p:m:000068	879	8/11/14 11:00:43.000 AM CEST
swi21000-2014-08-11T13:15:11.17330.000065Z	clsbus33_opdatajasvend4!p:m:000068	883	8/11/14 1:15:33.000 PM CEST
swi21000-2014-08-11T13:30:17.17330.000066Z	clsbus33_opdatajasvend4!p:m:000068	884	8/11/14 1:30:44.000 PM CEST
swi21000-2014-08-11T14:00:16.17330.000069Z	clsbus33_opdatajasvend4!p:m:000068	885	8/11/14 2:00:37.000 PM CEST
swi21000-2014-08-11T14:20:14.17330.000072Z	clsbus33_opdatajasvend4!p:m:000068	886	8/11/14 2:20:39.000 PM CEST
swi21000-2014-08-11T14:35:12.17330.000073Z	clsbus33_opdatajasvend4!p:m:000068	887	8/11/14 2:35:40.000 PM CEST
swi21000-2014-08-08T21:20:12.10682.000617Z	clsbus33_opdatajasvend4!p:m:000064	851	8/8/14 9:20:37.000 PM CEST
swi21000-2014-08-08T22:00:54.10682.000621Z	clsbus33_opdatajasvend4!p:m:000064	853	8/8/14 10:01:15.000 PM CEST

6.3 CLS Pay-In Schedule Report

Once CLS updates all trades to Settled Matured, Calypso extracts IPIS (Initial Pay In Schedule flagged in Calypso as type 'official') and reconciles between what CLS publishes to be the due amount per Ccy and what Calypso has for its transfers and if desired at this stage generates MT298s to Nostro agents.

When the RPIS is extracted (Revised Pay In Schedule flagged in Calypso as type 'official'), Calypso not only reconciles between amounts in Calypso vs. CLS but also generates Transfer Agent trades to cover payments to CLS and generates MT298s to Nostro agents if desired.

[NOTE: Because RPIS report will load in as type 'Official' as well, we have created a column in the CLS Pay-In Schedule Report which flags which of the 'official' schedules is the 'revised' one. Calypso in the upload process differentiates between the two (initial/revised) schedules from CLS based on timestamp of the schedule. It expects the 'revised' schedule to have a timestamp after 06:30am]

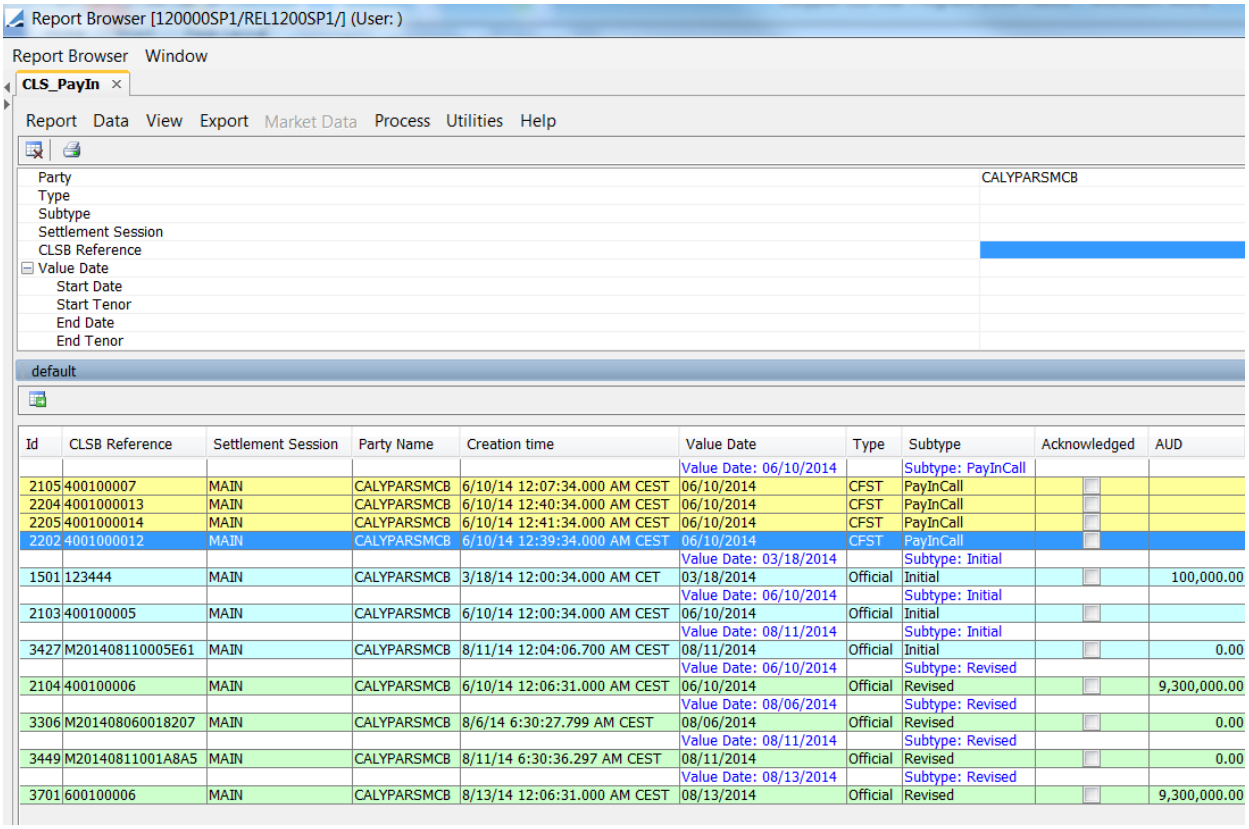
The Pay-In Schedules you request from CLS (which can be done at any time) will load in as type 'requested.' The CLS Pay-In Schedule report will import the three different types of reports (InitalPayIn, RevisedPayIn and on demand Requested).

Via the CLS Pay-In Schedule Report (for each of the schedules that have loaded) by going to Process menu, or by highlighting and right clicking on a schedule to display popup menu, you can:

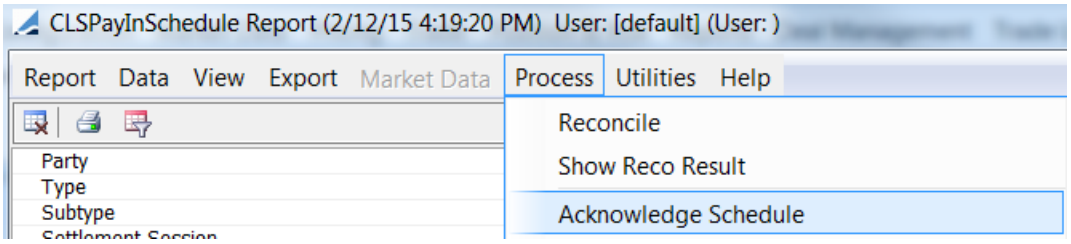
- Reconcile – You need the CLSRunReco access permission function
- View Reconciliation Results (reconciliation per Ccy)
- View PayIn Schedule (payment time expected per Ccy)
- Edit PayIn Schedule
- View Payment Trades (TransferAgent Payment trades)
- Generate Payment Trades

- View MT298 Messages
- Generate MT298 Messages

To view and identify CLS Pay-In amounts from Pay-Schedules, you can use the CLS Pay-In Schedule report (menu action `reporting.ReportWindow$CLSPayInSchedule`).

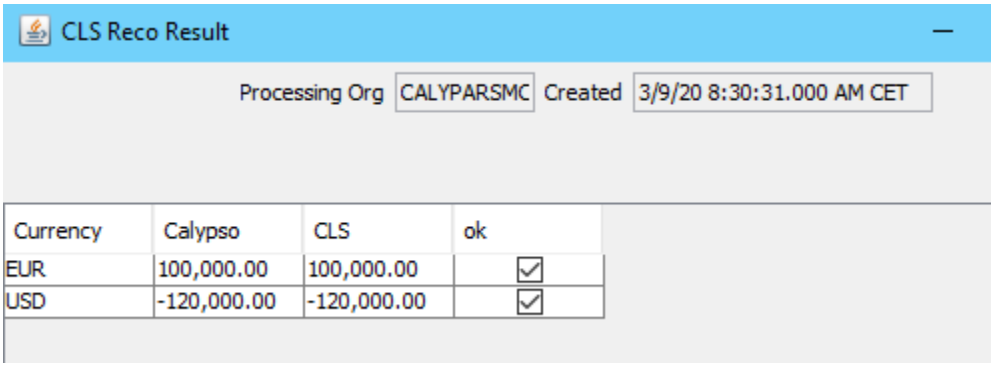


You can manually acknowledge the schedule using Process > Acknowledge Schedule. An Ack message will be sent to CLS.



6.3.1 Reconciliation

In Practice, once reconciliation has been launched, choose Process > Show Reco Result to view the reconciliation results between the total amount per Ccy due into CLS from the Calypso perspective and the total amount due per Ccy from CLS point of view.



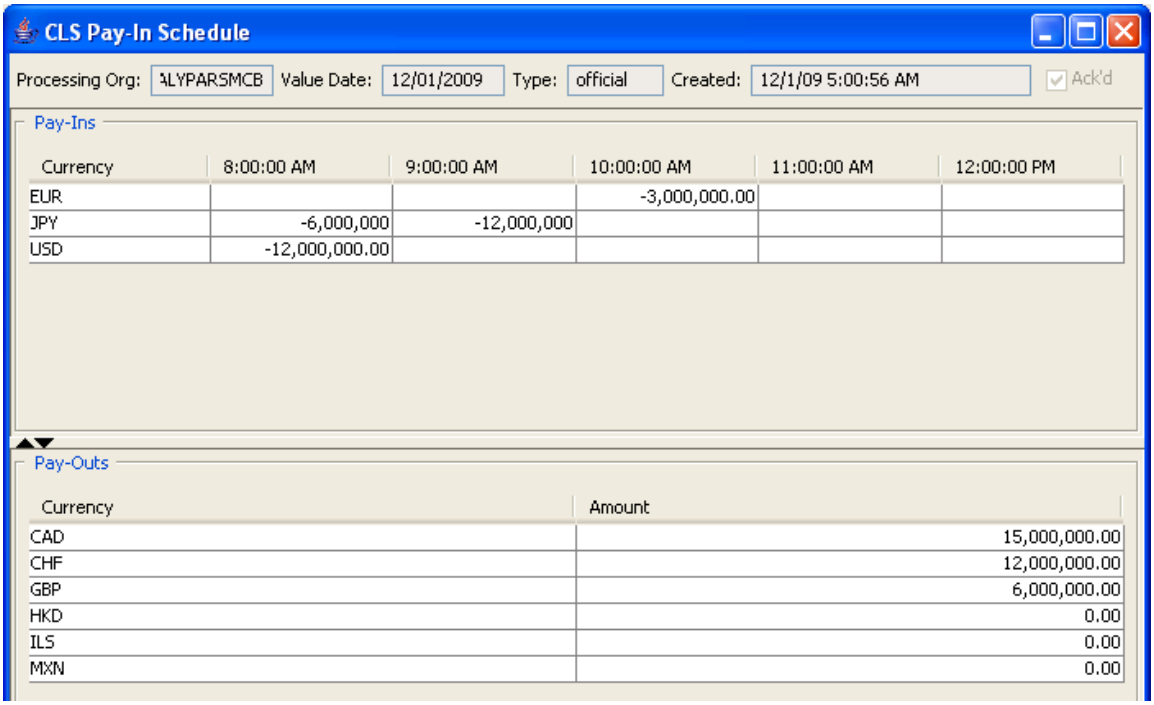
The "Show Reco Result" window shows you if the CLS transfer amounts in Calypso and the CLS due amounts per Ccy tie out. A checkbox will be ticked (dependent on the amount you have set in your clsRecoTolerance domain value) if there is a tie out per Ccy. If there isn't reconciliation for a Ccy type (as per the JPY example above) then we generate an EX_CLS_RECONCILIATION exception in the Task Station. At this point, you can double check the amount of the breaking currency and if need be correct, it in the system.

It should be noted that as soon as a PayInSchedule has been loaded in, the system will generate the relevant MT298 messages (even in the case where for a given currency the due amount is '0') as long as the Legal Entity attribute CLSMT298Trigger has been correctly defined for each relevant Nostro Agent.

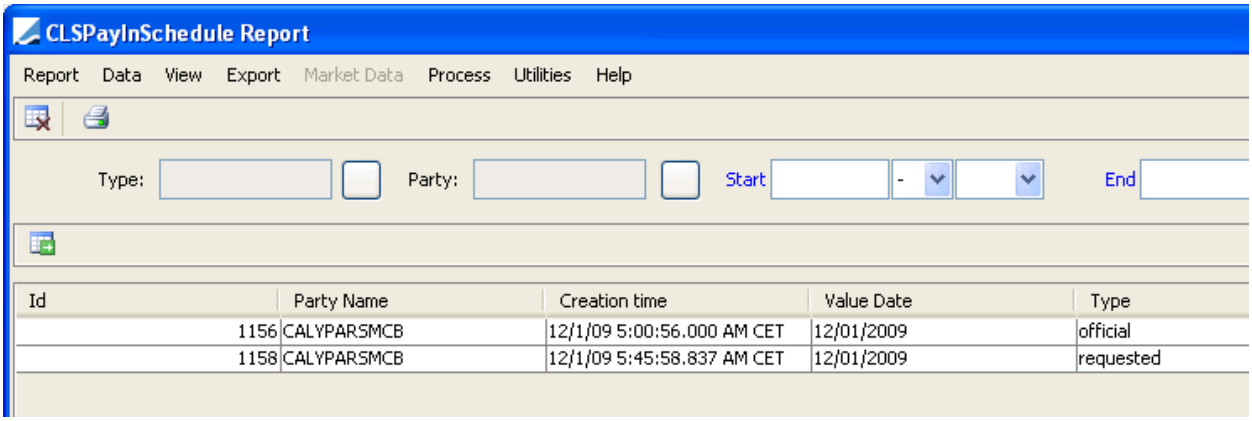
You can define a delay in the reconciliation process to allow the inclusion of late trade notifications. See the CLS parameter "RecoDelay" for details.

6.3.2 Show Schedule

If you would like to see the time the amounts are due per Ccy, right-click and choose Process > Show Schedule.

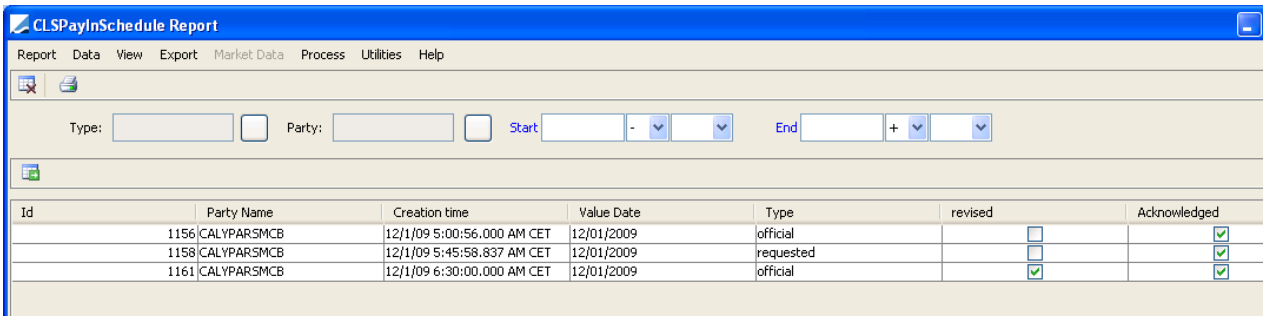


Should you need to get an update of the PayIn Schedule at any time between the InitalPayInSchedule and the Revised PayIn Schedule you can always request a Schedule which will come into the CLSPayInSchedule report as type 'requested.'



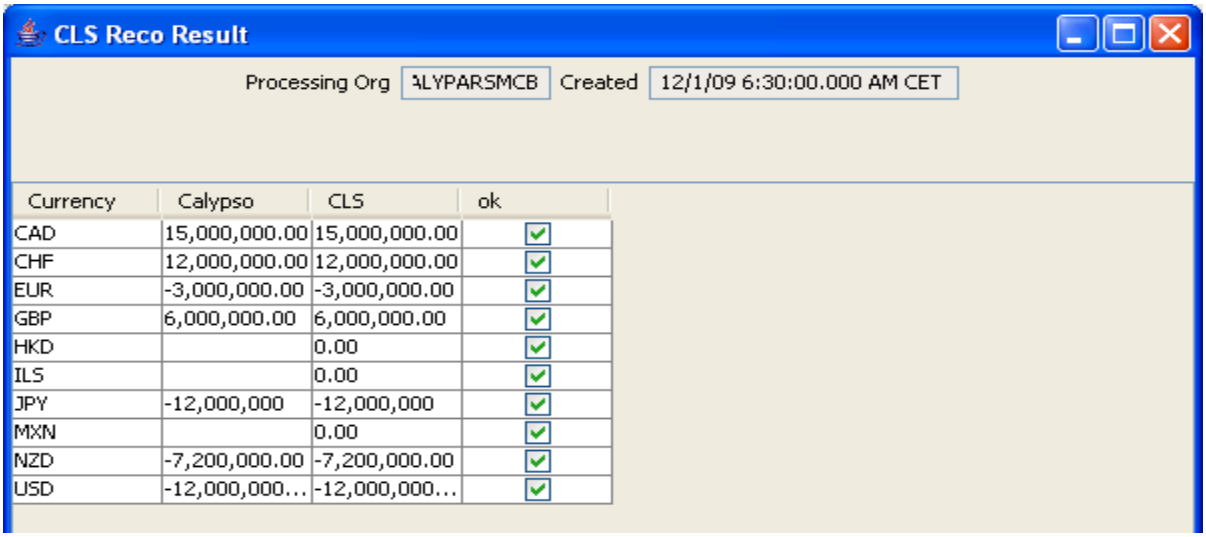
Id	Party Name	Creation time	Value Date	Type
1156	CALYPARSMCB	12/1/09 5:00:56.000 AM CET	12/01/2009	official
1158	CALYPARSMCB	12/1/09 5:45:58.837 AM CET	12/01/2009	requested

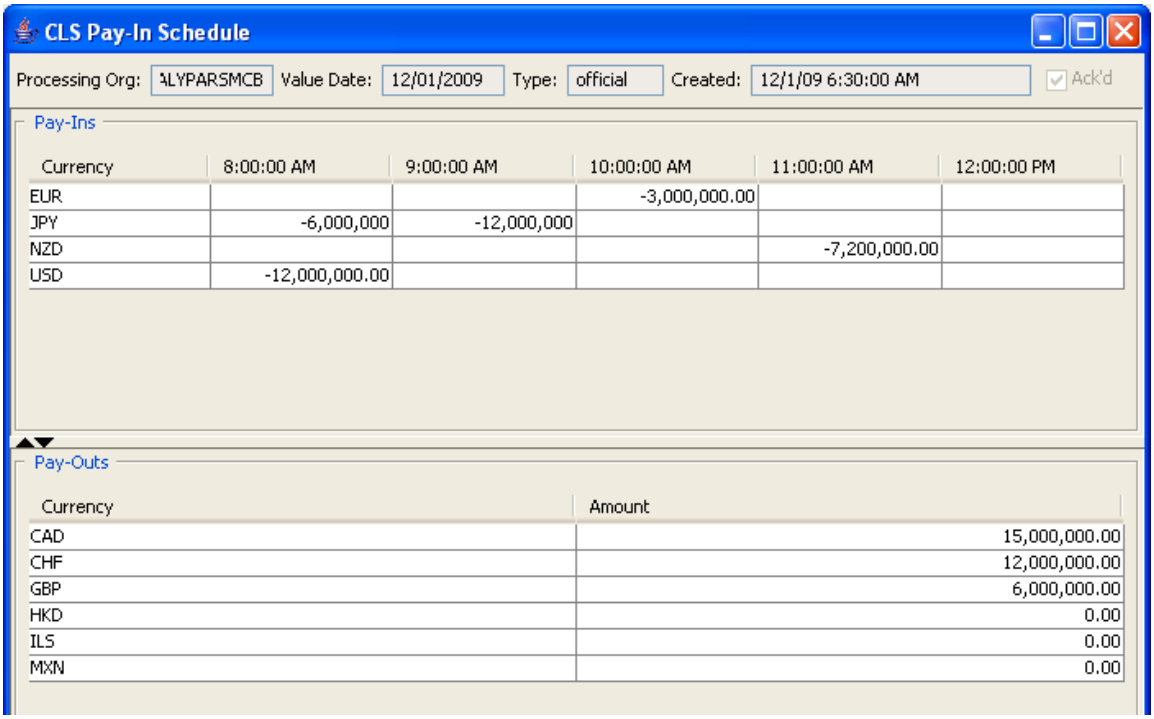
At 06:30, when CLS produces Revised Pay-In Schedules Calypso (in the same manner as the Initial Pay-In Schedule) provides a way to extract and view the schedule in the CLSPayInSchedule report. This schedule will be defined as type 'official' with 'revised' check flagged.



Id	Party Name	Creation time	Value Date	Type	revised	Acknowledged
1156	CALYPARSMCB	12/1/09 5:00:56.000 AM CET	12/01/2009	official	☐	☑
1158	CALYPARSMCB	12/1/09 5:45:58.837 AM CET	12/01/2009	requested	☐	☑
1161	CALYPARSMCB	12/1/09 6:30:00.000 AM CET	12/01/2009	official	☑	☑

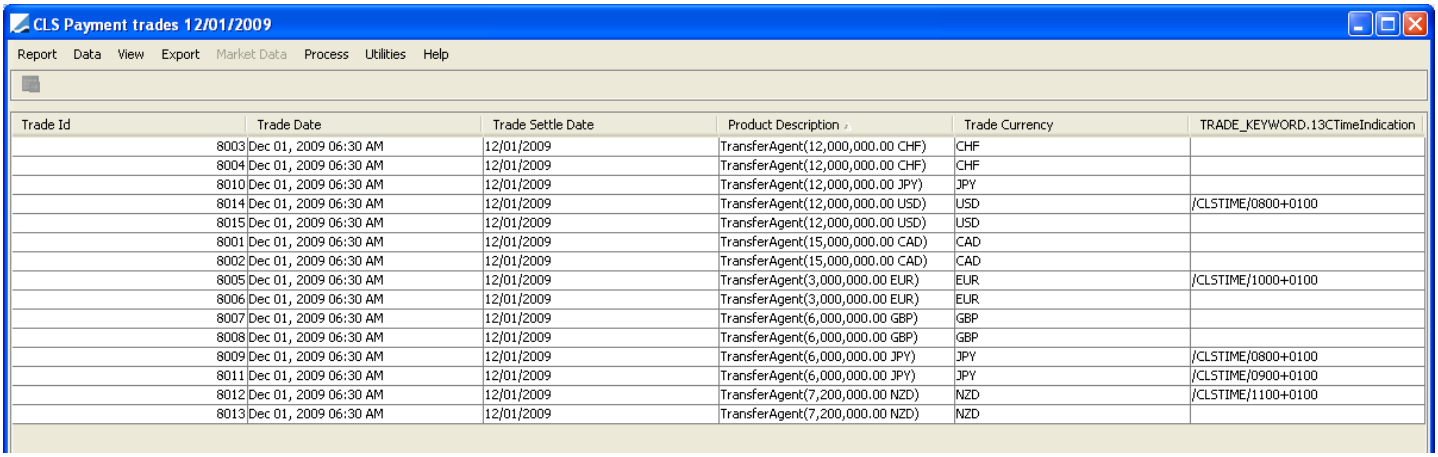
Once the schedule is downloaded into Calypso, in the same way you viewed the related reports when the Initial Pay-In was loaded (by going to Process > Show Recon Result), you can now view the reconciliation per currency and the time schedule the amounts are due.





6.3.3 Show Payment Trades

You can right-click a revised Pay-In Schedule and choose Process > Show Payment Trades to view the automatically generated Pay-In and Pay-Out Transfer Agent trades.



Trade Id	Trade Date	Trade Settle Date	Product Description	Trade Currency	TRADE_KEYWORD.13CTimeIndication
8003	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(12,000,000.00 CHF)	CHF	
8004	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(12,000,000.00 CHF)	CHF	
8010	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(12,000,000.00 JPY)	JPY	
8014	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(12,000,000.00 USD)	USD	/CLSTIME/0800+0100
8015	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(12,000,000.00 USD)	USD	
8001	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(15,000,000.00 CAD)	CAD	
8002	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(15,000,000.00 CAD)	CAD	
8005	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(3,000,000.00 EUR)	EUR	/CLSTIME/1000+0100
8006	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(3,000,000.00 EUR)	EUR	
8007	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(6,000,000.00 GBP)	GBP	
8008	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(6,000,000.00 GBP)	GBP	
8009	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(6,000,000.00 JPY)	JPY	/CLSTIME/0800+0100
8011	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(6,000,000.00 JPY)	JPY	/CLSTIME/0900+0100
8012	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(7,200,000.00 NZD)	NZD	/CLSTIME/1100+0100
8013	Dec 01, 2009 06:30 AM	12/01/2009	TransferAgent(7,200,000.00 NZD)	NZD	

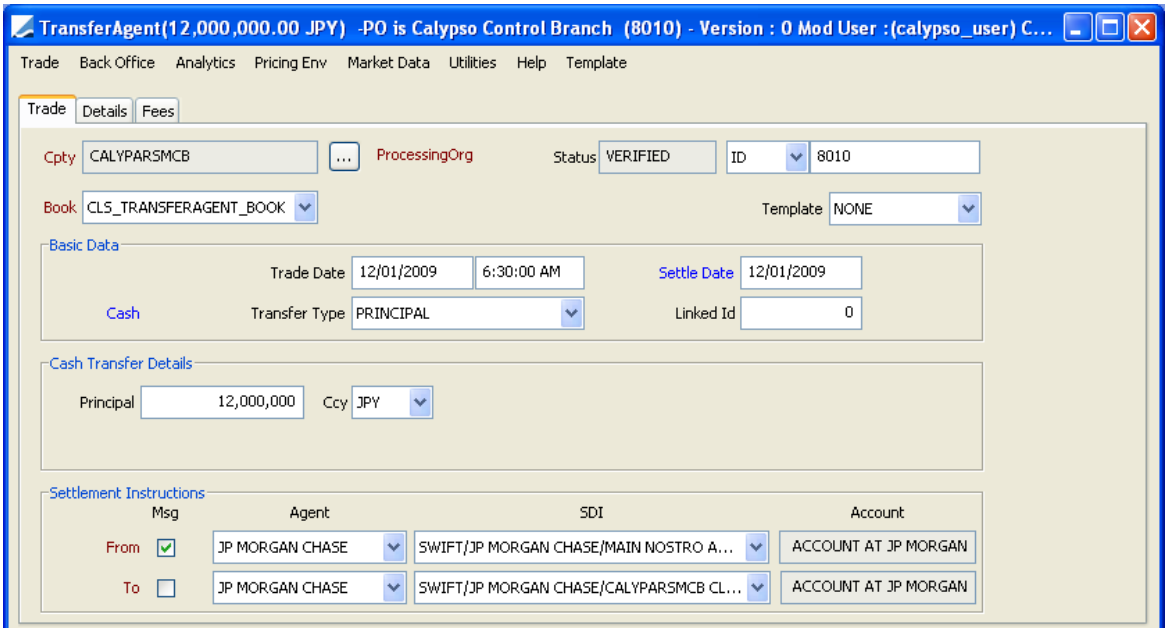
Double clicking on any line will open up the trades.

For Pay-In Transfer Agent trades, in the event you have sweep accounts set up calypso will generate a Sweep trade for the totality of your CLS due amount per Ccy which will serve to send payment messages from your Main Nostro to your CLS Nostro account.

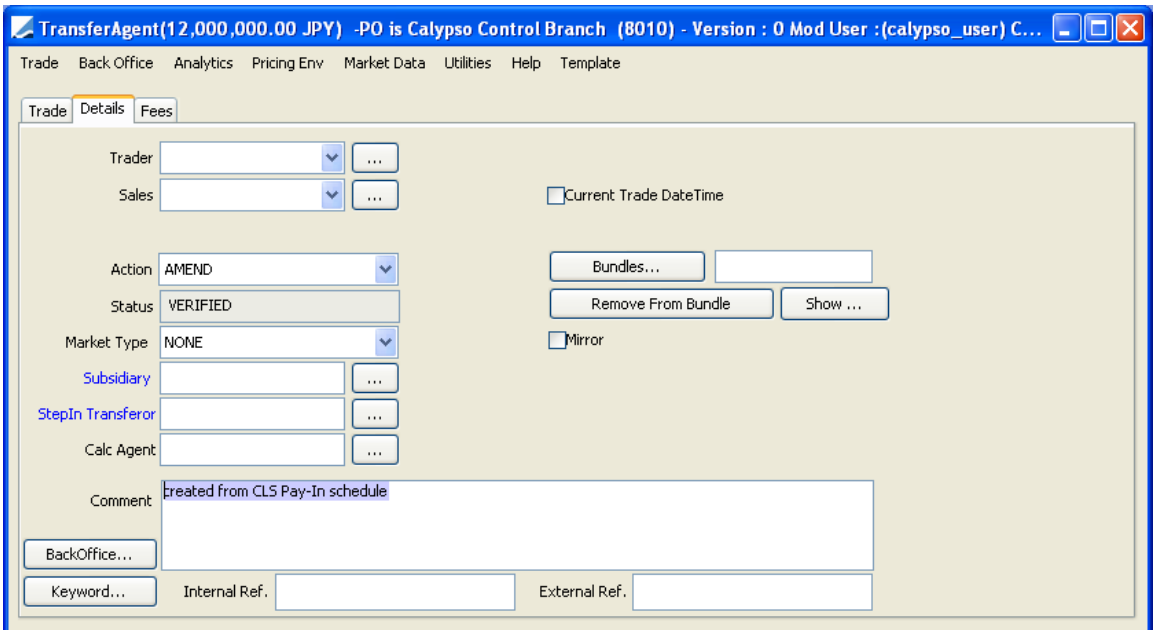
Thereafter Tranfer Agent trades (generating payment messages to Nostro agents advising flow from CLS Nostro account to account at CLS) will be generated based on quantity and due times at CLS. Should you not have sweep accounts, then Calypso will generate you only this portion of the Transfer Agent trade.

If we take the below JPY Pay-In as an example, where there is a sweep account involved, we see that the sweep Transfer Agent has been generated for the totality (12,000,000) of the JPY due amount. This trade generates the payment message from Main Nostro to CLS Nostro Account.

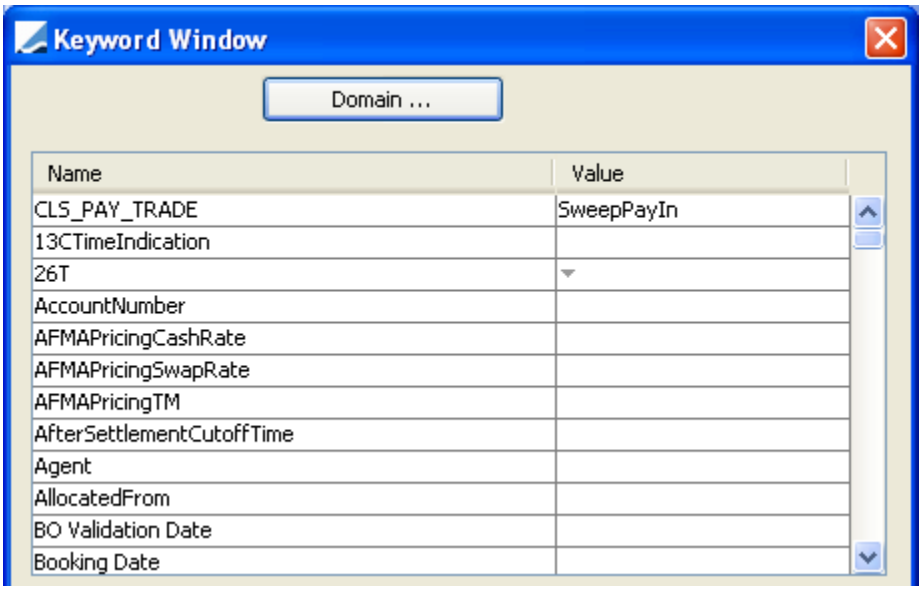
[NOTE: The book used is the default book defined in my domain "payInOutBook.<legal entity short name of control branch>"]



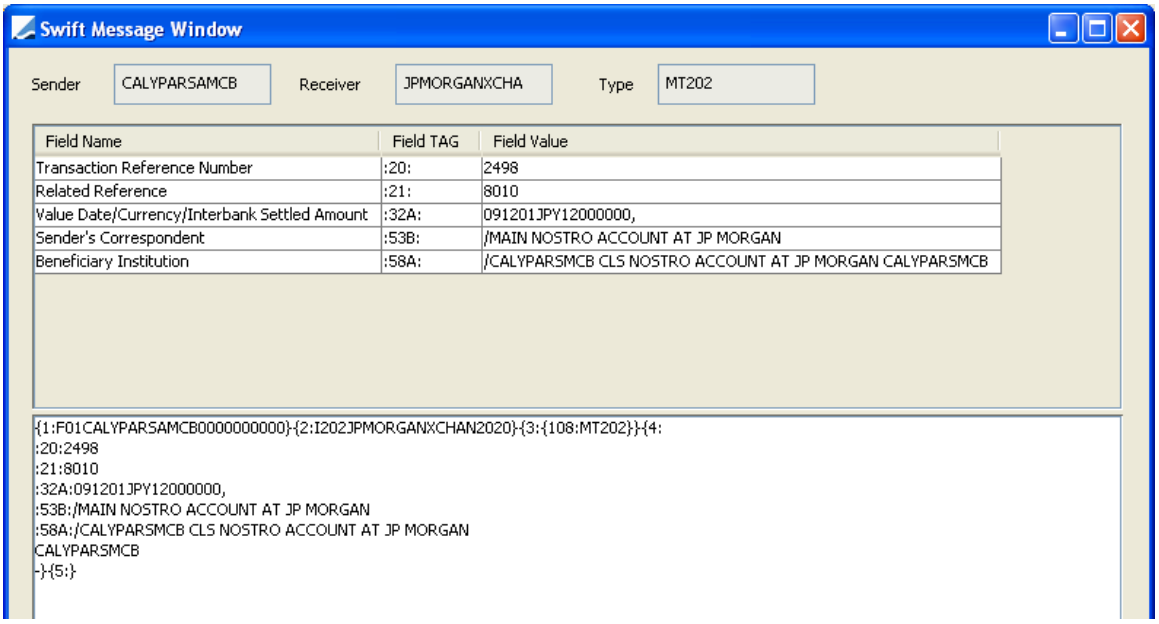
Each trade automatically generated after a Pay-In Schedule has been reconciled will have a comment in the details tab specifying that the trade was 'Created from CLS Pay-In Schedule.'



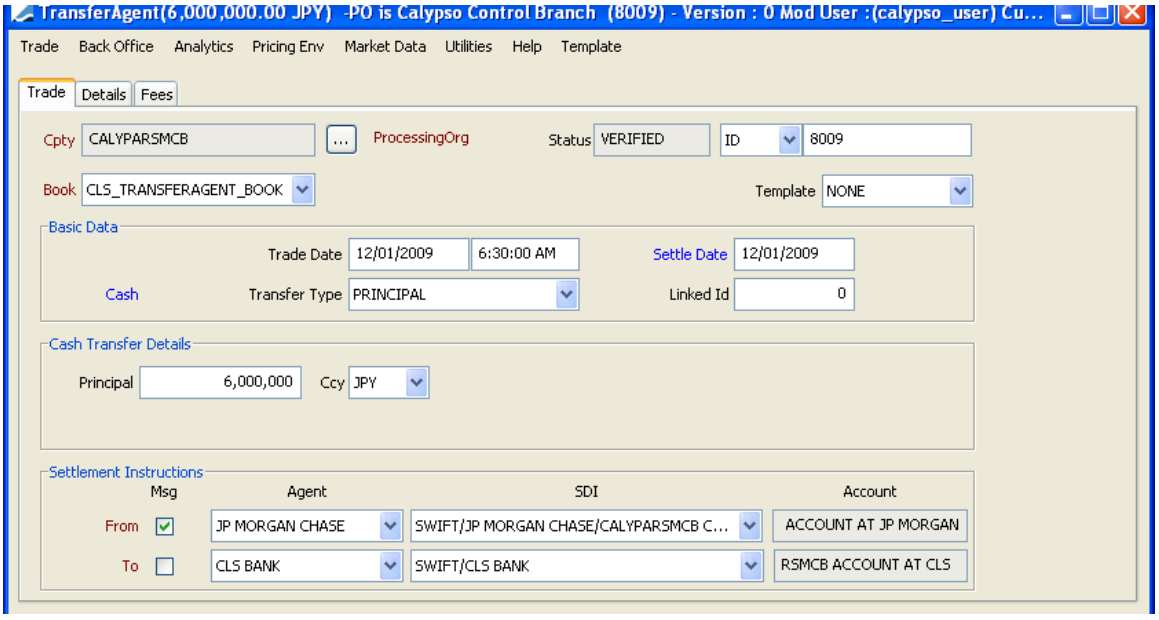
The trade will also be flagged as a 'SweepPayIn' on trade keyword CLS_PAY_TRADE. This keyword is a system keyword that cannot be edited by the user.



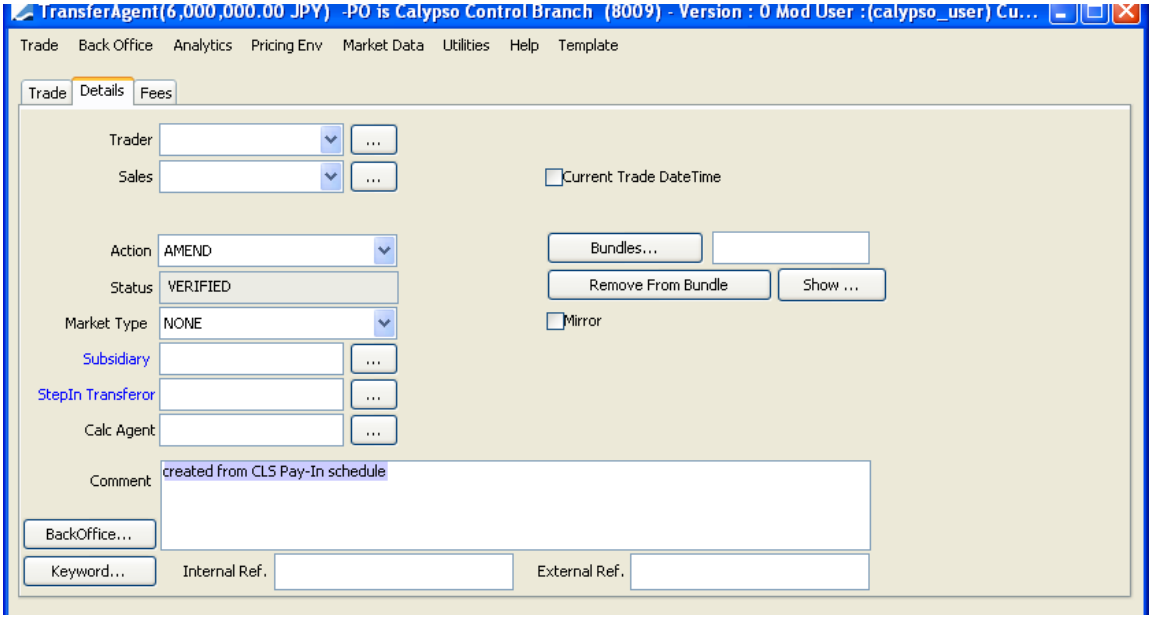
The related MT202 payment message will generate as follows:



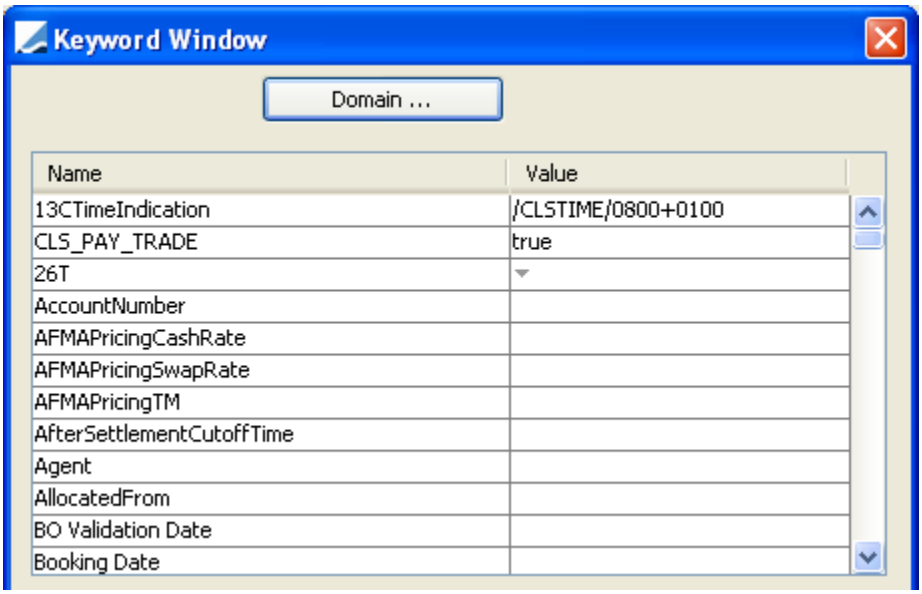
As my JPY due amounts at CLS are expected at two different times, calypso will generate the necessary Transfer Agent trades representing the specific due amounts at their specific times. Again, these will generate payment messages to your Nostro agent informing them of the payment details to move from CLS Nostro Account into account at CLS.



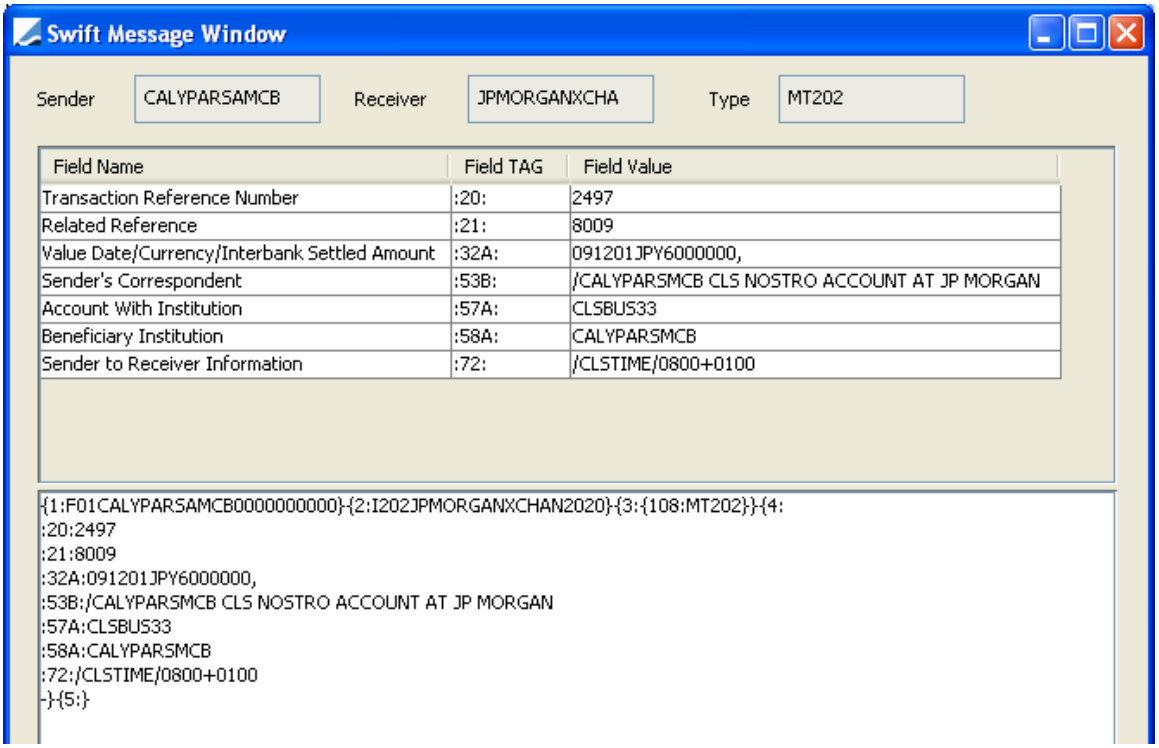
Each trade - automatically generated after a Pay-In Schedule has been reconciled - will have a comment in the details tab specifying that the trade was ‘Created from CLS Pay-In Schedule.’



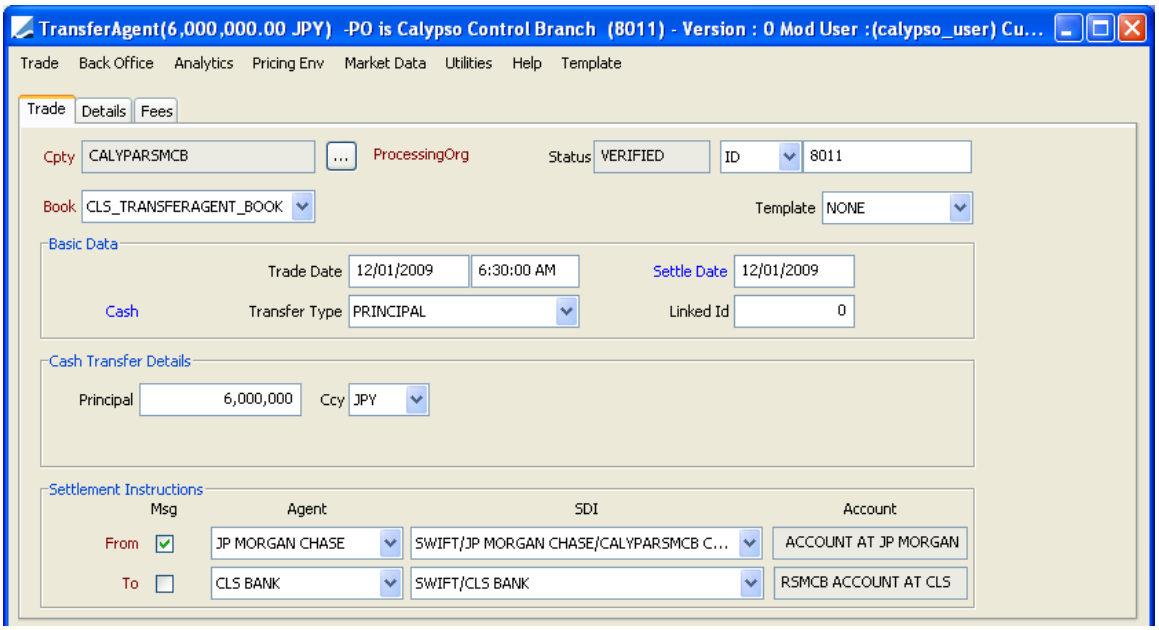
This trade will be flagged as 'true' on keyword CLS_PAY_TRADE and will also include the 13CTimeIndication (payment due time at CLS).

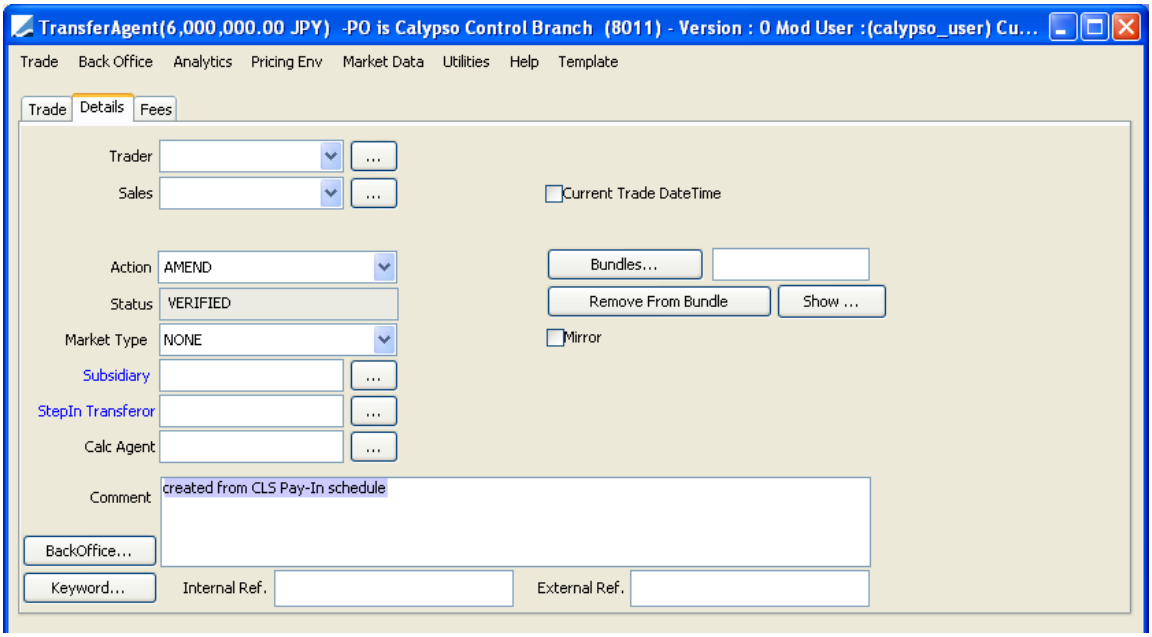


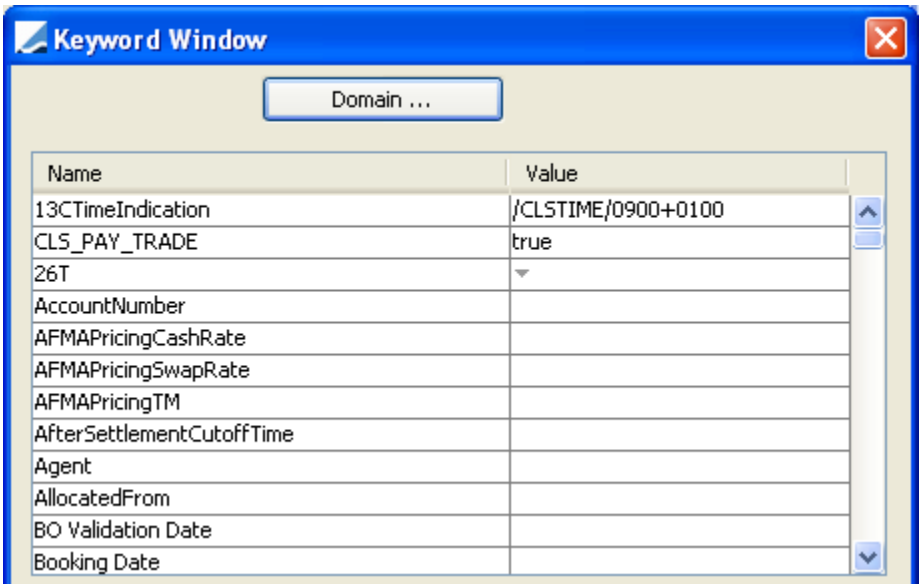
As payment is in 'verified' status, an MT202 is automatically generated with tag 72 indicating the due time into CLS.



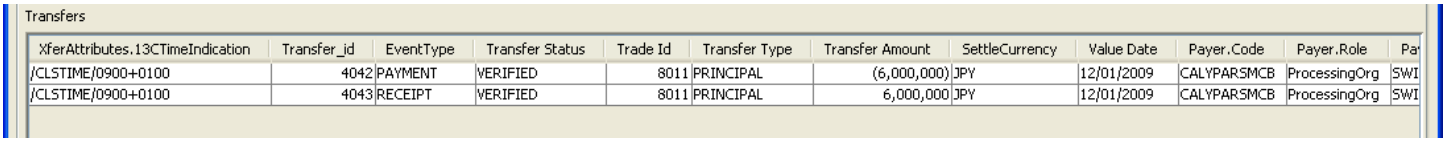
Second trade at the next JPY due time.

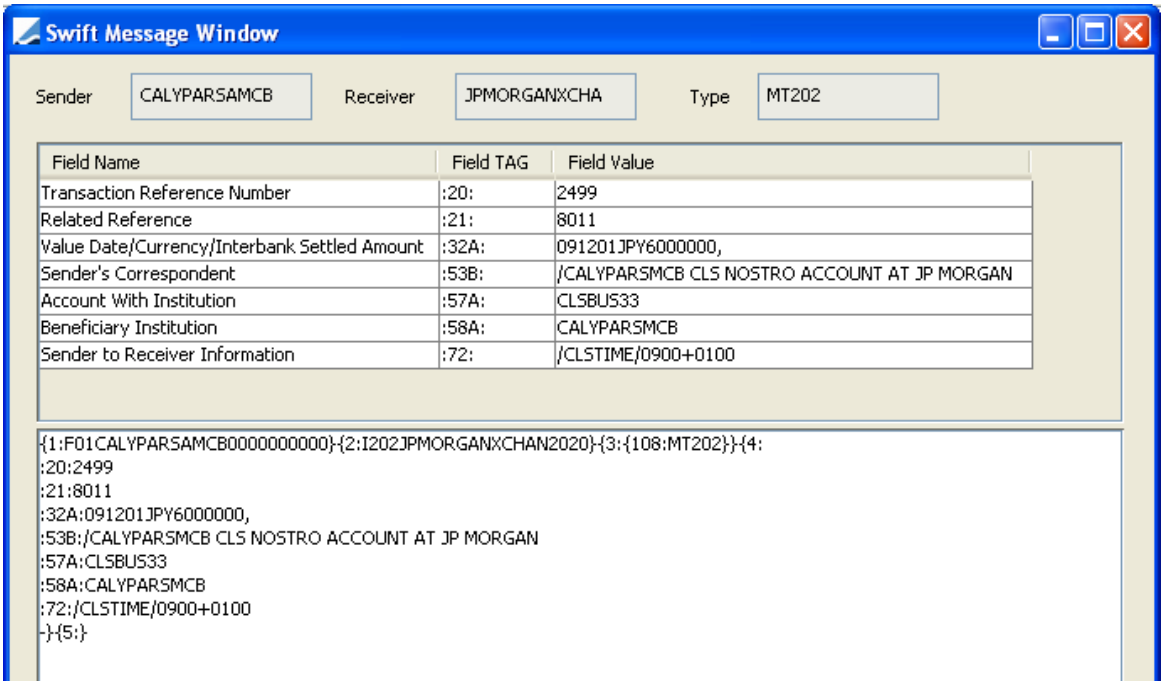




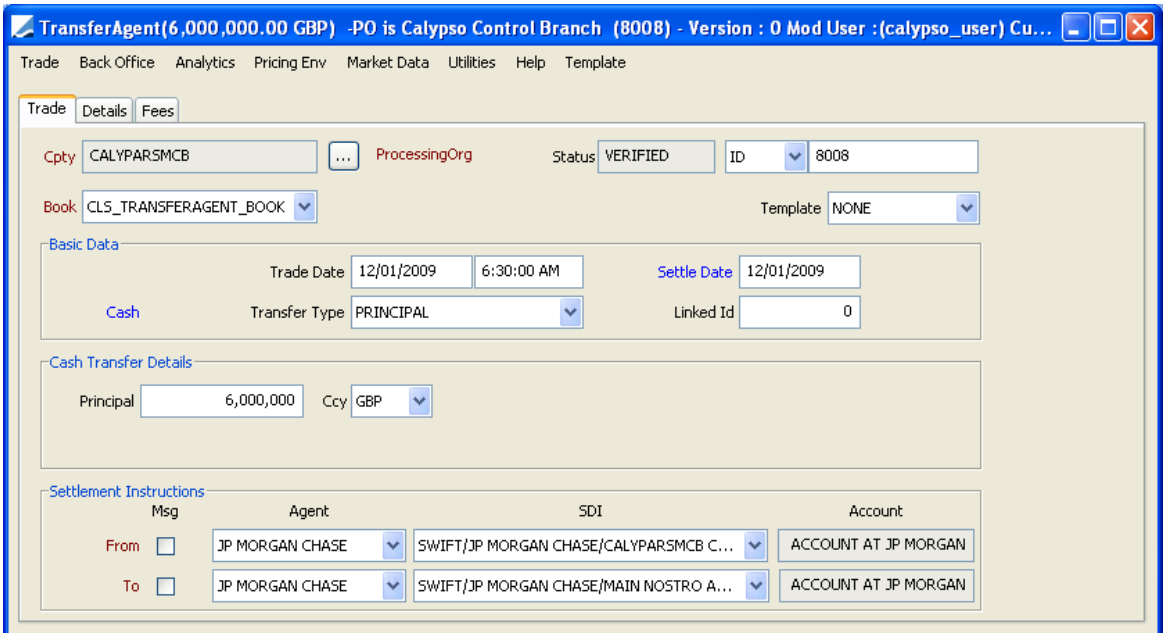


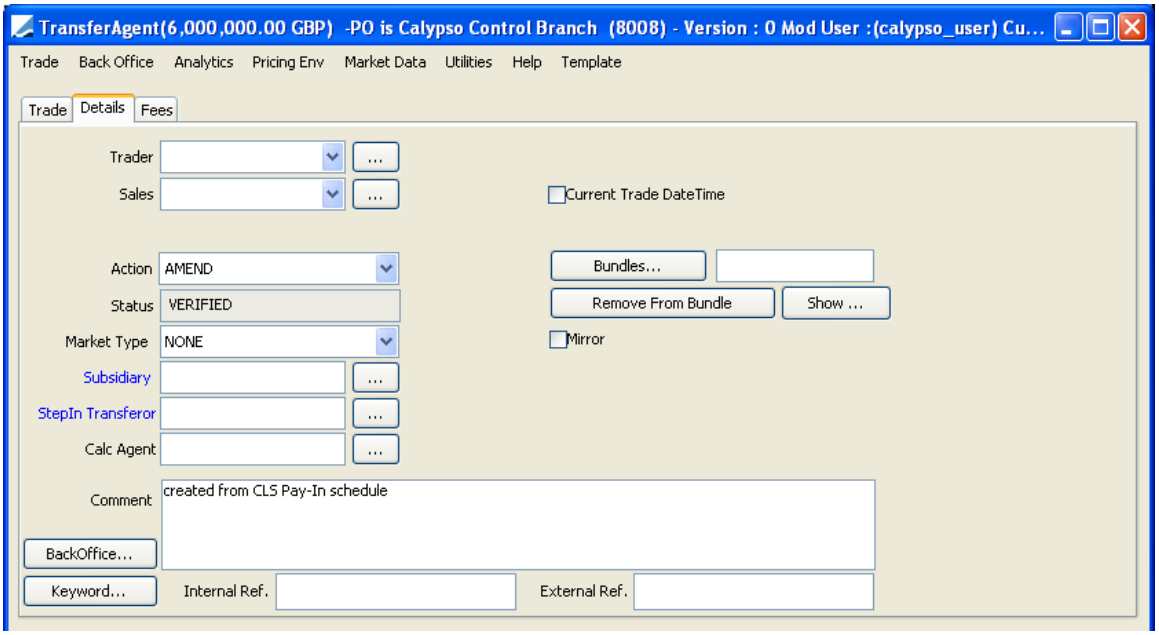
For these trades in the BO Browser, in the Transfer tab we can configure column to see the xferAttribute.13CTimeIndication.

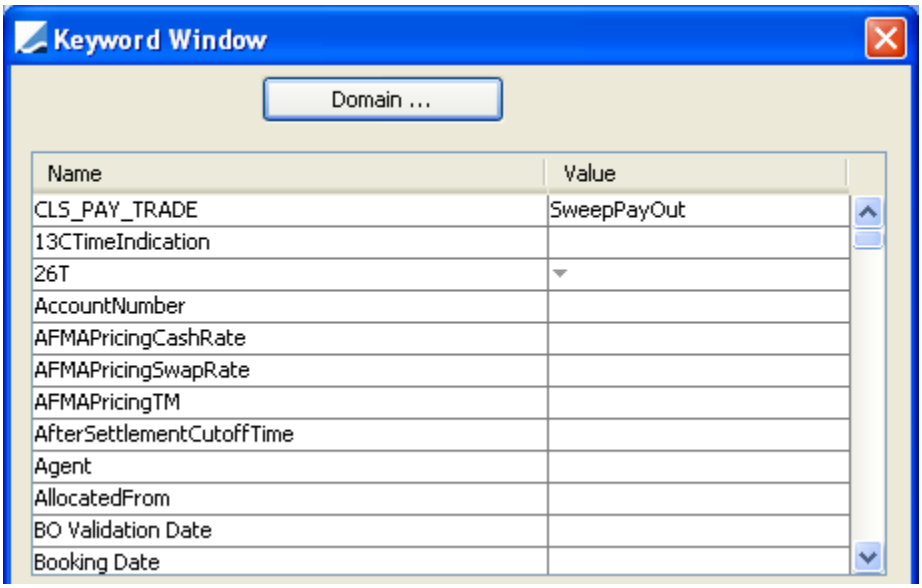


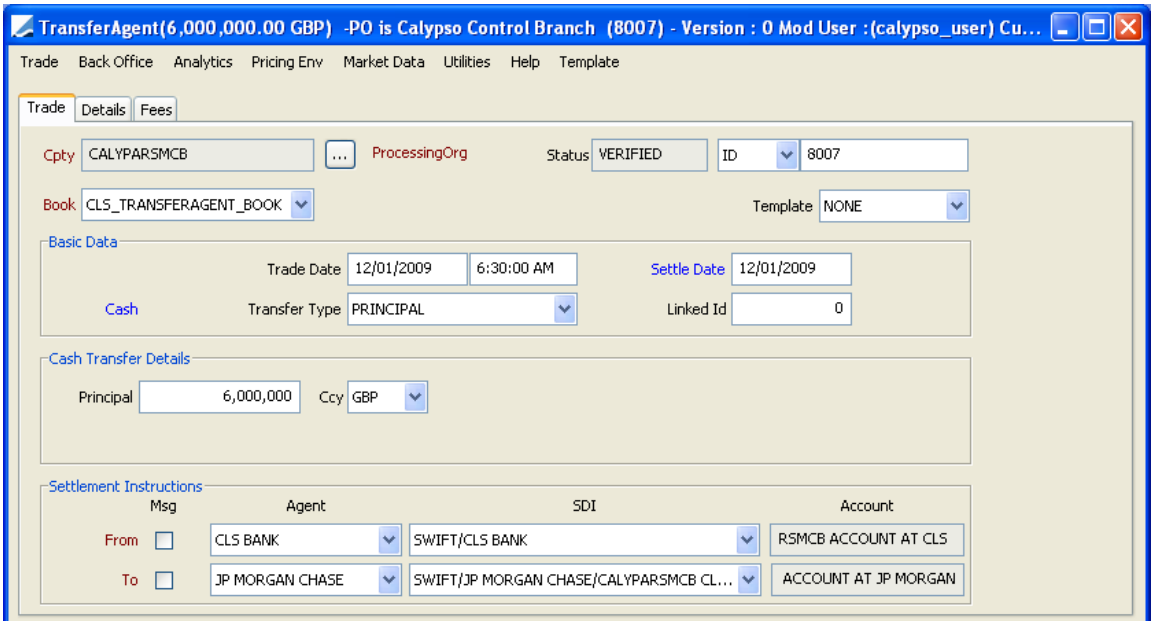


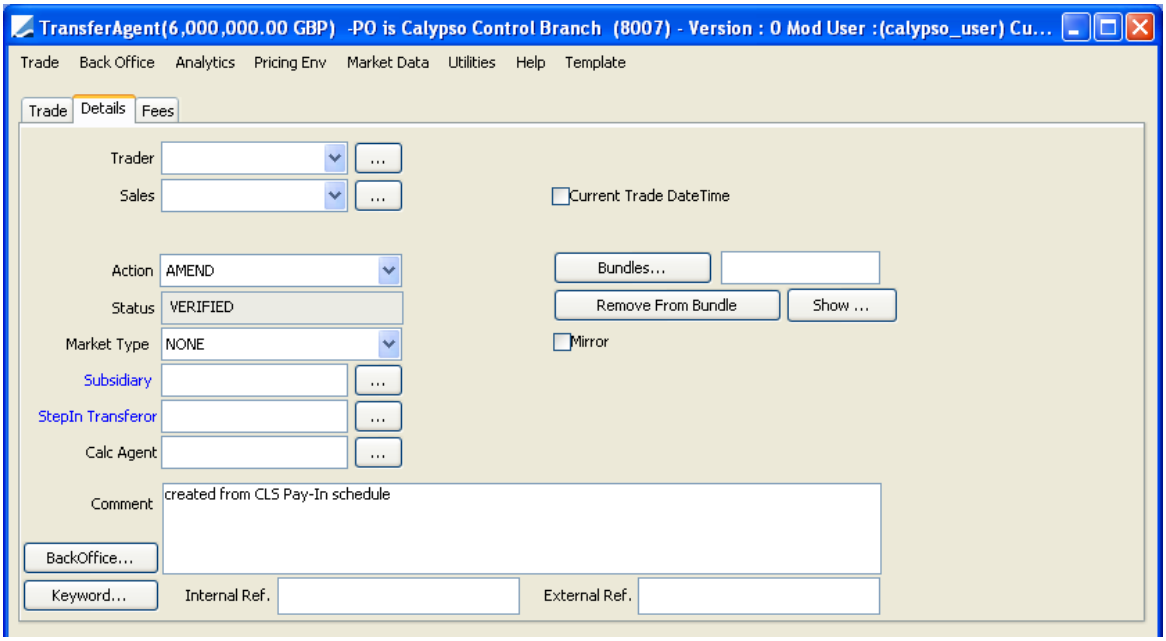
For the Pay-Out if we take my GBP as an example, we will see that Calypso has generated two trades. A Transfer Agent trade is created to represent the move from CLS to your CLS Nostro account, and another Transfer Agent trade is created to represent the move from your CLS Nostro account to your Main Nostro.

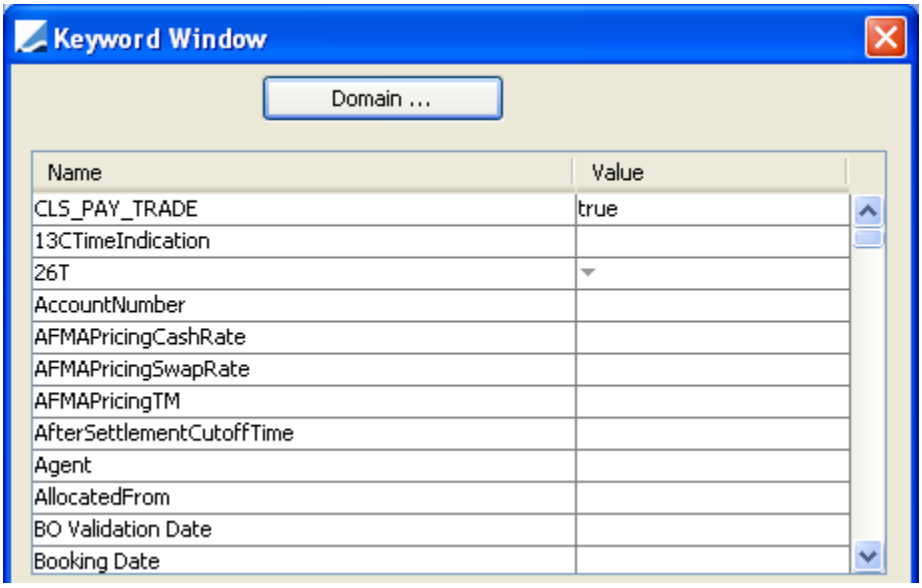



TransferAgent(6,000,000.00 GBP) -PO is Calypso Control Branch (8008) - Version : 0 Mod User :(calypso_user) Cu...
Trade
Back Office
Analytics
Pricing Env
Market Data
Utilities
Help
Template
Trade
Details
Fees
Trader
Sales
Current Trade DateTime
Action
AMEND
Status
VERIFIED
Market Type
NONE
Subsidiary
StepIn Transferor
Calc Agent
Bundles...
Remove From Bundle
Show ...
Mirror
Comment
created from CLS Pay-In schedule
BackOffice...
Keyword...
Internal Ref.
External Ref.


Keyword Window
Domain ...
Name
Value
CLS_PAY_TRADE
SweepPayOut
13CTimeIndication
26T
AccountNumber
AFMAPricingCashRate
AFMAPricingSwapRate
AFMAPricingTM
AfterSettlementCutoffTime
Agent
AllocatedFrom
BO Validation Date
Booking Date


TransferAgent(6,000,000.00 GBP) -PO is Calypso Control Branch (8007) - Version : 0 Mod User :(calypso_user) Cu...
Trade Back Office Analytics Pricing Env Market Data Utilities Help Template
Trade Details Fees
Cpty CALYPARSMCB ProcessingOrg Status VERIFIED ID 8007
Book CLS_TRANSFERAGENT_BOOK Template NONE
Basic Data
Trade Date 12/01/2009 6:30:00 AM Settle Date 12/01/2009
Cash Transfer Type PRINCIPAL Linked Id 0
Cash Transfer Details
Principal 6,000,000 Ccy GBP
Settlement Instructions
Msg Agent SDI Account
From CLS BANK SWIFT/CLS BANK RSMCB ACCOUNT AT CLS
To JP MORGAN CHASE SWIFT/JP MORGAN CHASE/CALYPARSMCB CL... ACCOUNT AT JP MORGAN


TransferAgent(6,000,000.00 GBP) -PO is Calypso Control Branch (8007) - Version : 0 Mod User :(calypso_user) Cu...
Trade Back Office Analytics Pricing Env Market Data Utilities Help Template
Trade Details Fees
Trader
Sales
Current Trade DateTime
Action AMEND
Status VERIFIED
Market Type NONE
Subsidiary
StepIn Transferor
Calc Agent
Bundles...
Remove From Bundle
Show ...
Mirror
Comment created from CLS Pay-In schedule
BackOffice...
Keyword...
Internal Ref.
External Ref.



[NOTE: These trades are necessary to properly maintain the inventory position]

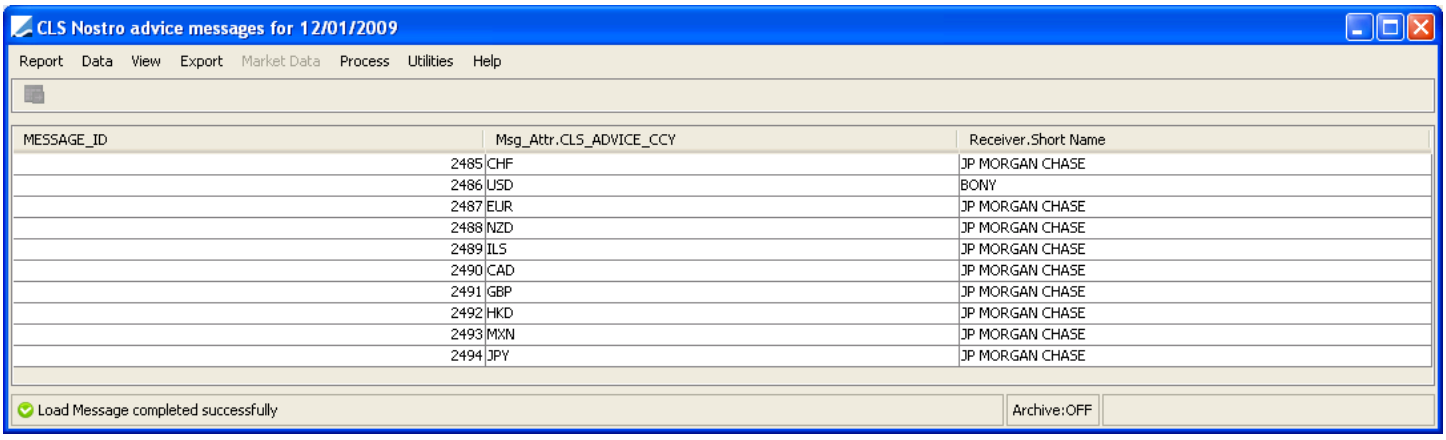
In the case where there is a sweep account involved for the Pay-Out trades you may look to set up a Kick-Off/Cut-Off workflow rule on your transfers or messages to prevent the sweeping to occur before the Pay-Out from CLS to the Nostro Agent is received (to intraday interest on your CLS account).

At this point, as my Nostro Agent Legal Entity attribute CLSMT298Trigger is defined to generate my MT298 messages at the loading of a 'revised' Schedule.

When messages are generated, the CLS_PAY_TRADE trade keyword is set to "SweepPayOut".

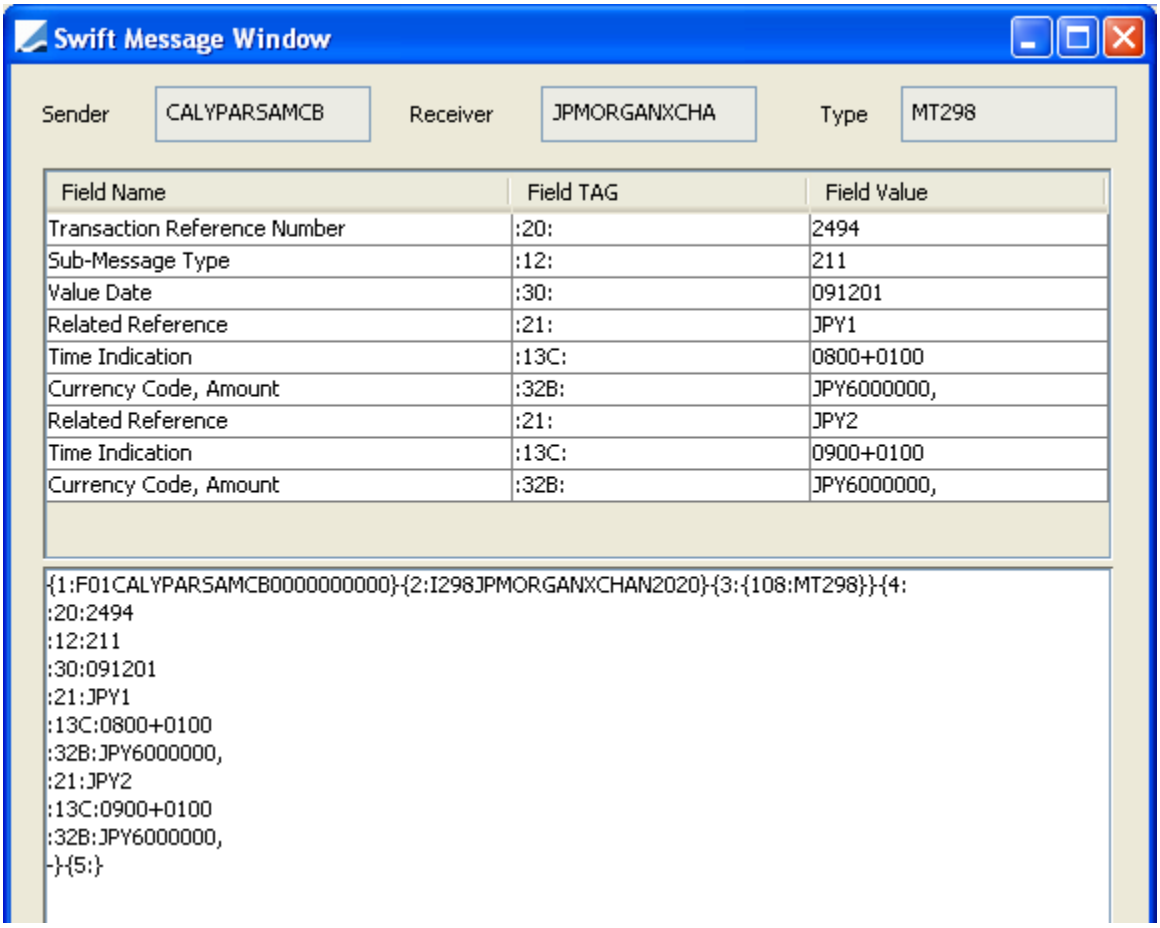
6.3.4 Show MT298

You can right-click a revised Pay-In Schedule and choose Process > Show MT298 to view the MT298 message.

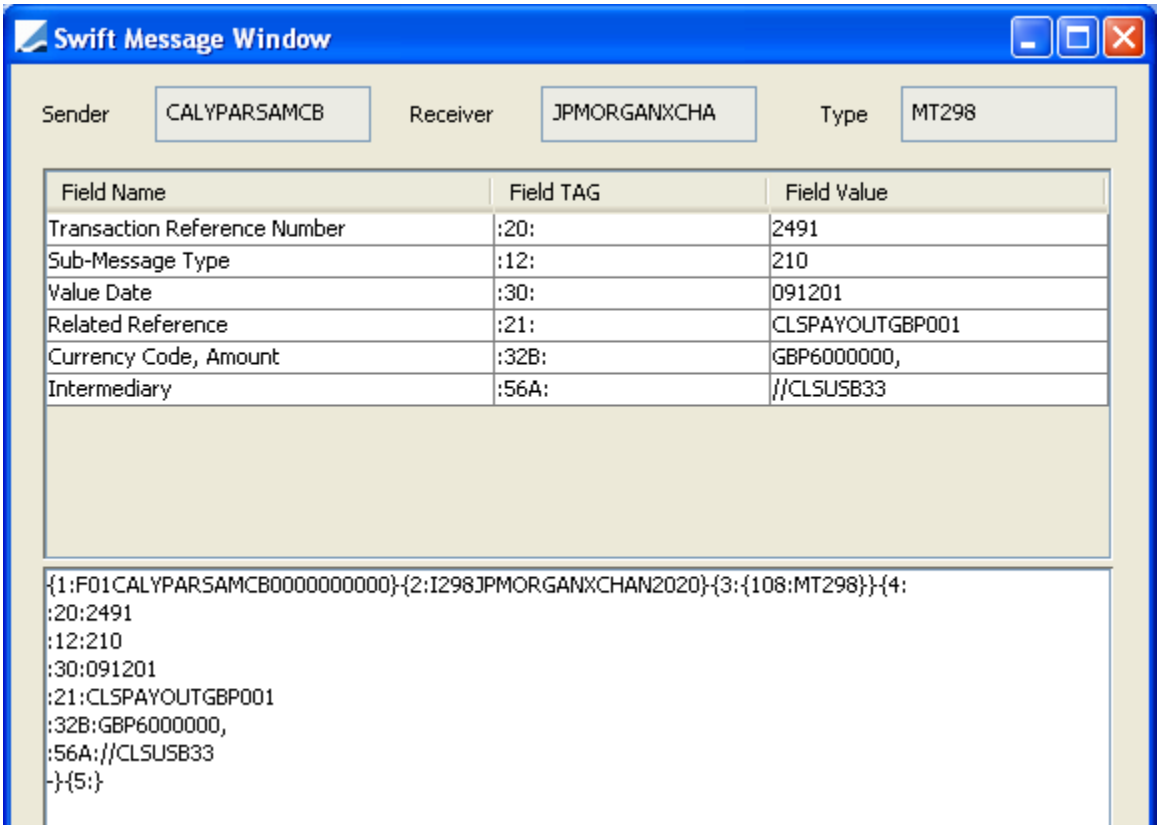


Notice that the attribute Nostro Agent CLSMT298Format is defined as 'tagged'. The Pay-In message will have tags ":20:", ":21:", ":13C:", ":32B:" generated and Pay-Outs will have tags ":20:", ":21:", ":32B:", "56A:"

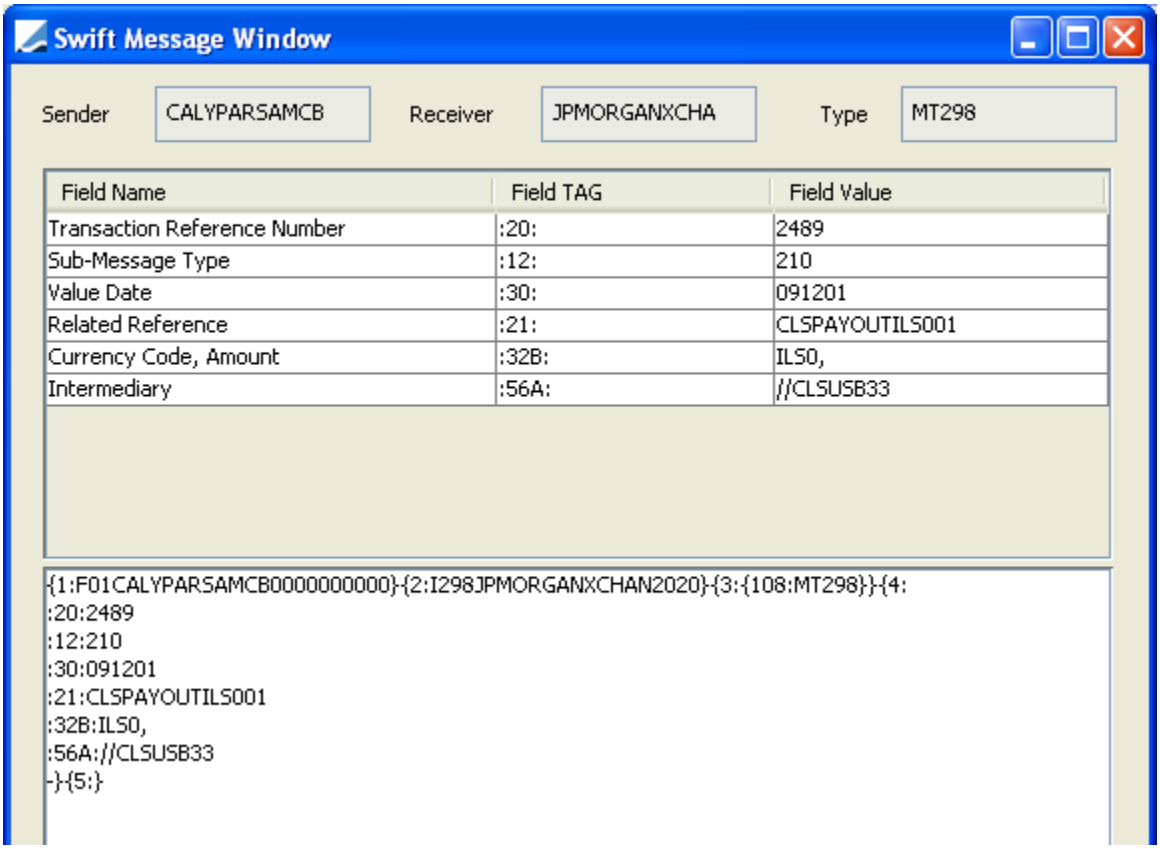
Pay-In MT298's (if we reuse the JPY Pay-In scenario as above) will be formatted as the below:



Pay-Out MT298's (if we reuse the GBP Pay-Out scenario as above) will be formatted as the below:



MT298 for any currency that has a “0” amount will look like the below:



6.4 Manual Generation

Generate MT 298

In the event that the automatic generation of MT298s malfunctions for any reason, the below dialog box from the CLS Pay-In Schedule Report via the menu item Process > Generate MT 298 allows you to indicate for which Ccy you would like to generate your MT298’s. This provides a manual method to generate them per Ccy type.

You can select one or multiple currencies. Note that only currencies defined with the CLS group will appear for selection.

If the agent is not configured for this kind of schedule (initial/revised) in your Legal Entity Attribute CLSMT298Trigger you will get a warning. Otherwise an event will be generated for the Message engine.

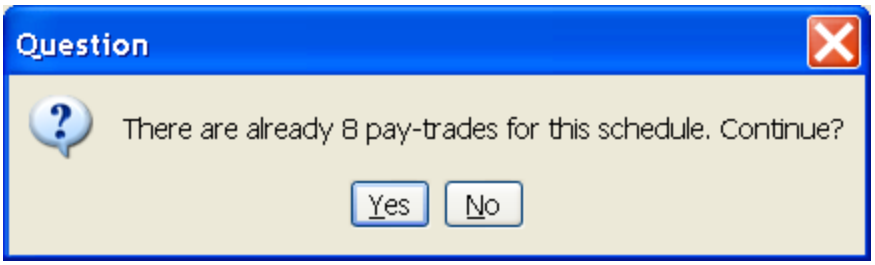
[NOTE: You need access permission function “CLSMT298Generation” to manually generate MT298s messages]

Generate Payment Trades

In the same way, after the loading of a Revised Pay-In Schedule, in the event that the automatic generation of Transfer Agent trades malfunctions for any reason, you can manually generate them via the menu item Process > Generate Payment Trades.

Should you have already the Transfer Agent trades generated, and you try to manually generate, then you will get a warning (as below). In the same way, you will get a warning if Schedule is not reconciled or if schedule is anything but 'revised' and you are trying to generate trades.

Only the first revised pay-in schedule per day will be reconciled and trigger payment/message generation. Subsequent ones (to avoid duplicate payments) will just be saved, and an exception task will be generated.



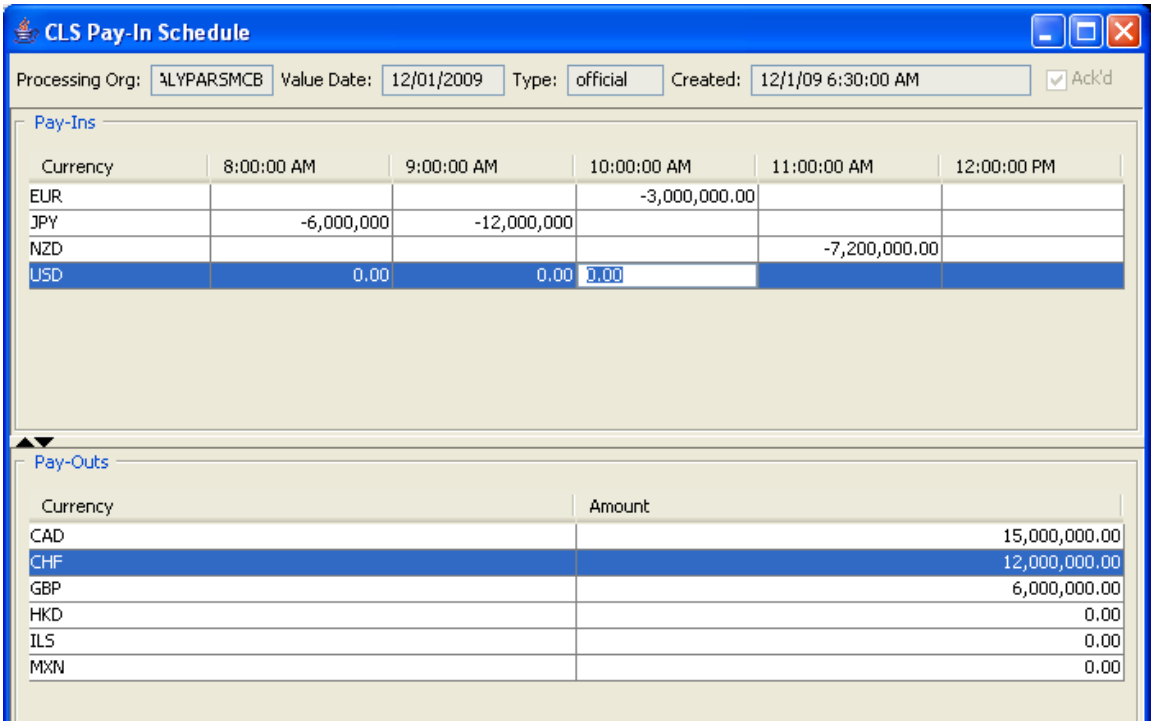
[NOTE: To have access to manually generate Transfer Agent trades you need the access permission function "CLSPayTradeGeneration"]

Edit Schedule

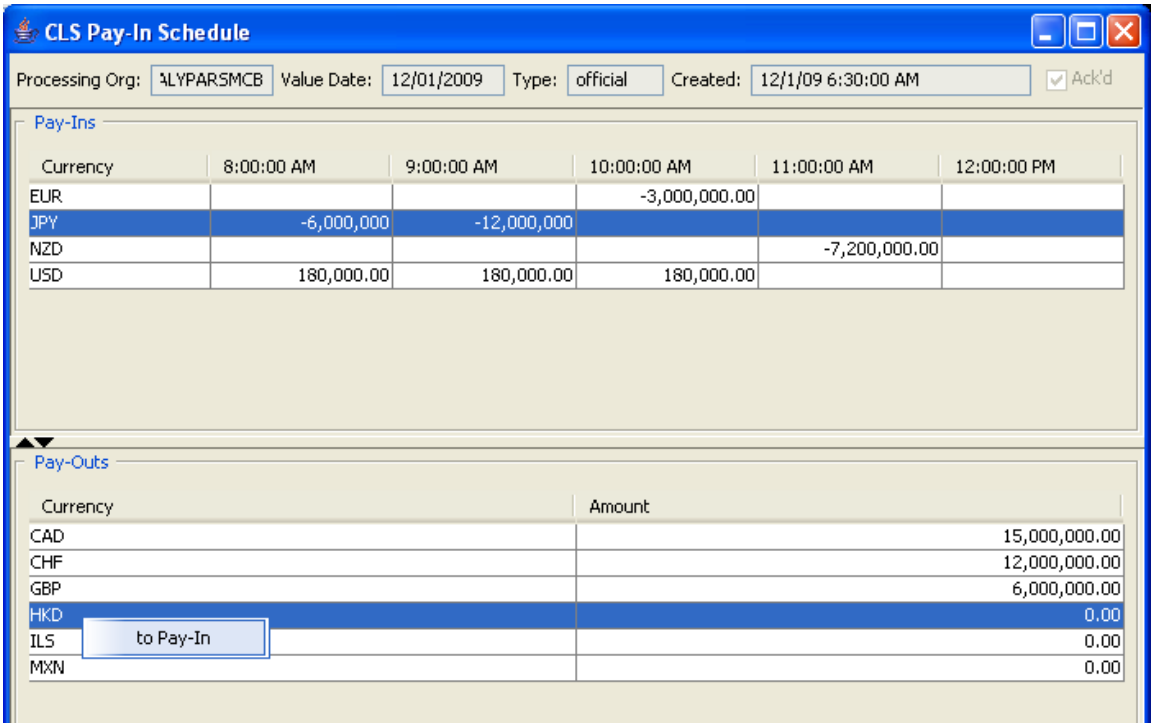
In an extreme case where the loading of a Revised Pay-In Schedule does not work, Calypso allows you to edit your downloaded Initial Pay-In Schedule and resave it as a Revised Pay-In.

Should this occur, then the reconciliation and generation of Transfer Agent trades and MT298 Messages has to be done manually.

To do this, choose the menu item Process > Edit Schedule, and manually adjust the schedule as desired. By double clicking on a time slot, you can adjust the amounts.



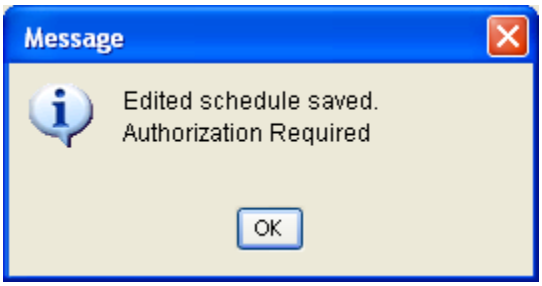
As per the below, by right clicking on a currency you can move them from Pay-In to Pay-Out or vice versa.



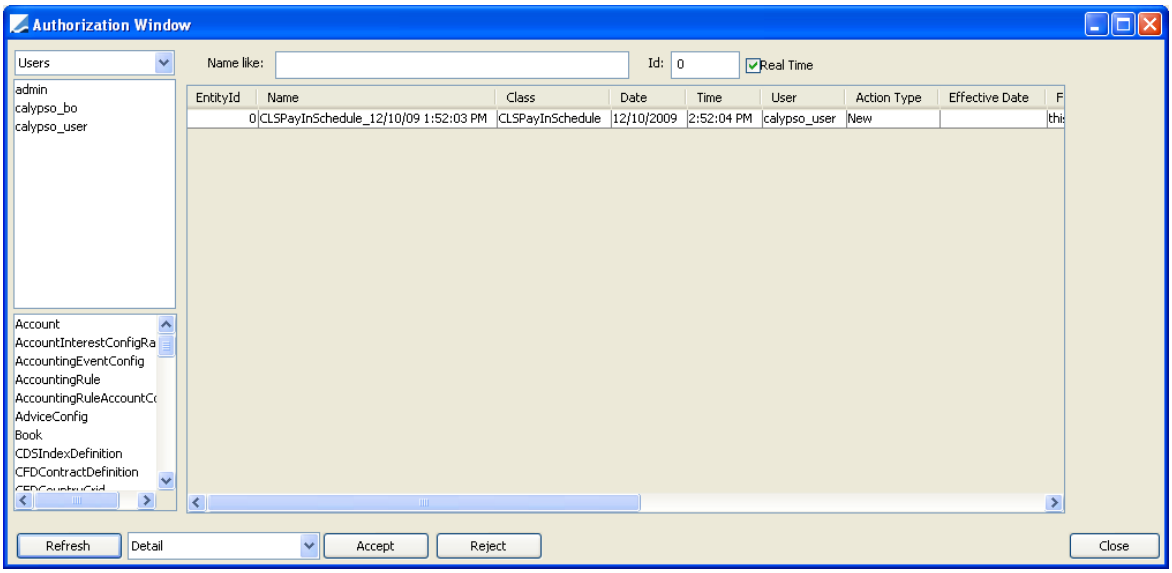
[Note: When an amount is changed in an editable cell, the prior times to the currency are set to the same value. E.g., if you set the 08:00am amount, the 09:00am amount and the 10:00 am amounts will be set to

the same value as well. If you want to pay everything in one sum, this can be done by setting the 12:00 am (or 10:00am for the Asian currencies). You can move currencies to the Pay-Out via a Popup menu. An error will appear if amounts for the Pay-In are not increasing. A save in that case will fail]

Once you have made the necessary changes, you can 'Save' the Schedule and doing that will instigate a pop up message letting you know that editing a Schedule requires authorization.



Thereafter in order for the Revised Pay-In Schedule to be saved with your changes, you need to go to Processing > Data Authorization from the Calypso Navigator, and 'Accept' the saved changes.

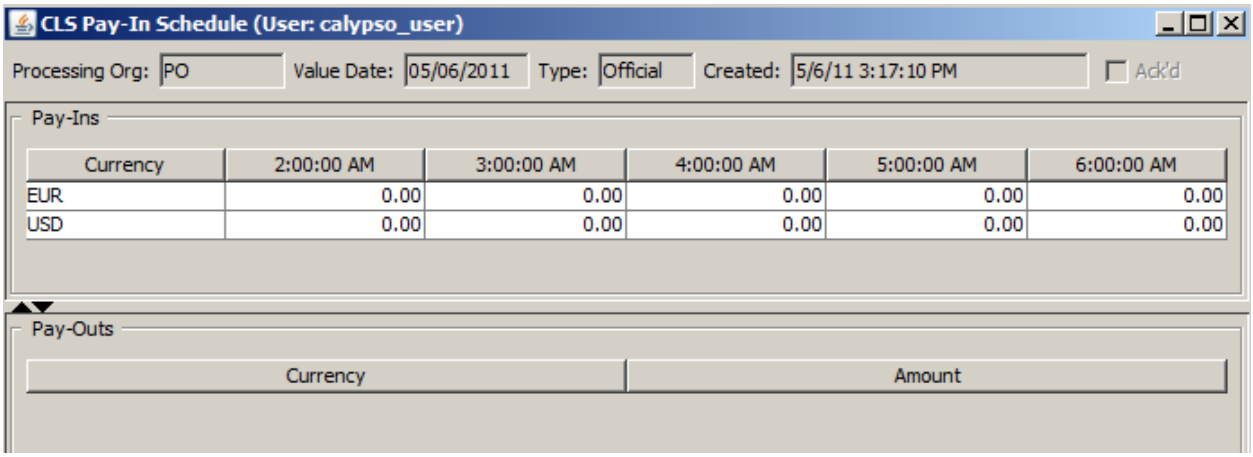


Create Schedule

In case neither an Initial Pay-In Schedule (IPIS), nor a Revised Pay-In Schedule (RPIS) are received, you can create an empty Pay-in schedule manually using Process > Create Schedule.

You will be prompted to select a PO and enter a value date.

The empty schedule that is generated has all currencies (defined with CLS group) with amount zero in the Pay-in section, and an empty Pay-out section.



You can right-click a row and choose “to Pay-Out” to move the row to the Pay-Outs area.

You can modify the amounts as needed.

Click **Save** when you are done.

The new schedule must be authorized.

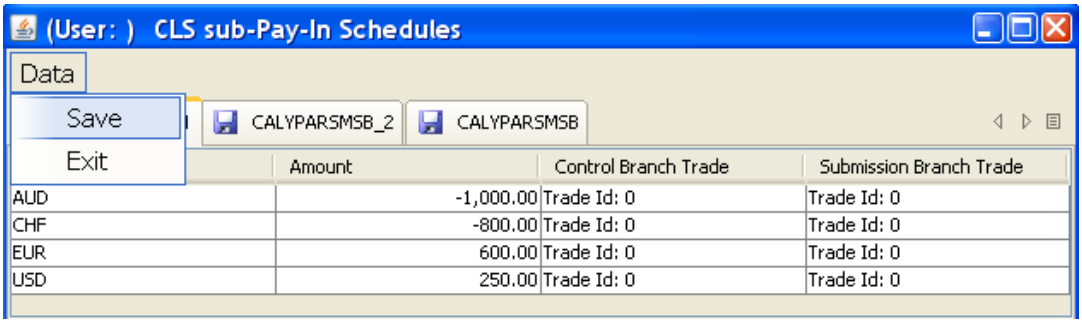
Generate Submission Branch Sub-Schedule

You can generate the sub-schedule for a given branch.

Select a row in the CLS Pay-In Schedule report and choose Process >Generate Submission Branch Sub-Schedule. You will be prompted to select a value date and a set of branches.

Note that only submission branches that have a CLSControlBranch legal entity attribute defined will appear for selection.

The Sub-Schedule can be generated for any transfer that is not canceled (transfer status does not belong to domain “transferCanceledStatus”).

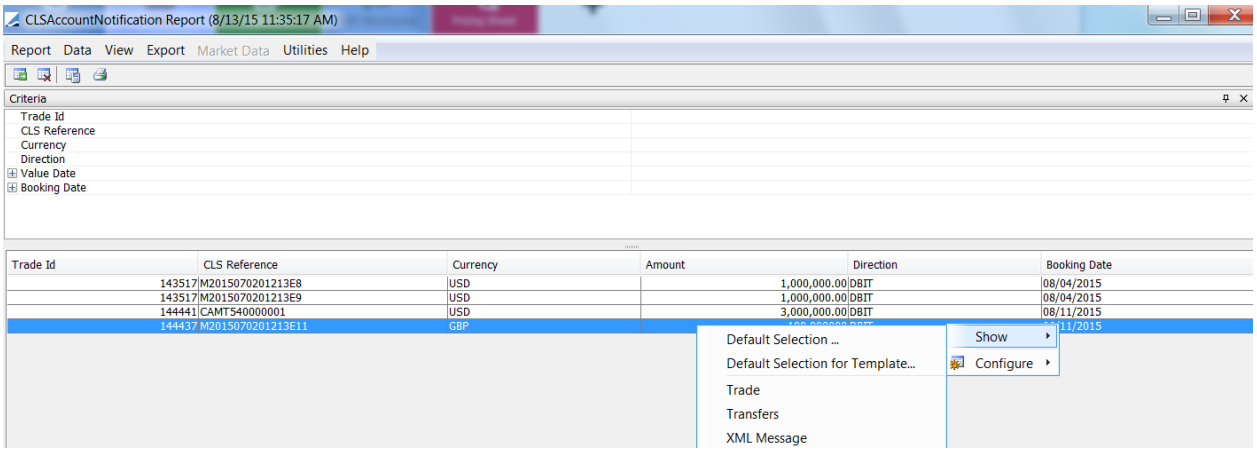


For each currency a Cash Transfer trade is generated between the submission branch and the control branch, with a CLS_ADJUSTEMENT fee. The trade keyword CLS_PAY_TRADE is set to SubPayIn or SubPayOut depending on the direction of the trade.

Choose Data > Save to save the trades.

Integration of Pay-In and Pay-Out XML Files from CLS

Pay-In and Pay-Out XML Files from CLS can be integrated using the CLS Message engine, and can be viewed in the CLS Account Notification Report (menu action `reporting.ReportWindow$CLSAccountNotification`).

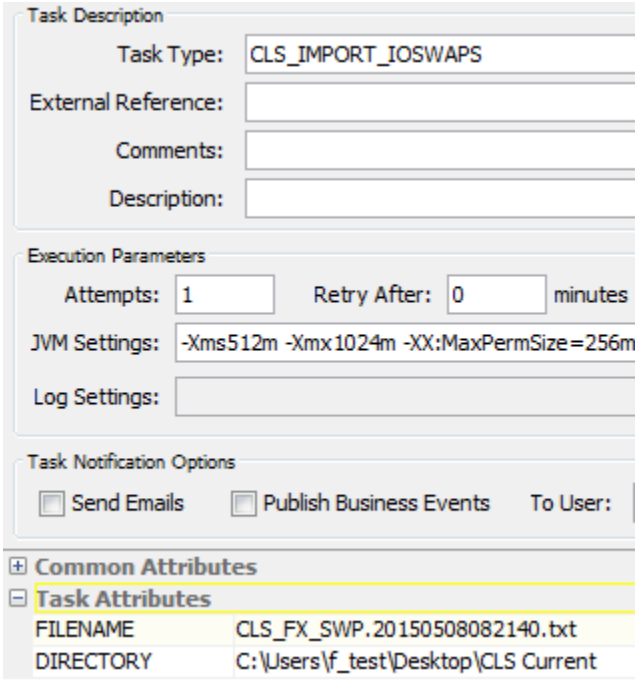


Pay-In messages are attached to the corresponding Pay-In Transfer Agent trade generated after the RPIS process. Once mapped, both transfers the one impacting CLS account and the one impacting the Nostro Account will be SETTLED.

Pay-Out messages are attached to the corresponding Pay-Out Transfer Agent trade generated after the RPIS process. Once mapped, only the transfer impacting the CLS account will be SETTLED.

6.5 Scheduled Task CLS_IMPORT_IOSWAPS

When I/O Swaps are made available by CLS you can run the scheduled tak CLS_IMPORT_IOSWAPS to import them.



Attributes

FILENAME: You can define:

- The exact name of the file to process.

  Examples:

  CLS_FX_SWP.20150508082140.txt

  CLS_FX_SWPBody.20150508082144.txt

- A file pattern like “CLS_FX_SWP_<yyyyMMdd>[0-9]*.txt”

  If you define the FILENAME as the pattern above, and you have in your DIRECTORY the following files:

  CLS_FX_SWP_20150812.txt

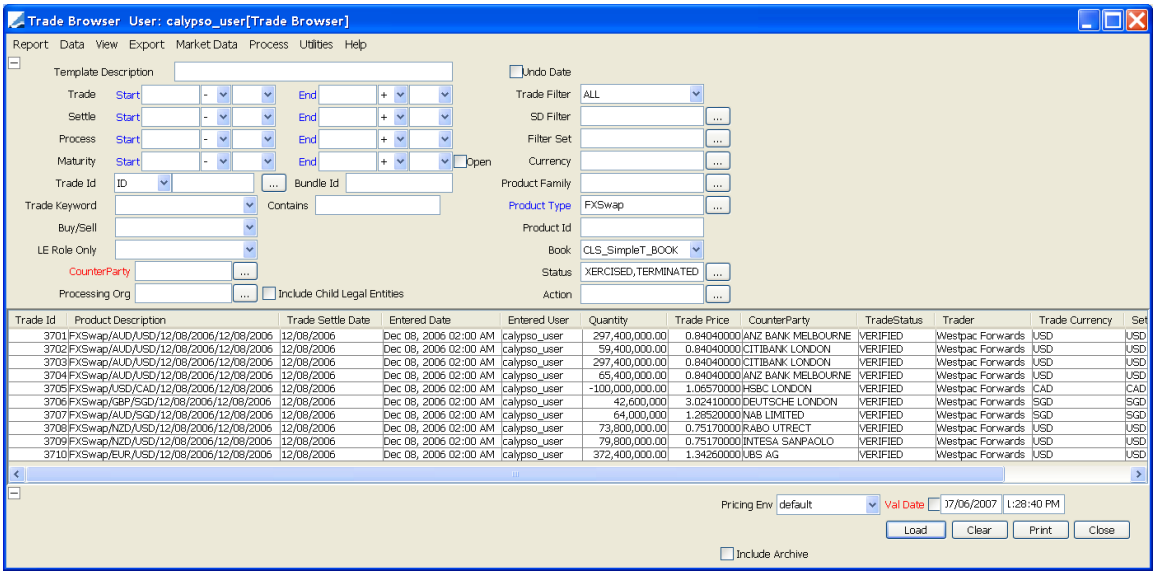
  CLS_FX_SWP_20150807.txt

  The new logic will allow the scheduled task to process the file defined with the date equal to the process date of the scheduled task.

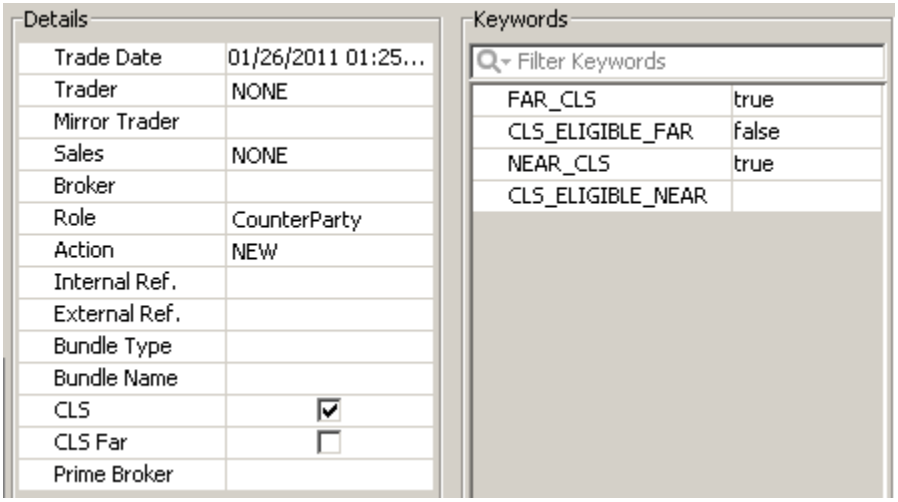
DIRECTORY: Define the directory path where the files to process are located.

Example: “C:\Users\f_test\Desktop\CLS Current”.

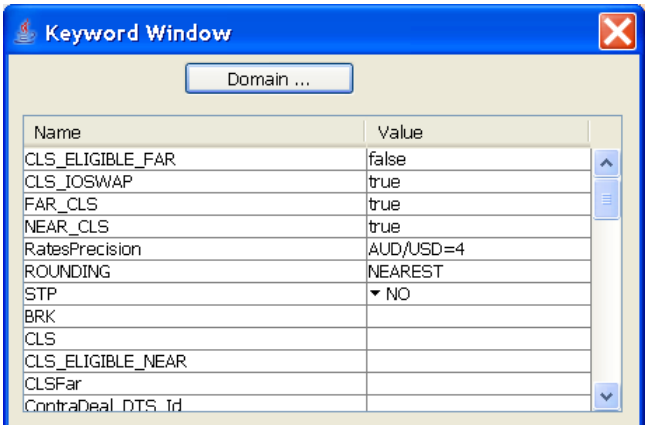
This will create FX Swap trades and book them into the book which you have defined in the “clsParameters” domain for the value “ioswapBook.<legal entity short Code>”.



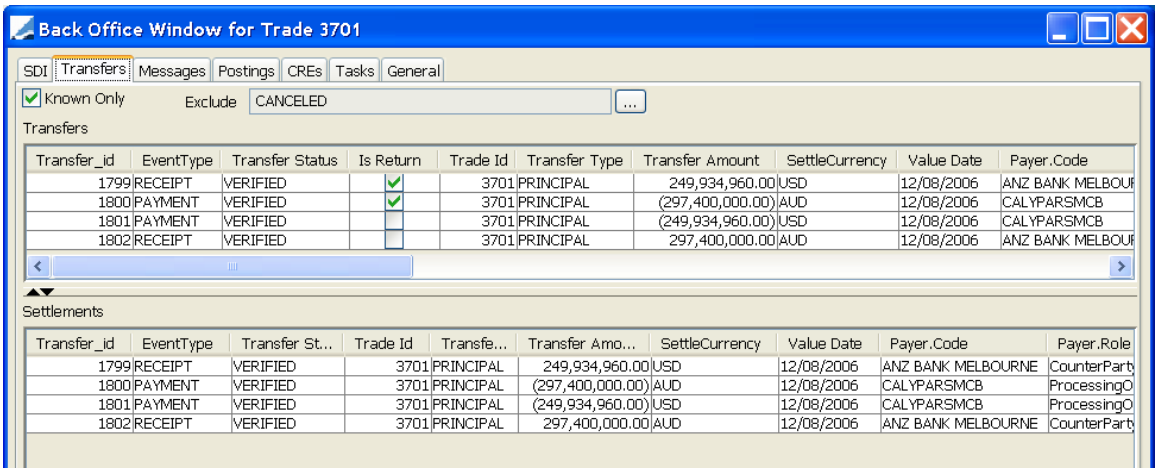
The FX Swap trades generated will have one leg in CLS and another Out of CLS. The checkbox on the trade screen will indicate Near Leg to be in CLS and Far leg to be out.



In addition, for these trades, trade keyword CLS_IOSWAP will be marked to true.



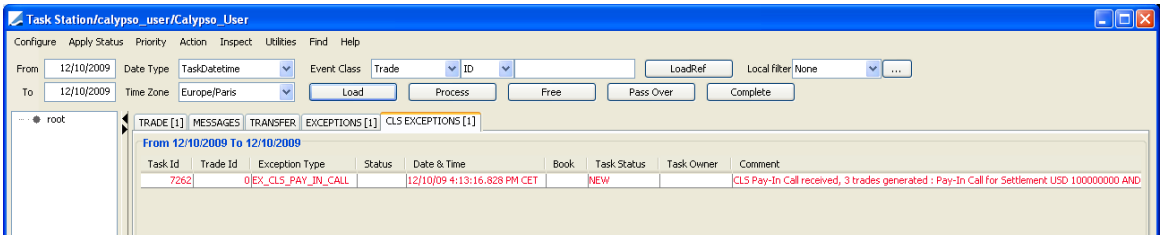
And on their transfers to facilitate recognition of which legs are the Near and which are the Far, an IS Return check box will mark the Far leg.

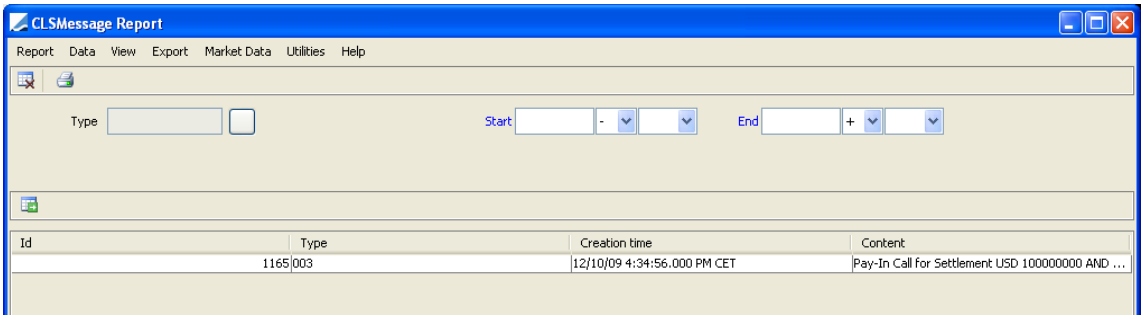


[Note: Be sure to have defined REUTERS Legal Entity Attribute for these counterparties to be fed in correctly and that your SDI is defined correctly so that each leg can be correctly settled, i.e. Near Leg in CLS and FAR leg out of CLS. You need to have set up counterparties with SDIs for CLS and another outside of CLS]

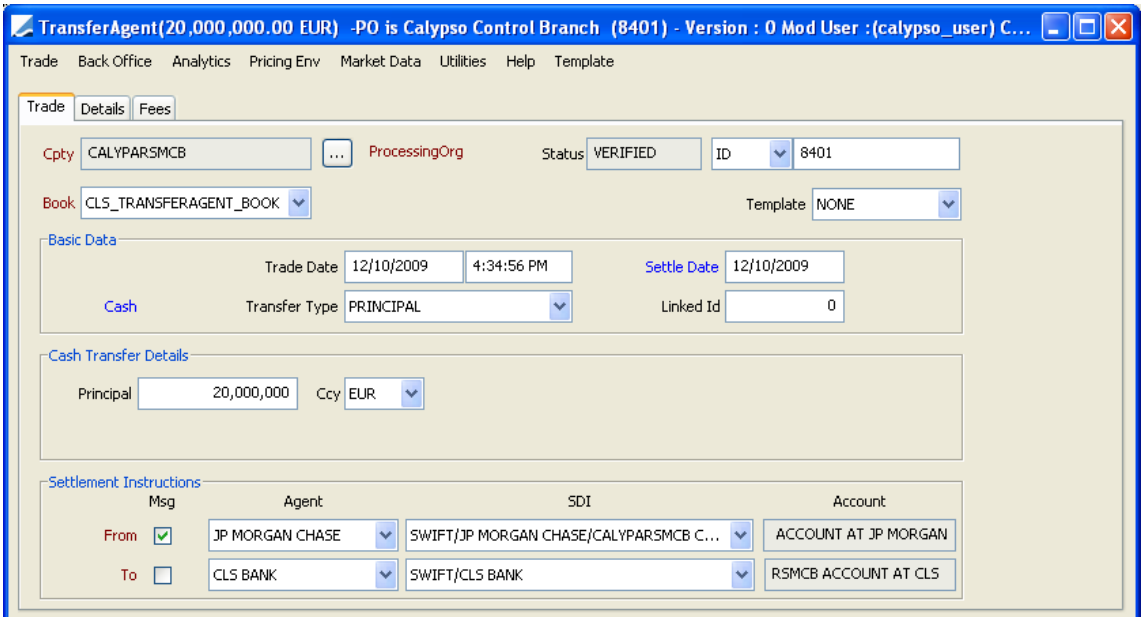
6.6 Pay-In Calls

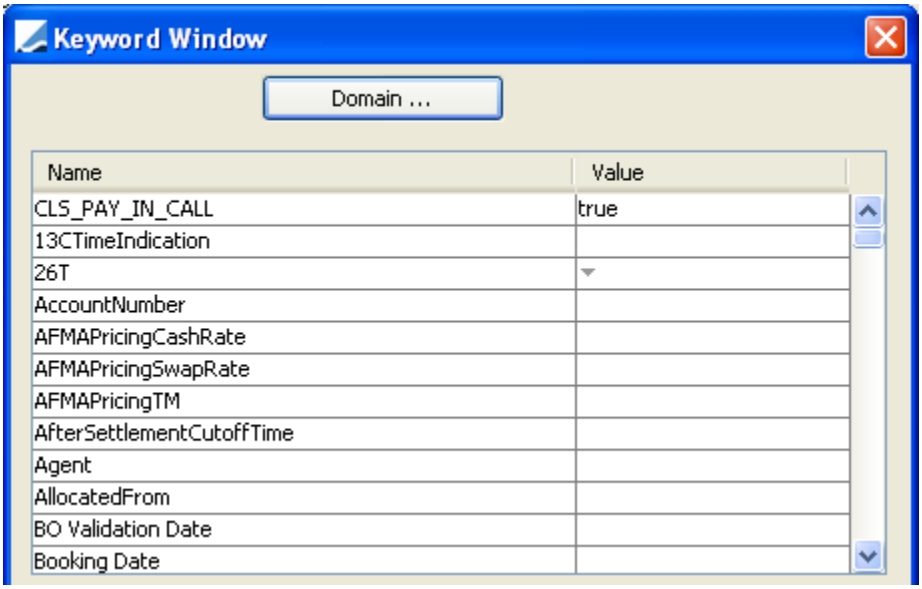
In the event Pay-In calls come in then there will be both an exception in the Task Station which will give notice that Pay-In Calls are available, and trades have been generated and the CLSMessageReport will also inform of the PayIn Call loading.



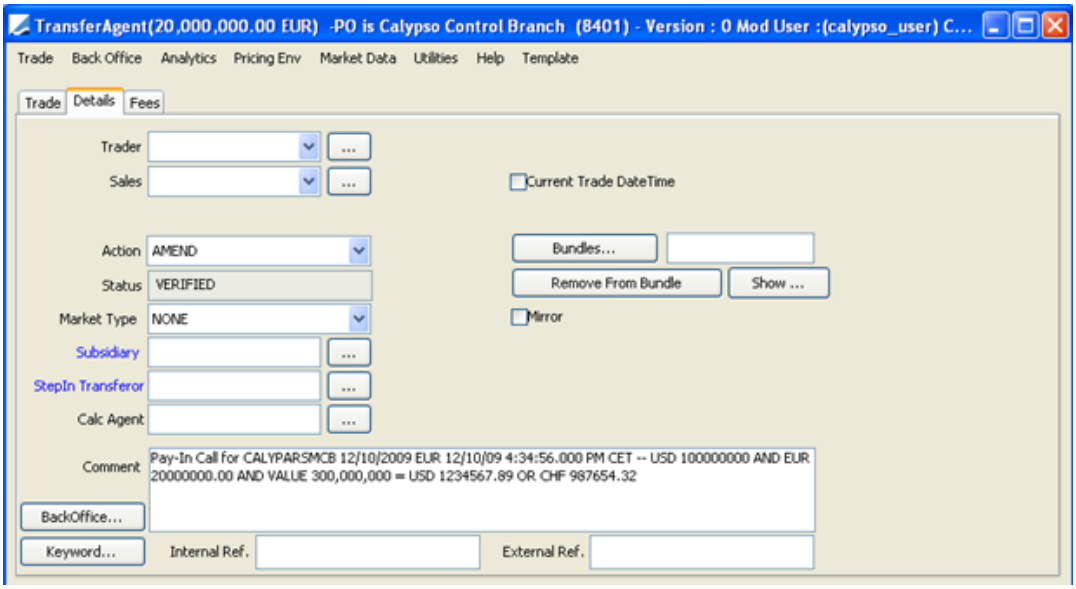


The Transfer Agent trades are automatically generated at such an event. These trades will be marked with trade keyword CLS_PAY_IN_CALL (this is a system keyword and user cannot edit).





In the same way as the Transfer Agent trades generated from Pay-In Schedules, these trades will also have a comment filled in the details tab of the trades.

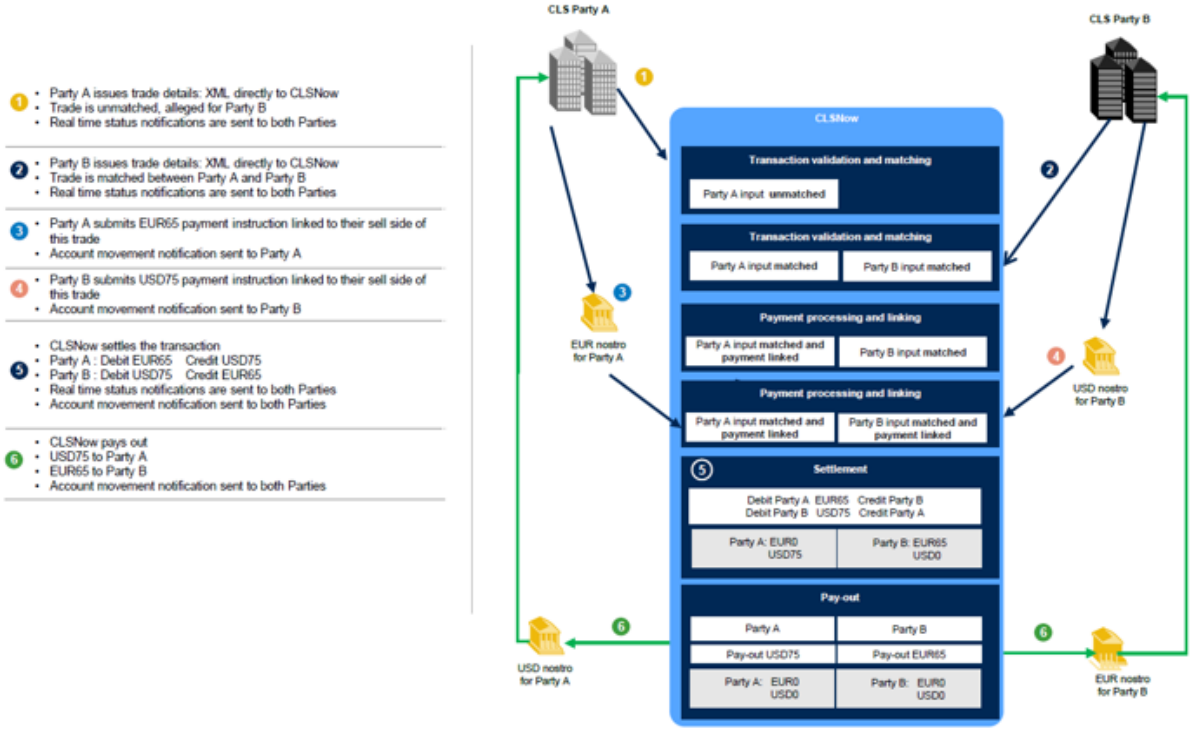


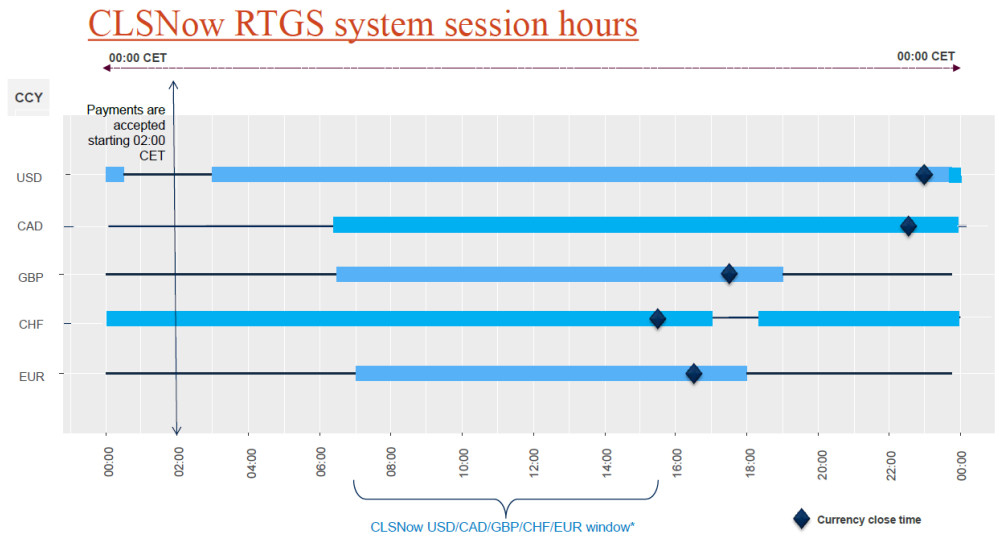
In the event that you would like to hold back the generation of the Pay-In Call's then, under "clsParameters" you need to go and add the domain value payInCallBook.<legal Entity Short Name> with comment as NONE.

CLS Now

CLS Now is a service launched by CLS in 2019, a bilateral same-day payment-versus-payment gross settlement initially in CAD, CHF, EUR, GBP and USD. The service enables counterparties to optimize the use of available liquidity in the same-day market while mitigating settlement risk.

At one glance, CLS Now processing:





In terms of Calypso processing, it is similar at CLS Settlement processing, and so we reuse the same logic:

- Identify when a trade is eligible to this new service
- Submit the trade to CLS
- Integrate CLS respond
- Process payments

7.1 CLS Now Eligibility

Trades eligible to CLS Now must meet the following requirements:

- Parties are CLSNow participant
- Parties have a CLSNow agreement
- Trade is same day FX: trade date = value date
- Currency pairs: both USD, CAD, GBP, CHF, EUR
- SDI method= CLSNow
- Execution Time = before earliest currency soft close

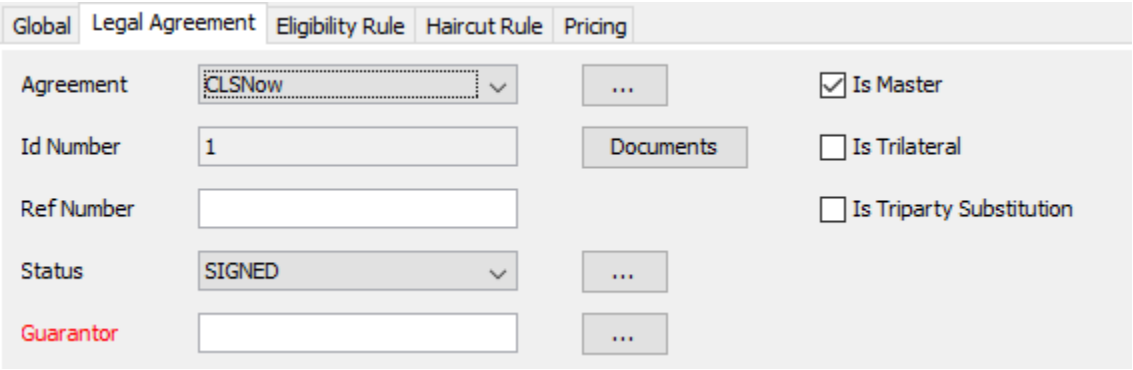
  Best practices:

  CLSNow currency close (currently) to be set at 90 minutes prior to RTGS close with a “soft close” (an internal participant close after which new trades would not be submitted for settlement in CLSNow) set at 120 minutes prior to RTGS close. CLSNow however will not reject trades submitted after the “soft close” in order to continue to facilitate settlement via CLSNow

Mark a trade ‘CLSNow = true’ to identify that trade is CLSNow eligible.

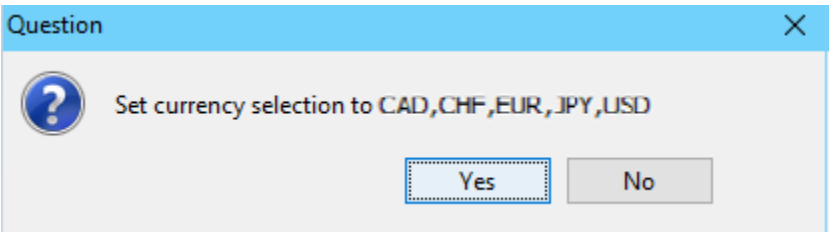
- PO and Counterparty must be CLS Now member:
- CLSNowPartyId legal entity attribute must be set
- CLSNowBranchId legal entity attribute must be set

A Legal Agreement must be defined between the PO and the Counterparty

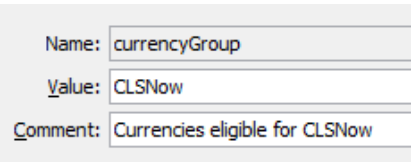


With Agreement = CLSNow

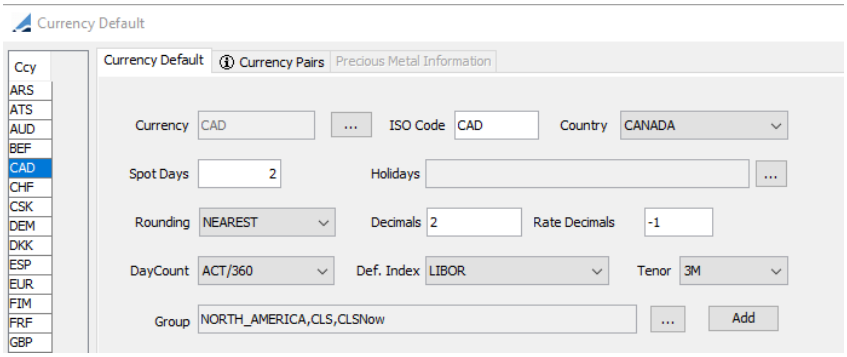
As per CLS agreement, when user selects CLSNow agreement, a popup window appears to display:



The list of the currencies is based on Currency Definition and the currency group CLSNow, it will be created automatically by the CLSSchemaData.



USD, CAD, GBP, CHF, EUR Currencies are defined in this new Group.



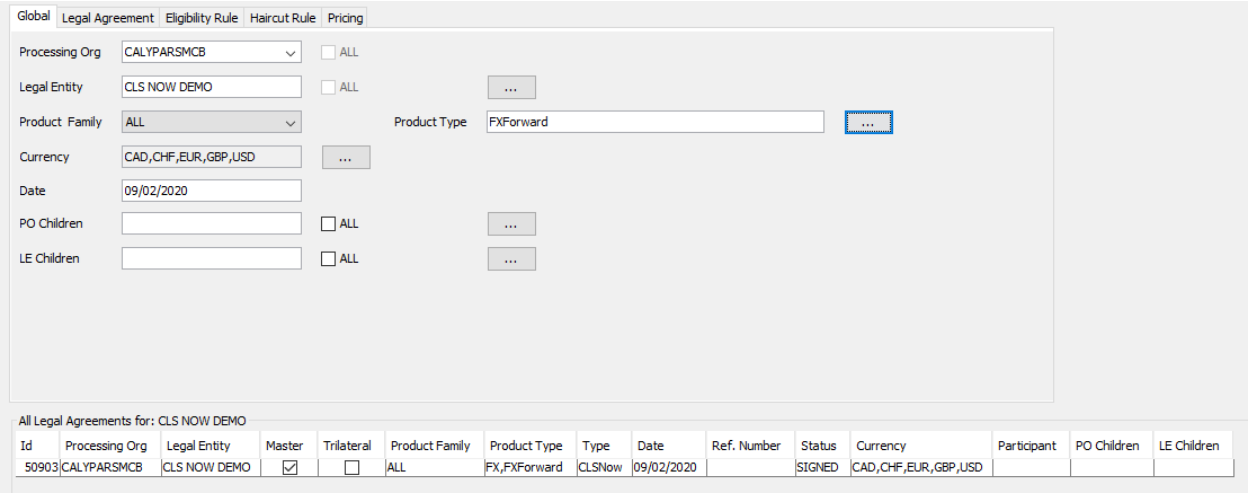
Note that FX trade which has Trade Date=Settle Date is by default in calypso a FX Forward (input comes from FX PM). We can leave FX Product family or set specific Product Type = FXForward

Trade date = settle date

SDIs

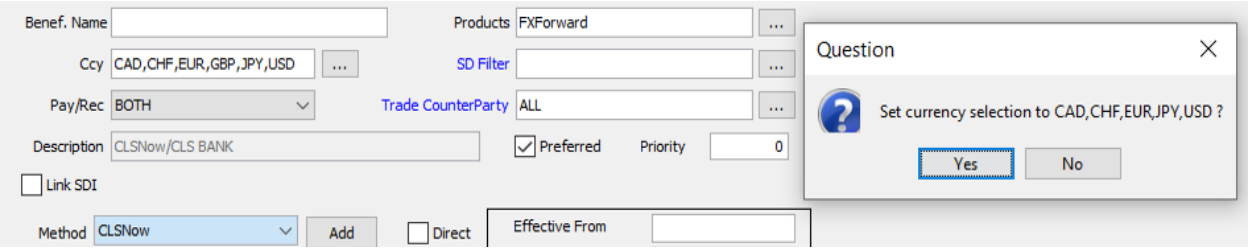
- Method = CLSNow
- Settlement Method = CLSNow
- This settlement method is included in the logic of the environment property CLS_SDI_SELECTION

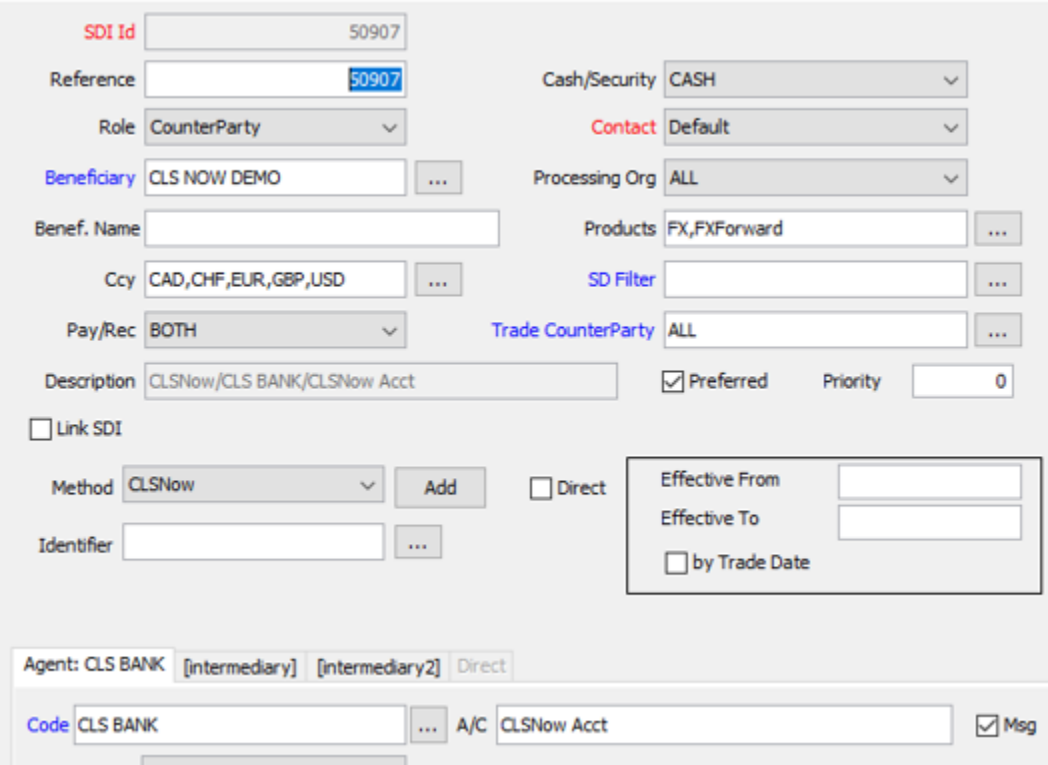
  If CLS_SDI_SELECTION = true and if a trade is flagged with CLSNow = true, only CLSNow SDIs will be automatically assigned



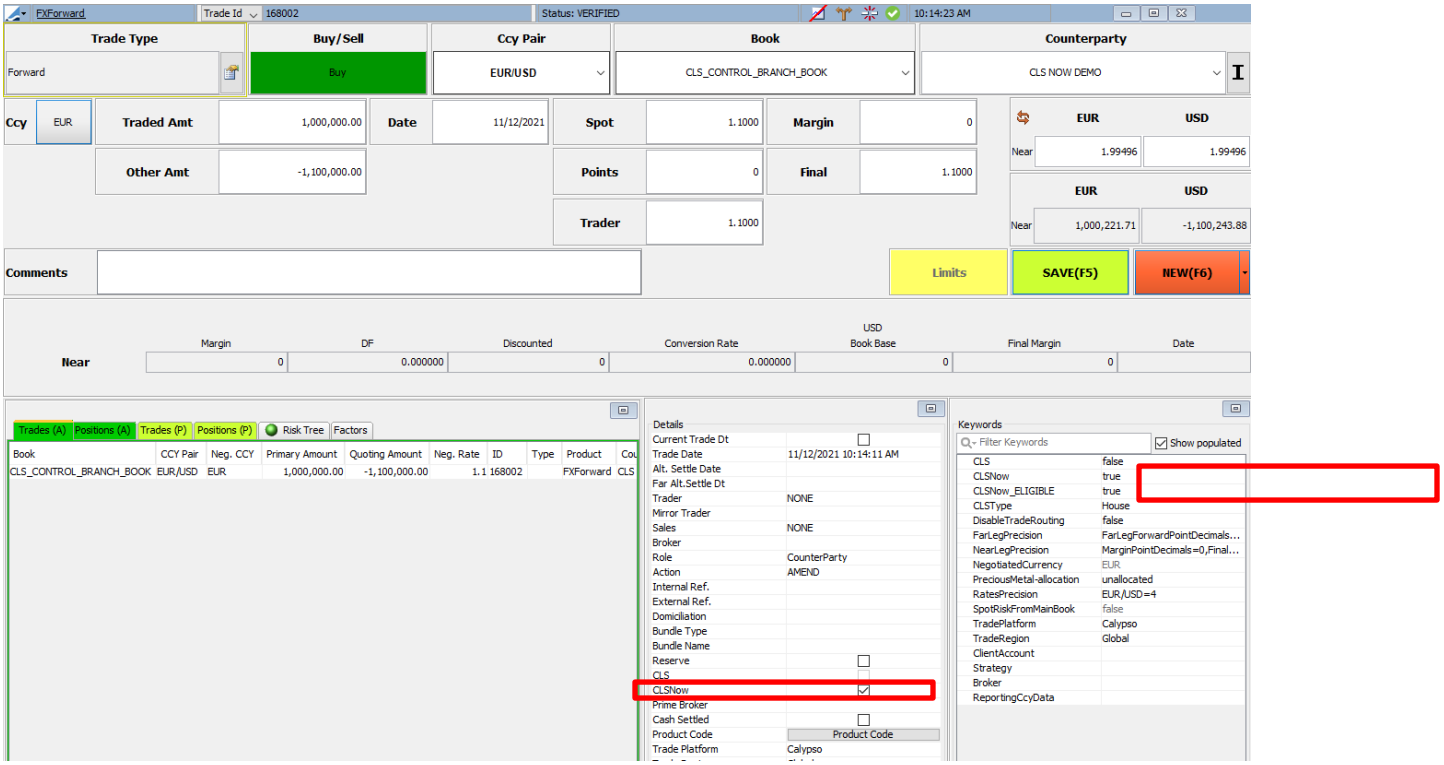
Same logic applied for SDI configuration:

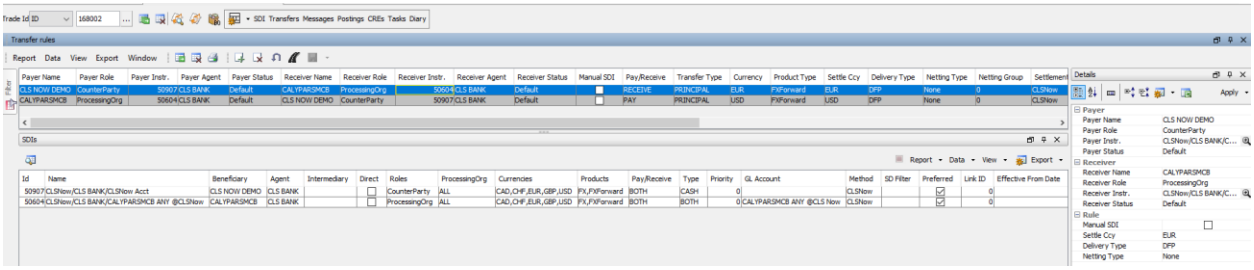
When created the SDI with CLSNow method:





Then in the trade window:

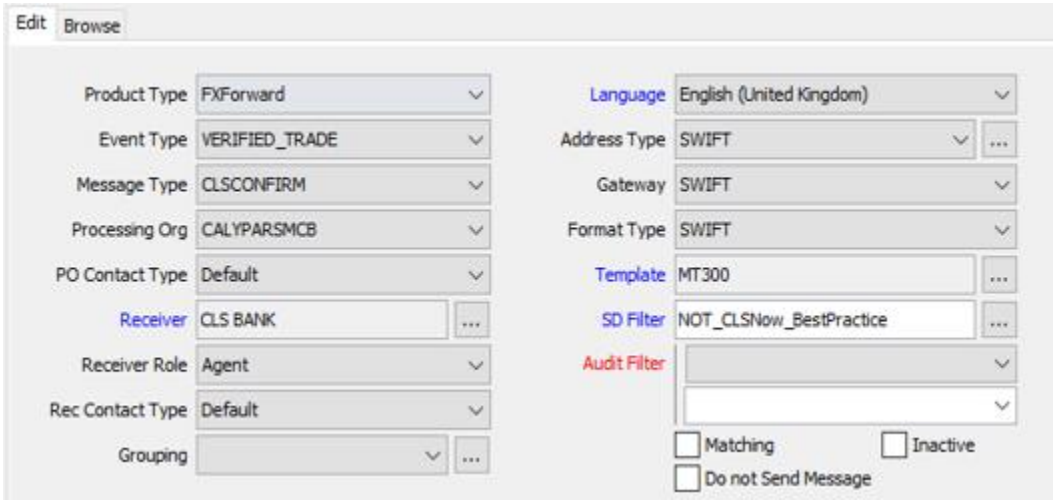


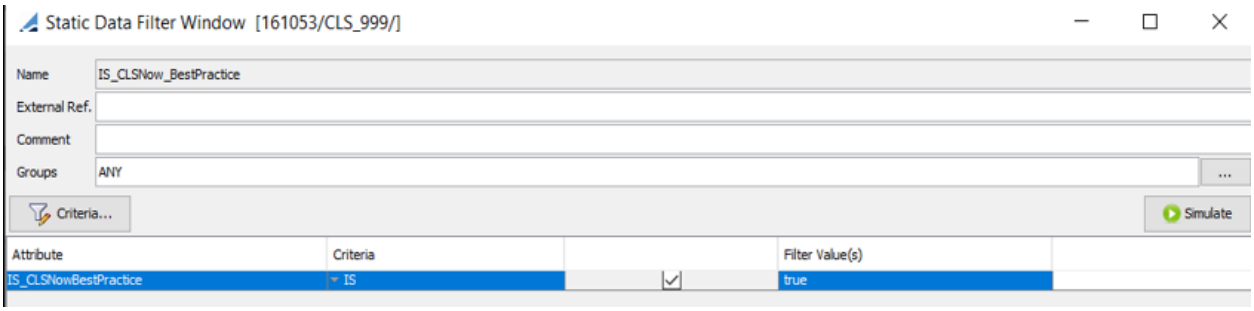


7.2 Message processing

Same messages will be exchanged between CLS Member and CLS platform: MT304 or fxtr

Note that fxtr015 is not used for CLSNow. In case of amendment, best practice is to use a cancellation and a new trade message.





Messages supported version	From Member to CLS	From CLS to Member
Bulk Trade Status Notification Versions		fxtr.030.001.04
Long Trade Status Notification		fxtr.017.001.04
Short Trade Status Notification		fxtr.008.001.06
Message Rejection		admi.002.001.01
Alleged Trade Withdrawal		fxtr.013.001.03
Account Notification		camt.054.001.06
New Trade Instruction	fxtr.014.001.04	
Rescind Trade Instruction	fxtr.016.001.04	
System Event Acknowledgement	admi.011.001.01	

Trade submission FIN and XML

In order for CLS to route the instructions to CLSNow session, messages must include '**CNOW**':

1- for MT300/304 in field 14E
   a. Field 22A: AMND is not supported
   b. Field 14E: Settlement session must be "CNOW"

```
{1:F01CALYUS33AXXX0000000000}{2:I304CLSBUS30XXXXN2020}{3:{108:MT304000016208}}{4:
:15A:
:20:16208
:22A:NEWT
:94A:ASET
:83J:/NAME/NA
:82A:CALYUS33XXX
:87A:CPNOWDEM
:14E:CNOW
:15B:
:30T:20211112
:30V:20211112
:36:1,1
:32B:EUR1000000,
:53A:CLSBUS30XXX
:57A:CLSBUS30XXX
:33B:USD1100000,
```

```
:53A:CLSBUS30XXX
:57A:CLSBUS30XXX
:58A:/CLSNOW ACCT
CPNOWDEM
-}{5:}
```

2- For XML, in field 'TradInf/SttlmSsnIdr' on fxtr.014/fxtr.016

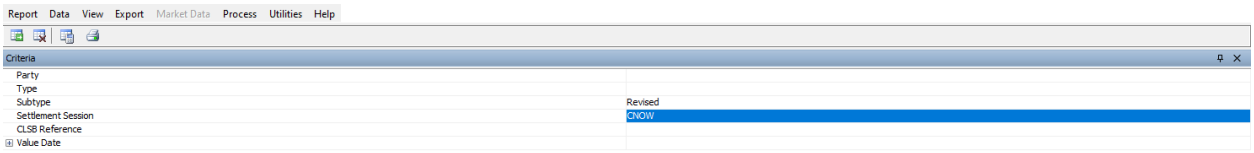
```
<?xml version="1.0" encoding="Cp1252"?>
<Body xmlns:tns="http://www.example.org/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <AppHdr xmlns="urn:iso:std:iso:20022:tech:xsd:head.001.001.01">
    <Fr>
      <FIId>
        <FinInstnId>
          <BICFI>CALYUS33XXX</BICFI>
        </FinInstnId>
      </FIId>
    </Fr>
    <To>
      <FIId>
        <FinInstnId>
          <BICFI>CLSBUS30XXX</BICFI>
        </FinInstnId>
      </FIId>
    </To>
    <BizMsgIdr>16209</BizMsgIdr>
    <MsgDefIdr>fxtr.014.001.02</MsgDefIdr>
    <CreDt>2021-11-12T09:15:02Z</CreDt>
  </AppHdr>
  <Document xmlns="urn:iso:std:iso:20022:tech:xsd:fxtr.014.001.02" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:iso:std:iso:20022:tech:xsd:fxtr.014.001.02 ../../Schemas/fxtr.014.001.02.xsd">
    <FXTradInstr>
      <TradInf>
        <SttlmSsnIdr>CNOW</SttlmSsnIdr>
        <TradDt>2021-11-12</TradDt>
        <OrgtrRef>16209</OrgtrRef>
      </TradInf>
      <TradgSdId>
        <SubmitgPty>
          <AnyBIC>
```

```
            <AnyBIC>CALYUS33XXX</AnyBIC>
          </AnyBIC>
        </SubmitgPty>
        <TradPty>
          <AnyBIC>
            <AnyBIC>CALYUS33XXX</AnyBIC>
          </AnyBIC>
        </TradPty>
      </TradgSdId>
      <CtrPtySdId>
        <SubmitgPty>
          <AnyBIC>
            <AnyBIC>CPNOWDEM</AnyBIC>
          </AnyBIC>
        </SubmitgPty>
        <TradPty>
          <AnyBIC>
            <AnyBIC>CPNOWDEM</AnyBIC>
          </AnyBIC>
        </TradPty>
      </CtrPtySdId>
      <TradAmts>
        <TradgSdBuyAmt Ccy="EUR">1000000</TradgSdBuyAmt>
        <TradgSdSellAmt Ccy="USD">1100000</TradgSdSellAmt>
        <SttlmDt>2021-11-12</SttlmDt>
      </TradAmts>
      <AgrdRate>
        <XchgRate>1.1</XchgRate>
      </AgrdRate>
    </FXTradInstr>
  </Document>
</Body>
```

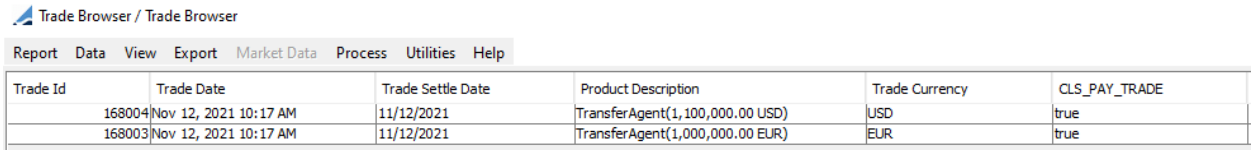
7.3 Payment processing

The system will automatically trigger a transfer agent upon receipt of the **Matched** notification from CLS (fxtr.017).

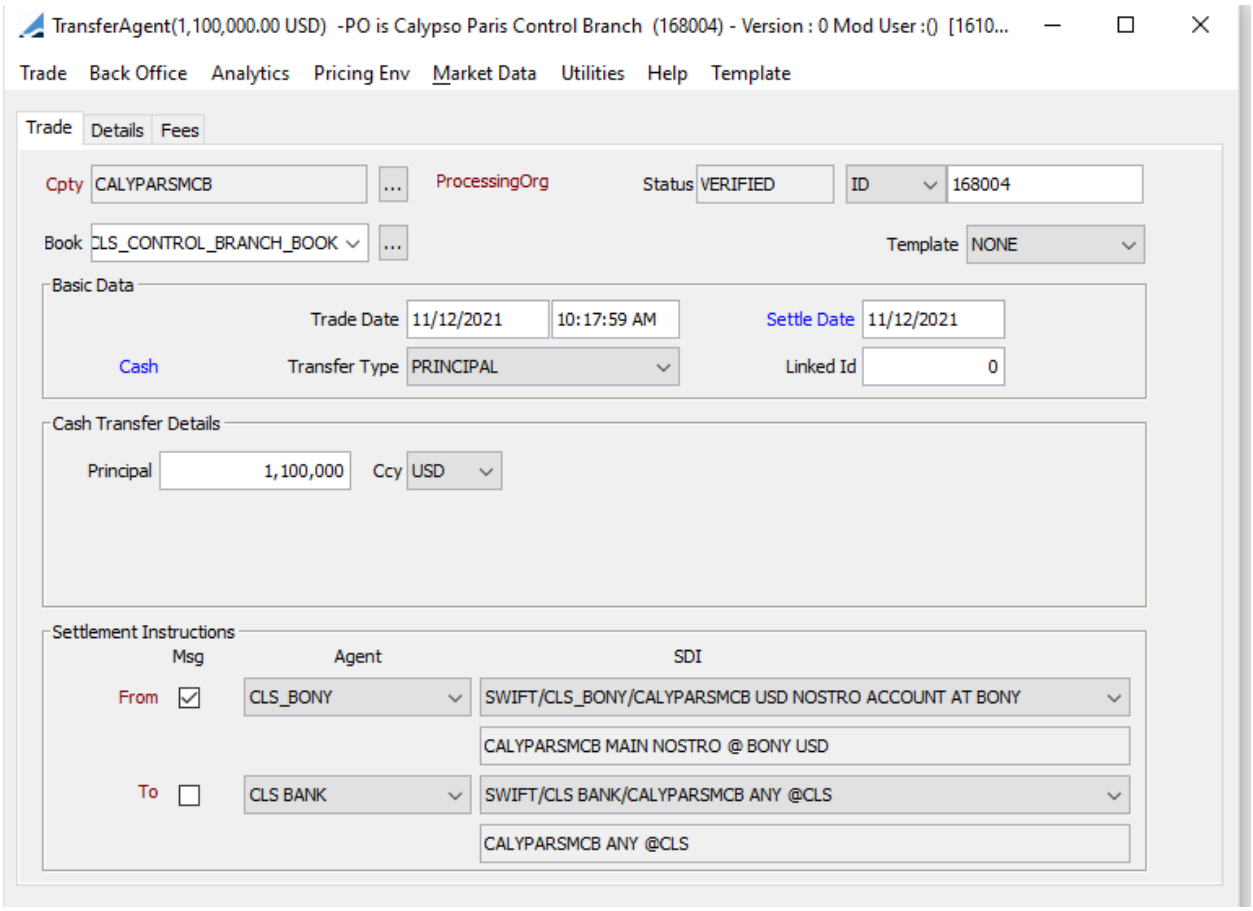
And then by message set up, a payment instruction will be generated. In the system, we will simulate the receipt of RPIS when the trade is moved to MATCHED.

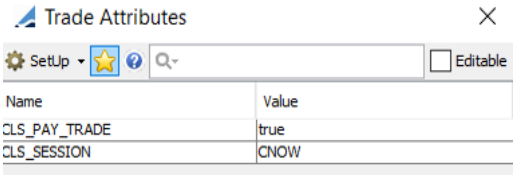


Id	CLSB Reference	Settlement Session	Party Name	Creation time	Value Date	Type	Subtype	Acknowledged	Ack Msg Identifier	EUR	GBP	ILS	JPY	USD
					Value Date: 11/12/2021		Subtype: Revised							
51402	CLSB00000000168002	CNOW	CALYPARSMCB	11/12/21 10:17:59.141 AM CET	11/12/2021	Official	Revised	☐		1,000,000.00				-1,100,000.00
					Value Date: 11/03/2021		Subtype: Revised							
51202	CLSB00000000167702	CNOW	CALYPARSMCB	11/3/21 2:54:23.787 PM CET	11/03/2021	Official	Revised	☐		100,000.00				110,000.00
51302	CLSB00000000167802	CNOW	CALYPARSMCB	11/3/21 3:46:59.954 PM CET	11/03/2021	Official	Revised	☐		1,000,000.00				-1,120,000.00
					Value Date: 10/18/2021		Subtype: Revised							
51102	CLSB00000000167602	CNOW	CALYPARSMCB	10/18/21 11:55:58.249 PM CEST	10/18/2021	Official	Revised	☐		2,000,000.00				2,200,000.00

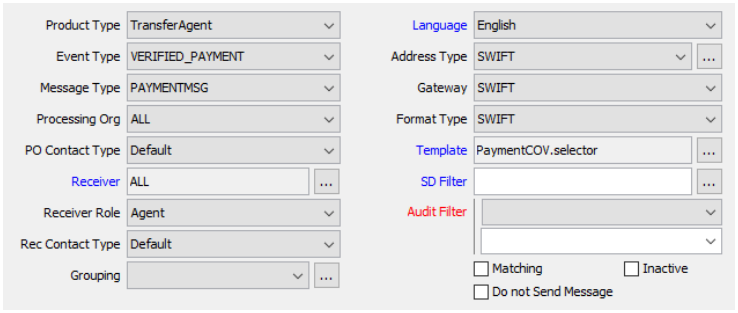


Trade Id	Trade Date	Trade Settle Date	Product Description	Trade Currency	CLS_PAY_TRADE
168004	Nov 12, 2021 10:17 AM	11/12/2021	TransferAgent(1,100,000.00 USD)	USD	true
168003	Nov 12, 2021 10:17 AM	11/12/2021	TransferAgent(1,000,000.00 EUR)	EUR	true





Message set up remains the same than for CLS settlement. MT202XferAgent will be generated thanks to:




```
{1:F01CALYUS33AXXX0000000000}{2:I202BONYFRPPXXXXN2020}{3:{108:MT202000016210}}{4:
:20:16210
:21:168004
:32A:211112USD1100000,
:57A:CLSBUS30XXX
:58A:/CALYPARSMCB ANY (AT)CLS
CALYUS33XXX
:72:/CLSNOW/2300
-}
{5:}
```

CLS member generates payment instruction to the Nostro agent with:

1- Beneficiary BIC Code + **CLSNow Account number**
CLSNow account Nb should be set in field A/C of the SDI configuration for CLSNow method
2- **CLSNow** Indicator to speed up the priority of this payment

In MT, field 72: include CLSTIME which is picked up from new currency attribute as described below

Even if it should be a bilateral agreement between the CLSNow participant and Nostro, best practice to be adopted is: 72:/CLSNOW/1630

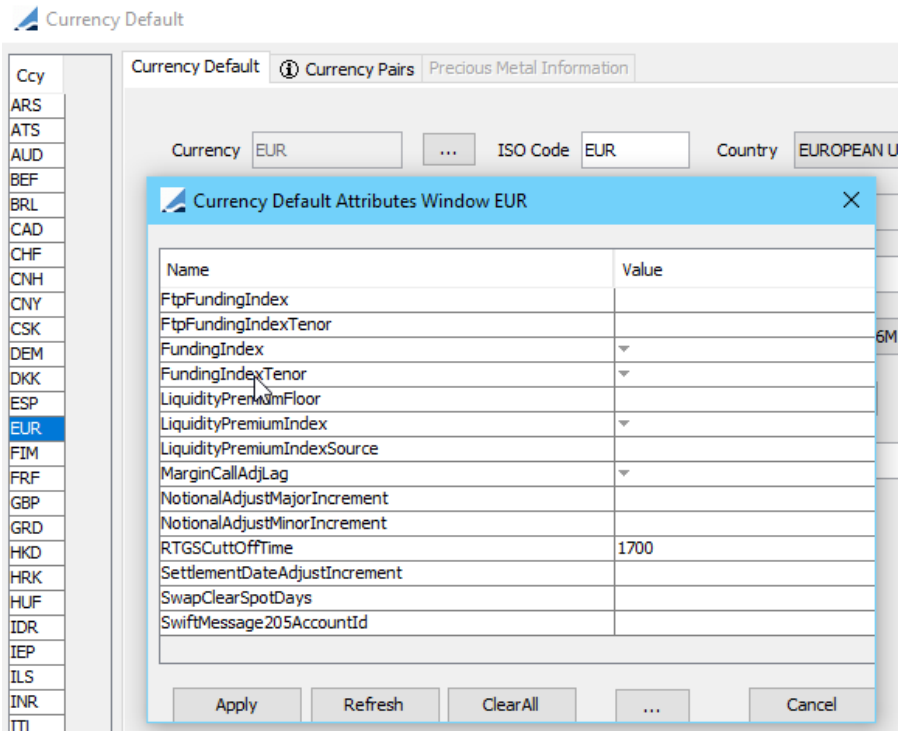
We create a new currency attribute: RTGSCuttOffTime

Value is the cut off time of the RTGS system (or the bilateral agreement between CLSNow member and the nostro) The hour has to be expressed in 24 hours format

The cutoff time for each currency should be set to 90 minutes prior to the relevant RTGS system close for payment time. We will provide the following default configuration:

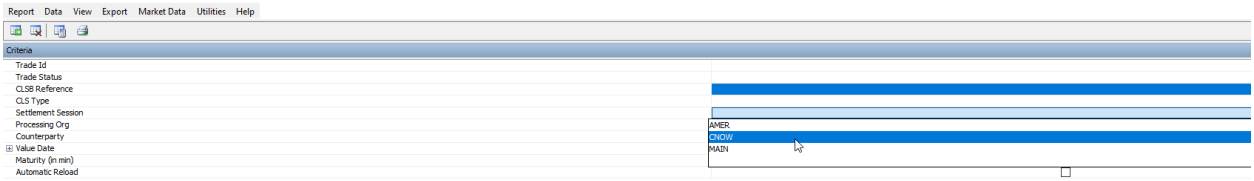
Currency	RTGSCuttOffTime
USD	2300
CAD	2230
GBP	1730
CHF	1530
EUR	1630

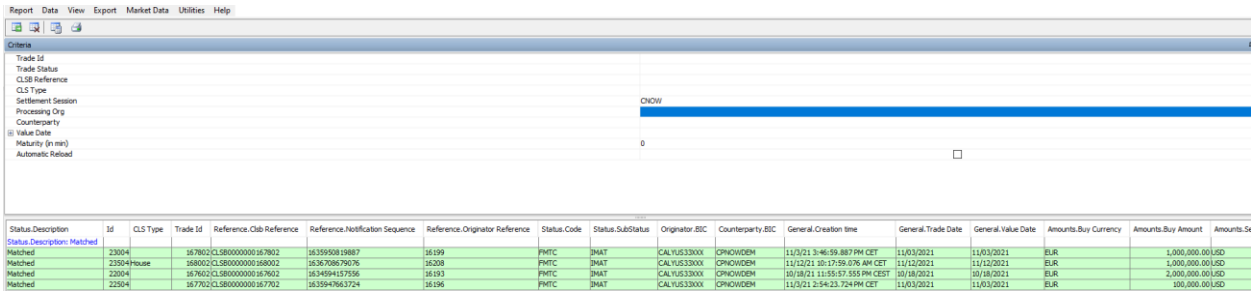
This new attribute is created while running the CLSSchemaData script



7.4 CLS Trade Info & Reports

CNOW added in Settlement Session





In the following reports:

1- CLSTradeInfo
2- CLS PayInSchedule
3- CLSMessage

CLS Resilience

8.1 Background

As part of the Convergence project, CLS introduced the concept of “Member Replay”, requesting that settlement members develop the capability to reconcile, re-book and replay instructions to CLSSettlement in the extreme, but plausible scenario that it would be needed to fall back to FXCore following the Convergence cut over.

CLS has established 2 recovery procedures in case of failure from which the system cannot be recovered through the application of standard and preferred fix-forward recovery methodology:

- Checkpoint Recovery: act of restoring the system to a previous point in time. The period from the time the checkpoint was taken to the time that the checkpoint is used to recover the service is described as the data loss window. During this window, any events processed by CLS will be lost and any resultant notifications sent to Members will be declared invalid.

  CLS will only invoke Checkpoint Recovery in the event that the data loss window does not span a period of settlement.

- Cold Start: In the event that Checkpoint Recovery cannot be used, CLS will complete recovery by invoking the Cold Start process. This process removes all transactional data (e.g., trade data) from the CLSSettlement system while retaining static data such as account information, user profiles or XML endpoint configuration

Both mechanisms result in data loss within CLS systems and as such Members will be required to reconcile their positions with the target restore point, re-submit lost Instructions and re-process the newly notified events.

The ability to invoke Checkpoint Recovery and Cold Start will be maintained as part of CLS’s on-going enhanced resilience capability.

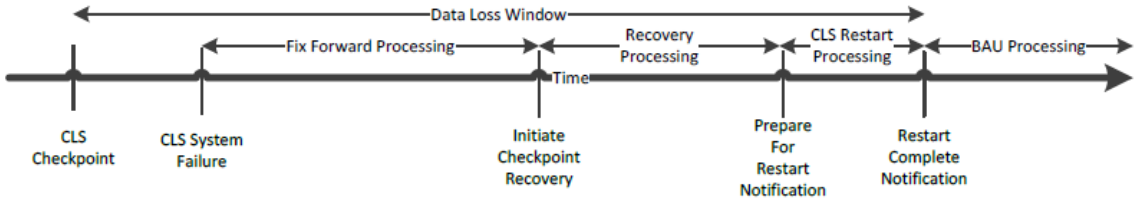
Both recovery procedures provide mechanisms for managing cyber events and they are in line with the mandated CPMI-IOSCO guidance

Those cases are supported but as they are exceptional events, the solution requires some manual reconciliation and actions.

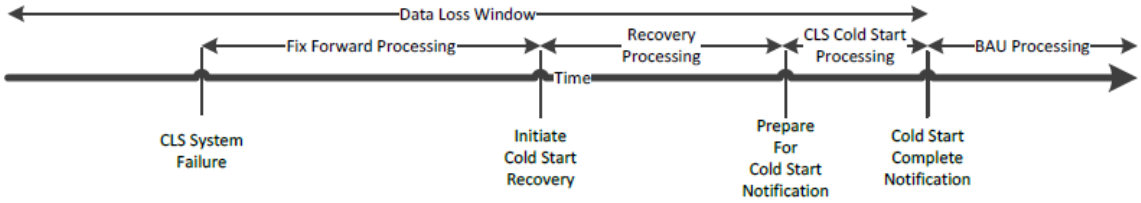
Out of Scope: Trade instructions submitted by the CLS Web UI.

8.2 Process Overview

Checkpoint Recovery



Cold Start Recovery



Main Points

- Notification from CLS to Member when system stops: Member stops sending messages to CLS
- Reconciliation phase: Member identifies messages to be resent and messages to be removed (=Undo actions)
- Notification from CLS to Member when system restarts: Member resends messages
- Member to reconcile with CLS Web UI

Main Points	Checkpoint Recovery	Cold Start Recovery	Calypso Process
Data Loss	Any trade instructions sent to CLS between the weekend service shutdown and the completion of the failback will be removed from CLS Systems and will need to be rebooked.	All trade instructions sent to CLS will be removed from the CLS Systems and will need to be rebooked.	NA
Notifications To prevent failure system	From CLS to Member: **PREPARE_FOR_RESTART** **RESTART_COMPLETE** From Member to CLS: **Acknowledged** of Restart Complete Notification	From CLS to Member: **PREPARE_FOR_COLD_START** **COLD_START_COMPLETE** From Member to CLS: **Acknowledged** of Restart Complete Notification	1- Integration of Notifications from CLS 2- Stop sending Messages 3- Reconciliation 4- Acknowledgment sent to CLS 5- Undo process Messages

Main Points	Checkpoint Recovery	Cold Start Recovery	Calypso Process
Queue Purging	Purged trade instructions during the CLS Restart Phase: FACC notifications from CLS to Member with reason code "Message rejected as part of CLS recovery activities"	Purged trade instructions during the CLS Restart Phase will not be notified to members.	Integration of Failed Acceptance - FACC
Replay of Trade Instructions	A small number of hours of trade instructions will need to be rebooked to CLS conscious that some re-formatting may be required.	Trade instructions going back many years may need to be rebooked to CLS. Care must be taken to ensure that trades are rebooked in a format that is supported by CLS, and also matches the current set of validation rules	Resubmit Messages
Use in Failback	Will be used for Failback of the CLSSettlement service to FXCore prior to SSBD on day 1 of Convergence service.	Will be used for Failback of CLSSettlement service after the SSBD on day 1 of Convergence service	

It is strongly recommenced to CLS members to use the CLS WEB UI to verify and ensure that member systems are correctly aligned with CLS settlement services as soon as the CLS Web UI is accessible.

8.3 Integration of Operational Messages from CLS

8.3.1 New Operational Messages

The following messages are imported.

Message Supported	Ack	Task Station Exception	Description
Prepare to Restart Prepare for Cold Start admi.004.002	NO	EX_CLS_PREPARE_FOR_RESTART EX_CLS_PREPARE_FOR_COLD_START	CSL System Failure PREPARE_FOR_RESTART CSL System Failure PREPARE_FOR_COLD_START
Restart Complete Cold Start Complete admi.004.002	**YES**	EX_CLS_RESTART_COMPLETE EX_CLS_COLD_START_COMPLETE	CSL System Failure RESTART_COMPLETE CSL System Failure COLD_START_COMPLETE

As part of the recovery process, messages are generated for each impacted service and delivered over the SWIFT MI (XML) and CLS Web UI (User Gateway) channels to inform Members which recovery process is being invoked and provide sufficient information to re-align their systems following the completion of the CLS recovery

processes. These messages are sent as structured System Event notifications, of the Operational Message type (MSGM), to each Member branch.

Some of these Operational Message notifications will require acknowledgment from each Member branch (refer to column Ack in table above).

Each notification is specific to a particular member branch. If a member only has one branch, the notifications will list all XML endpoint DN(s) and BIC(s) associated with that Member branch. If a member has multiple branches, then each notification will carry a subset of the Member's XML endpoint DN(s) and BIC(s).

This notification will carry the necessary details that will allow the Members to start to reconcile and resynchronize their internal systems to align to the CLS ‘point of recovery’

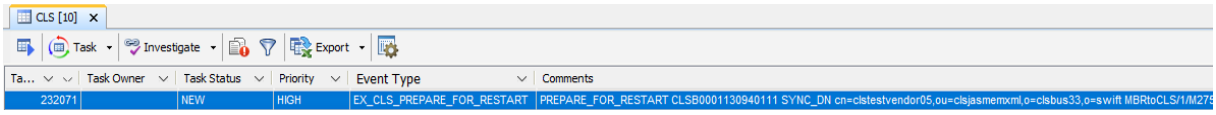
The admin.004 contains: 1 component for MI channel (in blue which is 1 MX message) and 1 component for FIN channel (in green represented by 2 MX messages)

All of those messages are linked by the same reference Id (here in red) in the example below. You can see there are 3 MX messages for the same reference Id

```
<?xml version="1.0" encoding="UTF-8" ?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:admi.004.001.02"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:iso:std:iso:20022:tech:xsd:admi.004.001.02 ../Schemas/admi.004.001.02.xsd">
<SysEvtNtfctn>
<EvtInf>
<EvtCd>MSGM</EvtCd>
<EvtParam>001</EvtParam>
<EvtParam>M201809141234600</EvtParam>
<EvtParam>MAIN</EvtParam>
<EvtParam>Operational Message</EvtParam>
<EvtParam>N</EvtParam>
<EvtParam>N/A</EvtParam>
<EvtParam>N/A</EvtParam>
<EvtDesc>PREPARE_FOR_RESTART M201809141234600 1 of 3
SYNC_DN cn=xyz,ou=abc,o=testgb40,o=swift
MbrToCLS/1/SENDERSREF111111
MbrToCLS/2/NULL
CLStoMBR/1/M201809141234567
CLStoMBR/2/M201809141234568
CLStoMBR/3/M201809141234555
SYNC_DN cn=xyz,ou=def,o=testgb40,o=swift
MbrToCLS/1/SENDERSREF222222
MbrToCLS/2/SENDERSREF333333
CLStoMBR/1/M201809141234561
CLStoMBR/2/M201809141234560
CLStoMBR/3/M201809141234562</EvtDesc>
<EvtTm>2018-09-14T15:30:22.978</EvtTm>
</EvtInf>
</SysEvtNtfctn>
</Document>
<?xml version="1.0" encoding="UTF-8" ?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:admi.004.001.02"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:iso:std:iso:20022:tech:xsd:admi.004.001.02 ../Schemas/admi.004.001.02.xsd">
<SysEvtNtfctn>
<EvtInf>
<EvtCd>MSGM</EvtCd>
<EvtParam>001</EvtParam>
<EvtParam>M201809141234601</EvtParam>
<EvtParam>MAIN</EvtParam>
<EvtParam>Operational Message</EvtParam>
<EvtParam>N</EvtParam>
<EvtParam>N/A</EvtParam>
<EvtParam>N/A</EvtParam>
<EvtDesc>PREPARE_FOR_RESTART M201809141234600 2 of 3
SYNC_BIC TESTGB40XXX
SENDERSREF123456/Y
SYNC_BIC TESTGB40ABC
SENDERSREF246802/N
SYNC_BIC TESTGB40DEF
SENDERSREF135791/Y</EvtDesc>
<EvtTm>2018-09-14T15:30:23.978</EvtTm>
</EvtInf>
</SysEvtNtfctn>
</Document>

<?xml version="1.0" encoding="UTF-8" ?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:admi.004.001.02"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:iso:std:iso:20022:tech:xsd:admi.004.001.02 ../Schemas/admi.004.001.02.xsd">
<SysEvtNtfctn>
<EvtInf>
<EvtCd>MSGM</EvtCd>
<EvtParam>001</EvtParam>
<EvtParam>M201809141234602</EvtParam>
<EvtParam>MAIN</EvtParam>
<EvtParam>Operational Message</EvtParam>
<EvtParam>N</EvtParam>
<EvtParam>N/A</EvtParam>
<EvtParam>N/A</EvtParam>
<EvtDesc>PREPARE_FOR_RESTART M201809141234600 3 of 3
SYNC_BIC TESTGB40GHI
SENDERSREF223344/Y
SYNC_BIC TESTGB40JKL
SENDERSREF334455/N
SYNC_BIC TESTGB40MNO
SENDERSREF445566/Y</EvtDesc>
<EvtTm>2018-09-14T15:30:23.978</EvtTm>
</EvtInf>
</SysEvtNtfctn>
</Document>
```

For **each Notification, an exception is created** to facilitate the monitoring through the Task Station.

For the “Complete” Notification, the system will send automatically to CLS an acknowledgment as soon as the message is integrated into the system.



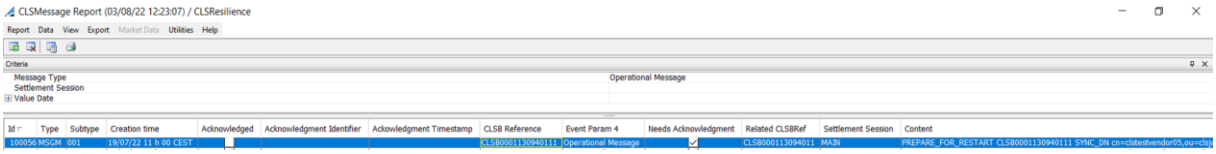
CBPR+ SR 2023

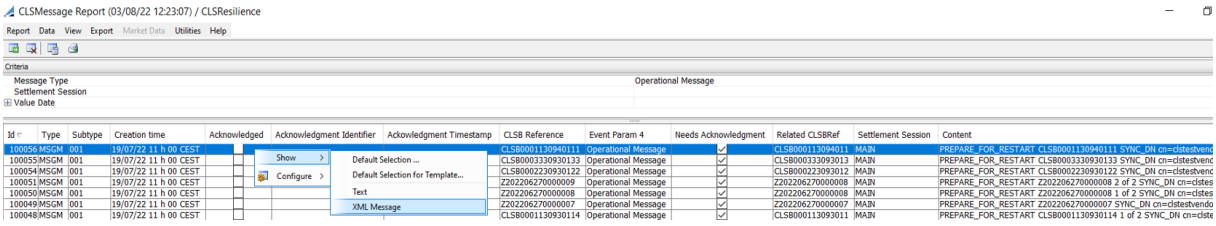
If domain “USE_SR_2023” contains Value = TRUE and domain "MXUseCBPR2023" contains Value = TRUE, the following changes apply to admi.004.001.02 integration:

/Document/admi.004.001.01/EvtInf changes to /Document/SysEvtNtfctn/EvtInf

8.3.2 CLSMessage Report

admin.004 messages are displayed in the CLSMessage Report with Message type= Operational Message.





In ‘Text’, the system displays the content of the <EvtDesc>ABCDEFGHIJKALMOPQRSTUVWXZY</EvtDesc> of the event information.

In XML Message, you can see the XML message.

Columns description:

- <EvtParam> is visible under CLSBRef column
- The description of the event: is visible under Content column
- The reference under <EvtDesc> represents the common reference of all related event messages and is visible under Related CLSBRef column

For CLStoMBR syncpoints (i.e. messages sent from CLS to the Member) the 16-byte CLSB Message reference (e.g. M201808081234567) of the last message sent from CLS to the Member is recorded.

8.3.3 PREPARE_FOR_RESTART Notification

admi.004.002 XML message is sent from CLS to member. It has information necessary to allow members to reconcile their system with CLS.

The 'Prepare for restart' notification will be split into two component parts, the first will detail the SWIFT MI channel recovery data, and the second will detail the SWIFT FIN recovery data. These and all subsequent recovery notifications will have a common CLSB reference value in the header row of the <EvtDesc> field of the admi.004.002 XML message which can be used by members to link all recovery related messages together. The individual notifications will have their own CLSB reference listed in the second <EvtParam> field of the admi.004.002 XML message.

Following the CLSB reference in the header row of the <EvtDesc> field there will be additional data showing how many messages need to be concatenated to get the full set of recovery data. This will take the form of 'message_number of total_number_of_messages'. Given that CLS will split the SWIFT MI channel and SWIFT FIN channel recovery information into separate admi.004.002 notifications there will always be at least 2 'Prepare for restart' notifications.

- As soon as this message is integrated into Calypso, the system will stop sending STP the messages to CLS – process description below
- EX_PREPARE_FOR_RESTART Exception created

8.3.4 RESTART_COMPLETE Notification

admi.004.002 XML message should be issued by CLS at about 30min after the PREPARE_FOR_RESTART notification.

The 'RESTART_COMPLETE' recovery notifications will have the same common CLSB Reference value as the 'PREPARE_FOR_RESTART' notifications in the header

```
<EvtDesc>RESTART_COMPLETE M201809141234600 1 of 3
SYNC_BIC_COMPLETE TESTGB40XXX</EvtDesc>
<EvtTm>2019-07-14T16:00:22.978</EvtTm>
```

The 'RESTART_COMPLETE' notification is composed by:

- A description of the event: <EvtDesc> RESTART_COMPLETE M201809141234600

  PREPARE_FOR_RESTART event

  CLSB Reference (common to all recovery messages, here corresponding to the PREPARE_TO_RESTART notification)
- The Checkpoint Recovery: Each BIC (11 characters) which is impacted by the Checkpoint Recovery preceding by 'SYNC_BIC_COMPLETE'

  For example:

SYNC_BIC_COMPLETE CALYFRPPXXX

- As soon as this message is integrated into Calypso, the system will automatically generate an ACK message
- Create EX_RESTART_COMPLETE Exception
- User to Remove MANUALLY true value on domain value CLSFailed to come back on the STP on messages - process description below

8.3.5 PREPARE_FOR_COLD_START Notification

Message for Cold Start has same structure than PREPARE_FOR_RESTART: split into two component parts, the first will detail the SWIFT MI channel DNs that were impacted by the Cold Start, and the second will detail the SWIFT FIN BICs that were impacted by the Cold Start. These and the subsequent 'COLD_START_COMPLETE' recovery notifications will have a common CLSB Reference value in the header which can be used by Members to link all recovery related messages together. The individual notifications will have their own CLSB Reference listed in the second <EvtParam> field of the admi.004 XML message.

```
<EvtDesc>PREPARE_FOR_COLD_START M201809141234610 2 of 2
COLD_BIC CALYFRPPXXX
</EvtDesc>
<EvtTm>2019-07-14T16:30:23.978</EvtTm>
```

SWIFT FIN Channel

The notification is composed by:

- A description of the event: <EvtDesc>PREPARE_COLD_START M201809141234600

  PREPARE_FOR_ COLD_START'event

  CLSB Reference (common to all recovery messages)

- The Checkpoint Recovery:

  Each BIC (11 characters) which is impacted by the Checkpoint Recovery preceding by 'COLD_BIC'

  Sync point information for the BIC: the reference which is the Sender's reference taken from field 20 of the last SWIFT FIN message

  - As soon as this message is integrated into Calypso, the system will stop sending STP the messages to CLS
  - EX_PREPARE_FOR_COLD_START Exception created

8.3.6 COLD_START_COMPLETE Notification

Message for COLD_START_COMPLETE has same structure as RESTART_COMPLETE

- As soon as this message is integrated into Calypso, the system will automatically generate an ACK message
- Create EX_COLD_START_COMPLETE Exception

» User to Remove MANUALLY true value on domain value CLSFailed to come back on the STP on messages – process described below

8.4 Integration of Failed Acceptance - FACC

During CLS's restart processing, it is likely that there will be unprocessed trade instructions held both within SWIFT (MI and FIN channels) and the internal CLS queues. CLS will temporarily open its SWIFT MI and SWIFT FIN inbound trade instruction channels and all unprocessed trade instructions will be discarded and result in a Failed Acceptance (FACC) notification being generated.4 The FACC notifications will have a new reason code of '**Message rejected as part of CLS recovery activities**'.

FACC notifications will not be sent to the Members until CLS has issued the 'RESTART_COMPLETE' notification.

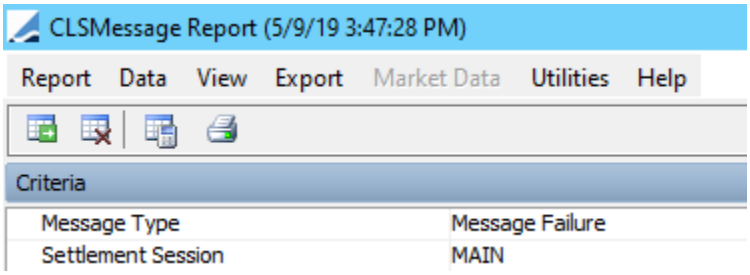
XML	<Rsn> <RjctgPtyRsn> Column 'Subtype'	<Rsn> <RsnDesc> Column 'Content'
Admi.002.001.001	FACC-RES	Message rejected as part of CLS recovery activities

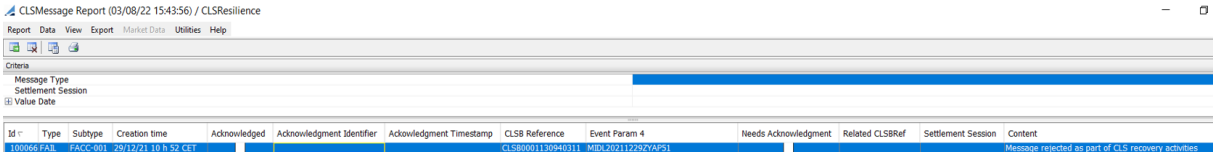
```
<admi.002.001.01>
      <RltdRef>
            <Ref>X202010110000001</Ref>
      </RltdRef>
      <Rsn>
            <RjctgPtyRsn>FACC-001</RjctgPtyRsn>
            <RjctnDtTm>2020-10-11T12:25:59</RjctnDtTm>
            <RsnDesc>Message rejected as part of CLS recovery
activities</RsnDesc>
```

In CLSMessage Report, those messages will be displayed under Message Type = Message Failure:

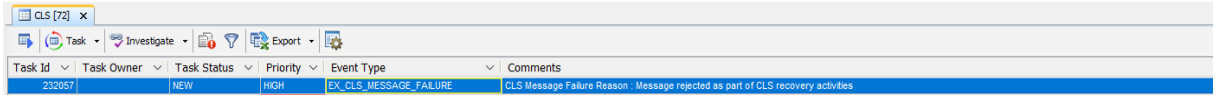
`<RsnDesc>Message rejected as part of CLS recovery activities</RsnDesc> is displayed the 'Content' column of the report`

<RltRef> is already stored under CLSB Reference





Exception created:



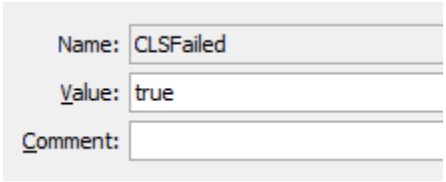
8.5 Stop Sending Messages

As soon as member receives the notification from CLS PREPARE_FOR_RESTART or PREPARE_FOR_COLD_START, Calypso must stop sending messages to CLS.

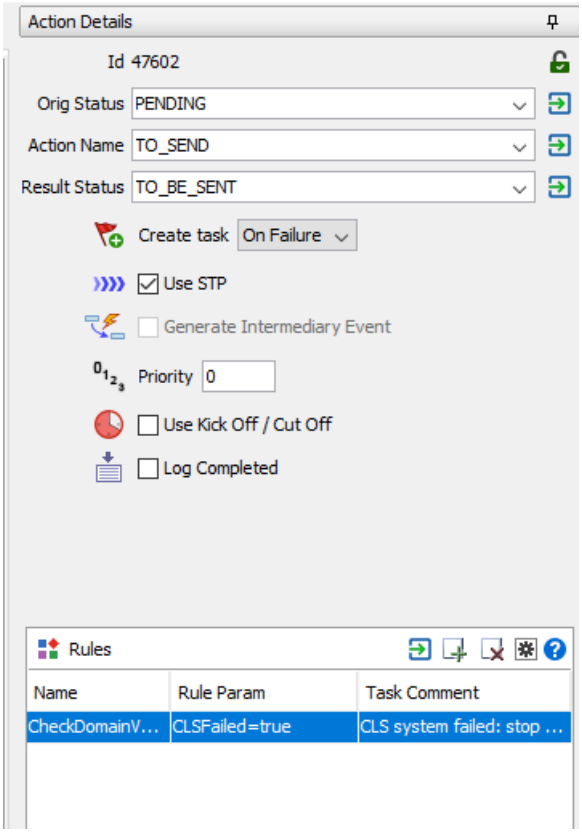
We recommend setting a STP transition before status taken by Sender Engine and remove STP when member is informed by CLS of the failure.

Message WF Rule ‘CheckDomainValue’ with Rule Param 'the name of the DV = value'. For example: CLSFailed=true

As soon as Notification PREPARE_FOR_RESTART or PREPARE_FOR_COLD_START from CLS is integrated into Calypso, the system will update the Domain value CLSFailed with value true.



Based on the Message WF, when failure happened, the messages to be sent to CLS will move to status 'PENDING', Tasks are raised into Task Station.



As soon as CLS system is restarted, CLS sends message RESTART_COMPLETE or COLD_START_COMPLETE, we recommend that a system user will remove manually the value true in CLSFailed DV (when he is ready to 'restart' CLS process)

Then User will be able to reprocess manually the messages and STP process could resume.

All messages will be stopped in PENDING status: Tasks will be raised

When CLS system is restarted, user will have to manually process the tasks and move the messages to TO_BE_SENT

» The system will not automatically remove the value in the Domain Value CLSFailed used to stop sending the messages to CLS . Removing true MUST be done MANUALLY by User

8.6 Reconciliation

This is a manual reconciliation:

User needs to identify all messages that may be impacted during the Data Loss Window.

Based on CLS Report of Operation Message, User will have to identify:

- For messages sent from Calypso to CLS: Tag20 (BOMsg Id by default)

- For messages sent from CLS to Calypso (CLSB reference)

From BOMessage Report:

When receiving the PREPARE_FOR_RESTART or PREPARE_FOR_COLD_START Notification, he identified the recovery point.

From a message report, all messages sent to CLS and received from CLS are displayed. sorted by datetime.

User will look for CLS_REF msg attribute and Msg Id identified as recovery point.

He can already have an idea of the messages lost during the data loss window, and then prepare to replay those messages when CLS system restarts:

- Messages sent to CLS which have to be resent → refer to section 8.8
- Messages received from CLS which have to be removed → refer to related Acknowledgment sent to CLS

The CLS Message engine can already generate the acknowledgement to CLS after the reception of some messages:

Pay-in calls (camt.061.001.02), Pay-in Schedule (camt.062.001.03) and System Event Operational Messages (admi.004.001.01).

The system will send an acknowledgement message to CLS when receiving these following notifications:

- RESTART_COMPLETE
- COLD_START_COMPLETE

» The system will generate Acknowledge messages automatically, using standard admi.011 message when receiving admi.004.001.02 messages (even if it is not required for PREPARE_FOR_RESTART and PREPARE_FOR_COLD)

8.7 Revert CLSTradeInfo Status

8.7.1 Process

This process is manual.

Our recommendation is:

- During reconciliation phase, user needs to prepare “the differences” between the position in Calypso system and the CLS position based on the information he received from CLS. For that, tools to use are the message report and CLS Trade info report.

- While CLS system is back, user needs to double check that his manual reconciliation is aligned with the production. In all cases, the CLS Web UI should be used as point of reference to aid any reconciliation processing.
- While user knows what to perform, first, align CLS TradeInfo to same status than CLS
- Then resubmit the messages that were lost during the failure

For the CLS TradeInfo status you can apply the following reserved actions: **REVERT** and **UNDO_REVERT** (in case a REVERT action was performed by mistake)

Those 2 actions will be visible and then can be performed manually only when CLSFailed is set to true and they are used to synchronize the status of the message in calypso system and CLS system

REVERT: This action will "move back" to a previous version of the CLS trade info, meaning that when user applies this action on the CLSTradeInfo, the object will create a new version of the object with information of a previous status selected.

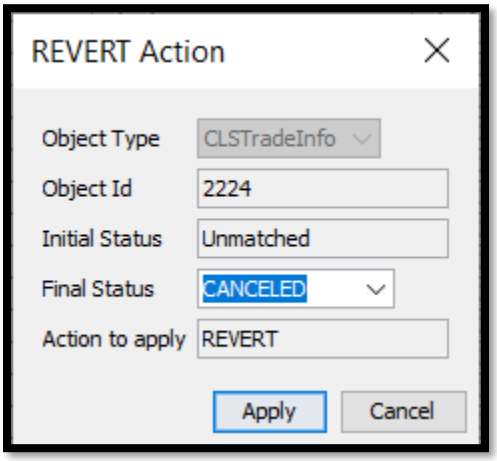
In the case there are several version of the object for same status, the system will select the last version.

For example:

Class Name	Id	Name	Field Name	Date	User Name	Old Value	New Value	Action	Field Type	LE	PO	Type	Version
CLSTradeInfo	18504	CLSTradeIn	_CREATE_	12/27/19 11:23:37.393 AM CET	calypso_user			NONE	NONE			CLSTradeInfo	0
CLSTradeInfo	18504	CLSTradeIn	_status	12/27/19 12:06:25.682 PM CET	calypso_user	UNMATCHED	MATCHED	NONE	com.calypso.tk.bo.TradeStatus			CLSTradeInfo	2
CLSTradeInfo	18504	CLSTradeIn	matchedSid	12/27/19 12:06:25.682 PM CET	calypso_user		CLSB0000000165901S	NONE	java.lang.String			CLSTradeInfo	2
CLSTradeInfo	18504	CLSTradeIn	matchingRe	12/27/19 12:06:25.682 PM CET	calypso_user		CLSB0000000165901M	NONE	java.lang.String			CLSTradeInfo	2
CLSTradeInfo	18504	CLSTradeIn	notificatior	12/27/19 12:06:25.682 PM CET	calypso_user	1.57744E+12	1.57744E+12	NONE	com.calypso.tk.bo.CLSNotificationNumber			CLSTradeInfo	2
CLSTradeInfo	18504	CLSTradeIn	timestamp	12/27/19 12:06:25.682 PM CET	calypso_user	12/27/2019 11:23	12/27/2019 12:06	NONE	com.calypso.tk.core.JDatetime			CLSTradeInfo	2
CLSTradeInfo	18504	CLSTradeIn	timestamp	12/27/19 12:15:04.610 PM CET	calypso_user	12/27/2019 12:06	12/27/2019 12:15	NONE	com.calypso.tk.core.JDatetime			CLSTradeInfo	3
CLSTradeInfo	18504	CLSTradeIn	_status	12/27/19 6:59:28.468 PM CET	calypso_user	MATCHED	SETTLED	NONE	com.calypso.tk.bo.TradeStatus			CLSTradeInfo	4
CLSTradeInfo	18504	CLSTradeIn	notificatior	12/27/19 6:59:28.468 PM CET	calypso_user	1.57744E+12	1.57747E+12	NONE	com.calypso.tk.bo.CLSNotificationNumber			CLSTradeInfo	4
CLSTradeInfo	18504	CLSTradeIn	timestamp	12/27/19 6:59:28.468 PM CET	calypso_user	12/27/2019 12:06	12/27/2019 18:59	NONE	com.calypso.tk.core.JDatetime			CLSTradeInfo	4

If MATCHED status is selected, the version of the object should be version 3.

You can also select the Canceled status on the REVERT action for Unmatched and . Unmatched Alleged.



UNDO_REVERT: This action will “undo” the previous REVERT action

For example, if the CLSTradeInfo was in UNMATCHED, MISMATCHED, MATCHED, SETTLED, CANCELED status, user reverts to MATCHED status but in fact user should have moved back to MISMATCHED status.

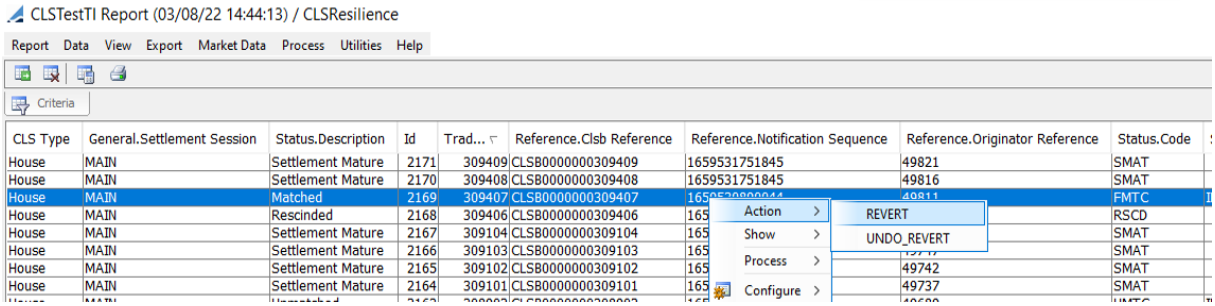
To do that, you first need to UNDO the previous Revert action to come back to SETTLED status and then REVERT to MISMATCHED status.

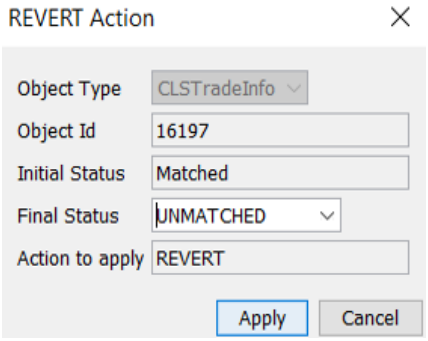
Audit entries are recorded at every step.

From CLSTradeInfo Report, load all objects, display CLSB Ref and Message Id, sort by DateTime

User will have to select the object(s), right click and do the action REVERT to move back to a previous status

When selecting 1 or several CLS Trade info, right click and select REVERT action, a new popup window will be displayed:





The following fields cannot be edited:

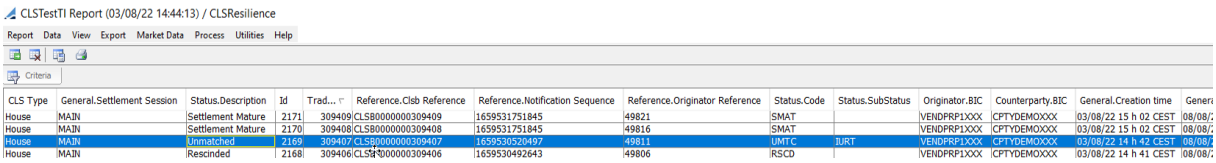
- Object type: CLSTradeInfo
- Object Id: Object Id or “many” if multiple object are selected
- Initial status: Initial status of the selected object selected
- Action to apply: REVERT

User can only select:

- Final Status: The status to revert to – It needs to be a previous status

  For examples if the CLSTradeInfo was in UNMATCHED, MATCHED, SETTLED, in final status, user won't be able to select MISMATCHED

Result:



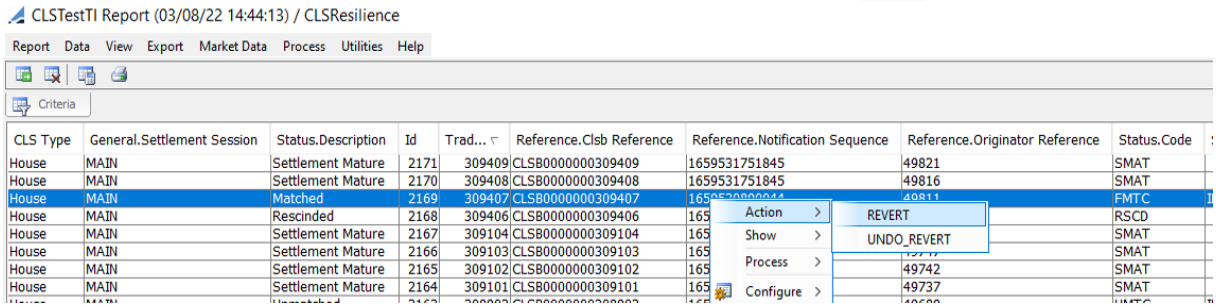
8.7.2 Sample Scenarios

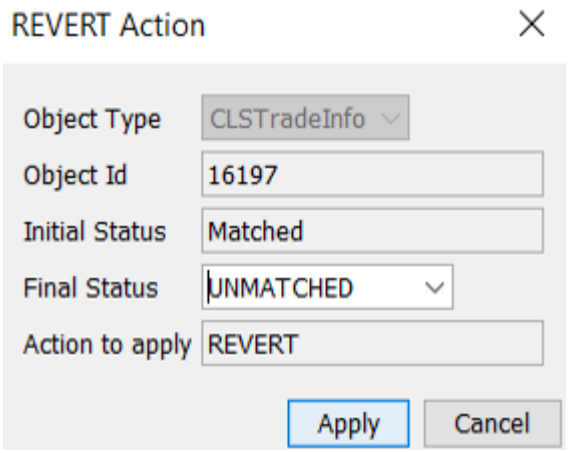
Reverting a Single Object

From CLSTradeInfo Report, load all objects, display CLSB Ref and Message Id, sort by DateTime.

User will have to select the object(s), right click and do the action REVERT to move back to a previous status.

When selecting 1 CLS Trade info, right click and select REVERT action, a popup window will be displayed:

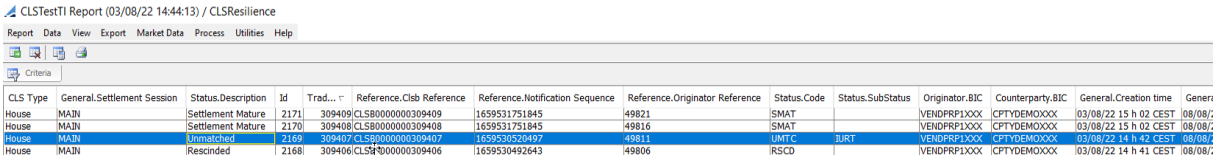




- Select the Final Status: The status to revert to – It needs to be a previous status

For examples if the CLSTradeInfo was previously in UNMATCHED, MATCHED, SETTLED status, you can only select UNMATCHED, MATCHED, SETTLED – You cannot select MISMATCHED.

Result:



Reverting Multiple Objects

Using CLSTradeInfo Window

The process of executing a REVERT/UNDO_REVERT on multiple objects is the same as that of a single CLSTradeInfo object. However, the Final Status will provide all the status that the individual objects went through.

For example:

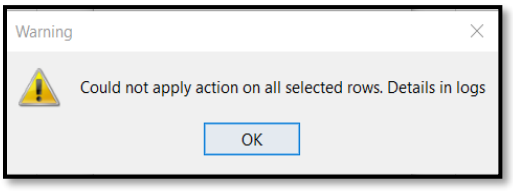
- CLSTradeInfo 1 was previously in UNMATCHED, MATCHED, SETTLED status
- CLSTradeInfo2 was previously in UNMATCHED, MATCHED, SETTLEMENT MATURE, SETTLED status

The final status provides the following options:

- UNMATCHED
- MATCHED
- SETTLEMENT MATURE
- SETTLED

If UNMATCHED or MATCHED is selected, then both CLSTradeInfo objects would be reverted.

If SETTLEMENT MATURE is selected, then CLSTradeInfo 2 will be reverted, and the following message will be displayed for CLSTradeInfo 1.



The navigator logs will provide users with the list of CLSTradeInfo objects where action could not be applied.

If applying action to more than 25 CLSTradeInfo objects, then it is recommended to use the scheduled task CLSTRADEINFO_ACTION instead. See Applying Bulk Action for details.

8.8 Resubmit Messages

As mentioned in the previous section, while the CLS system is restarted, the user will have to double check that the pre-reconciliation was prepared during the recovery.

STP should be reestablished manually.

The member needs to replay any trade instructions to CLS that have been lost through the 'Data loss window'

Action "RESUBMIT_NEW" or "RESUBMIT" needs to be applied to messages that have been lost.

- RESUBMIT_NEW – creates a copy of the message and sets the fields Msg Link ID = 0 and SUB_ACTION = NONE.
- RESUBMIT – creates an exact copy of the message.

Before processing actions, user should first look at CLSTradeInfo and check if the status is the same between calypso and CLS.

From Message report, upload all MT304 or fxtr messages sent to CLS and display the message attribute CLS_REF and the BOMessage Id. Sort by DateTime.

User will have to select the BO Message and apply the necessary action, "RESUBMIT_NEW" or "RESUBMIT", to reconcile Calypso with CLS status.

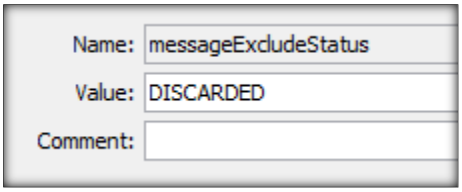
8.8.1 Configuration Advice

Domain

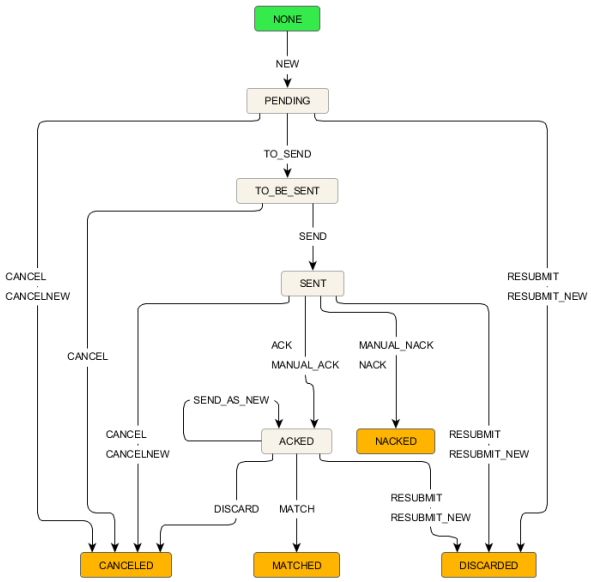
The message status 'DISCARDED' indicates trade instructions that have been lost/unprocessed due to CLS failure.

As DISCARDED messages are unprocessed it is important for the message engine to also stop the processing and inclusion of such messages during amendments or cancelations.

The domain 'messageExcludeStatus' should be updated with the value 'DISCARDED'.



Workflow



8.8.2 Sample Scenarios

Case 1: Single Message Failure with No Record Retention in CLS (SUB_ACTION = NEW)

Before Check Point Recovery:

Calypso Status: Trade (CLSTradeInfo) is Unmatched

CLS Status: CLS ref set and status Unmatched

Message Activity:

New Trade Confirmation SENT & ACKED (Template can be MT304 or fxtr)

MSG ID	MSG Link ID	MSG TYPE	MSG STATUS	SUB_ACTION	TEMPLATE NAME
1	0	CLSCONFIRM	ACKED	NONE	MT304

During recovery process:

Calypso Status: Trade (CLSTradeInfo) is Unmatched

CLS Status: No Record

Expected Result:

We need to align the system with CLS recording during the recovery process.

As there is no record for that trade on CLS, user can either “**RESUBMIT_NEW**” or “**RESUBMIT**” the same trade confirmation message. Both actions would yield the same result.

While applying RESUBMIT/RESUBMIT_NEW action:

- Select the BOMessage that needs to be resent.
- A New BOMessage will be created with:

  Same details as MT304 message

  Msg Linked Id= 0

  SUB_ACTION = NONE
- Previous BOMessage will move to DISCARDED

MSG ID	MSG Link ID	MSG TYPE	MSG STATUS	SUB_ACTION	TEMPLATE NAME
1	0	CLSCONFIRM	DISCARDED	NONE	MT304
2	0	CLSCONFIRM	ACKED	NONE	MT304

Case 2: Single Message Failure with Record Retention in CLS (SUB_ACTION = AMEND)

Before Check Point Recovery:

Calypso Status: Trade (CLSTradeInfo) is Matched

CLS Status: CLS ref set and status Matched

Message Activity:

New Trade Confirmation SENT & ACKED

AMEND Trade Confirmation SENT

MSG ID	MSG Link ID	MSG TYPE	MSG STATUS	SUB_ACTION	TEMPLATE NAME
1	0	CLSCONFIRM	ACKED	NONE	MT304
2	1	CLSCONFIRM	SENT	AMEND	MT304

During recovery process:

Calypso Status: Trade (CLSTradeInfo) is Matched

CLS Status: CLS ref set and status Unmatched

Expected Result:

We need to align the system with CLS recording during the recovery process.

As there is a record in CLS, user needs to "RESUBMIT" the most recent trade confirmation details. This is given in the AMEND message.

While applying RESUBMIT action:

- Select the 'AMEND' BOMessage that needs to be resent.
- A BOMessage will be created with:

  Same details as the 'AMEND' MT304 message

  Msg Linked Id= 1

  SUB_ACTION = AMEND
- Previous BOMessage will move to DISCARDED

MSG ID	MSG Link ID	MSG TYPE	MSG STATUS	SUB_ACTION	TEMPLATE NAME
1	0	CLSCONFIRM	ACKED	NONE	MT304
2	1	CLSCONFIRM	DISCARDED	AMEND	MT304
3	1	CLSCONFIRM	ACKED	AMEND	MT304

Case 3: Multiple Message Failure with No Record Retention in CLS

Before Check Point Recovery:

Calypso Status: Trade (CLSTradeInfo) is Matched

CLS Status: CLS ref set and status Matched

Message Activity:

New Trade Confirmation SENT & ACKED

AMEND Trade Confirmation SENT & ACKED

MSG ID	MSG Link ID	MSG TYPE	MSG STATUS	SUB_ACTION	TEMPLATE NAME
1	0	CLSCONFIRM	ACKED	NONE	MT304
2	1	CLSCONFIRM	ACKED	AMEND	MT304

During recovery process:

Calypso Status: Trade (CLSTradeInfo) is Matched

CLS Status: No Record

Expected Result:

We need to align the system with CLS recording during the recovery process.

As there is no record in CLS, user needs to "**RESUBMIT_NEW**" the most recent trade confirmation details. This is given in the AMEND message.

While applying RESUBMIT_NEW action:

- Select the 'AMEND' BOMessage that needs to be resent.
- A BOMessage will be created with:

  Same details as the 'AMEND' MT304 message

  Msg Linked Id= 0

  SUB_ACTION = NONE
- Previous BOMessage will move to DISCARDED

[NOTE: CLS records indicate that message 1 (New MT304) was never received therefore users should manually apply the action DISCARD to the message.]

MSG ID	MSG Link ID	MSG TYPE	MSG STATUS	SUB_ACTION	TEMPLATE NAME
1	0	CLSCONFIRM	CANCELED	NONE	MT304
2	1	CLSCONFIRM	DISCARDED	AMEND	MT304
3	0	CLSCONFIRM	ACKED	NONE	MT304

Case 4: Multiple Message Failure with Record Retention in CLS

Before Check Point Recovery:

Calypso Status: Trade (CLSTradeInfo) is Matched

CLS Status: CLS ref set and status Matched

Message Activity:

New Trade Confirmation SENT & ACKED

AMEND Trade Confirmation SENT & ACKED

Domain 'CLSFailed' → True

AMEND Trade Confirmation PENDING

AMEND Trade Confirmation PENDING

MSG ID	MSG Link ID	MSG TYPE	MSG STATUS	SUB_ACTION	TEMPLATE NAME
1	0	CLSCONFIRM	ACKED	NONE	MT304

MSG ID	MSG Link ID	MSG TYPE	MSG STATUS	SUB_ACTION	TEMPLATE NAME
2	1	CLSCONFIRM	ACKED	AMEND	MT304
3	2	CLSCONFIRM	PENDING	AMEND	MT304
4	3	CLSCONFIRM	PENDING	AMEND	MT304

During recovery process:

Calypso Status: Trade (CLSTradeInfo) is Matched

CLS Status: CLS ref set and status Unmatched

Expected Result:

We need to align the system with CLS recording during the recovery process.

According to CLS records the first message was received, however several AMEND messages have failed or remained in PENDING status due to CLS failure (domain 'CLS FAILED = True).

In this scenario users should apply the following actions:

- Message 1 → "SEND_AS_NEW" with SUB_ACTION 'CANCEL' to remove trade confirmation from CLS.
  Message 5 created which is a copy of message 1 (MT 304) with SUB_ACTION = CANCEL
- Message 2 → manual DISCARD
- Message 3 → manual DISCARD
- Message 4 → "RESUBMIT_NEW" as this message contains the most recent trade confirmation details.
  Message 6 created with Msg Link ID = 0 and SUB_ACTION = NONE

MSG ID	MSG Link ID	MSG TYPE	MSG STATUS	SUB_ACTION	TEMPLATE NAME
1	0	CLSCONFIRM	ACKED	NONE	MT304
2	1	CLSCONFIRM	CANCELED	AMEND	MT304
3	2	CLSCONFIRM	CANCELED	AMEND	MT304
4	3	CLSCONFIRM	DISCARDED	AMEND	MT304
5	0	CLSCONFIRM	ACKED	CANCEL	MT304
6	0	CLSCONFIRM	ACKED	NONE	MT304

[NOTE: The action SEND_AS_NEW allows users to create a copy of the message, specify the recipients and related sub action. To cancel the initial confirmation at CLS we use the sub action CANCEL.]

Case 5: Failure to Remove Record from CLS

Before Check Point Recovery:

Calypso Status: Trade (CLSTradeInfo) is Canceled

CLS Status: CLS ref set and status Canceled

Message Activity:

New Trade Confirmation SENT & ACKED

AMEND Trade Confirmation SENT & ACKED

CANCEL Trade Confirmation SENT

MSG ID	MSG Link ID	MSG TYPE	MSG STATUS	SUB_ACTION	TEMPLATE NAME
1	0	CLSCONFIRM	ACKED	NONE	MT304
2	1	CLSCONFIRM	ACKED	AMEND	MT304
3	2	CLSCONFIRM	SENT	CANCEL	MT304

During recovery process:

Calypso Status: Trade (CLSTradeInfo) is Canceled

CLS Status: Trade is Unmatched

Expected Result:

We need to align the system with CLS recording during the recovery process

Previously the trade was Unmatched and an error on PO side has been rectified, Trade has been Canceled in Calypso and 'CANCEL' message has been generated but CLS did not process it so we need to resubmit the 'CANCEL' message

While applying RESUBMIT action:

- Select the BOMessage that needs to be resent.
- A New BOMessage will be created with:

  Same details as MT304 message

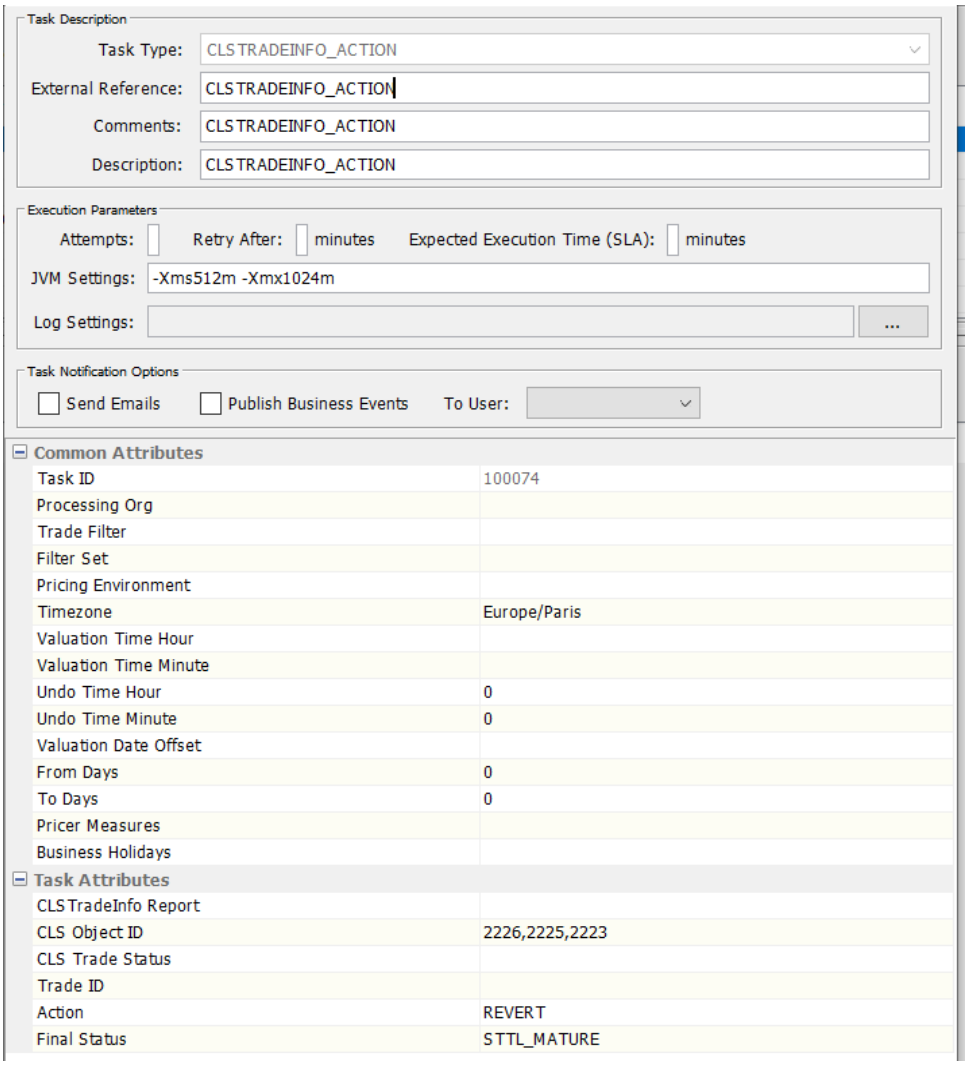
  Msg Linked Id = 2

  SUB_ACTION = CANCEL
- Previous BOMessage will move to DISCARDED

MSG ID	MSG Link ID	MSG TYPE	MSG STATUS	SUB_ACTION	TEMPLATE NAME
1	0	CLSCONFIRM	ACKED	NONE	MT304
2	1	CLSCONFIRM	ACKED	AMEND	MT304
3	2	CLSCONFIRM	DISCARDED	CANCEL	MT304
4	2	CLSCONFIRM	ACKED	CANCEL	MT304

8.9 Applying Bulk Action

You can use the scheduled task CLSTRADEINFO_ACTION to apply bulk actions based on a CLS Trade Info report template or Trade Info object IDs.



Task Attributes

- CLSTradeInfo Report: Select a saved CLSTradeInfo template
- CLS Object ID: Enter a CLSTradeInfo object ID. Multiple values can be entered and separated by a comma.
- CLS Trade Status: Enter a CLSTradeInfo status. Multiple values can be entered and separated by a comma.
- Trade ID: Enter a CLSTradeInfo trade ID. Multiple values can be entered and separated by a comma.
- Action: Select either REVERT or UNDO_REVERT
- Final Status:
  - If Action = UNDO_REVERT, then the final status will not be available and will be automatically set by the system
  - If Action = REVERT, then a final status can be selected. The list of available values is defined in the domain 'clsTradeInfoFinalStatuses'

Example

After proper scheduled task config the following objects and audit info are identified:

- CLSTradeInfo 1 was in UNMATCHED/ MATCHED/ SETTLED
- CLSTradeInfo 2 was in UNMATCHED/ MATCHED/ SETTLEMENT MATURE/ SETTLED

Case 1: Assume scheduled task Action = REVERT and scheduled task Final Status = SPLIT

- Result: The scheduled task is finished with errors, and both CLSTradeInfo objects would be shown in the scheduled task logs.

Case 2: Assume scheduled task Action = REVERT and scheduled task Final Status = MATCHED.

- Result: The scheduled task is successful with no errors, and both objects are reverted to MATCHED status.