Nasdaq Calypso

CA Indeval Integration Guide

Version 18

Revision 1.0
February 2024
Approved

Document History

Revision	Published	Summary of Changes
1.0	February 2024	First revision for version 18

This document guides you through the setup and integration for Corporate Actions for Mexican custody.

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

Table of Contents

Introduction

The corporate action (CA) process for BONDS works in two steps in the system: first the system creates the CA product from the bond cash-flows. Then the system applies this CA product to the positions (inventory and P&L positions) and creates the CA trades.

Calypso offers the possibility to:

- Import MT564 messages that are indexed and reconciled to existing CA products
- Import MT566 (NEWM and REVR) messages that are indexed and reconciled to existing CA transfers.

Supported CAs:

Swift code	CA TYPE	CA SUBTYPE	Comments
INTR	CASH	INTEREST	This is the payment of interest. It results in a CASHMOVE only.
PCAL	REDEMPTION	PRINCIPAL	This is a partial redemption of an Amortizing bond. It results in a SECMOVE + CASHMOVE subsequences in MT566 to be matched with one CA trade.
PCAL	REDEMPTION	CALL_REDEMPTION	This is a partial redemption via bond schedule. It results in a SECMOVE + CASHMOVE subsequences in MT566 to be matched with one CA trade.
PRED	AMORTIZATION	AMORTIZATION	This is a partial redemption of sinking bond via a change of pool factor (uses the difference between two factors to calculate the principal reduction) It results in a CASHMOVE only.
PRED	PAYDOWN	PAYDOWN	This is a partial redemption of ABS bond via a change of pool factor (uses the difference between two factors to calculate the principal reduction) It results in a CASHMOVE only.
REDM	REDEMPTION	REDEMPTION	This is a full redemption of a bond at maturity. Note that REDM should also be used when making the final payment for a fixed income security that has paid-down in part during its term. It results in a SECMOVE + CASHMOVE.
MCAL	REDEMPTION	CALL_REDEMPTION	This is an early full redemption. Note that MCAL should also be used when making the final payment for a fixed

Swift code	CA TYPE	CA SUBTYPE	Comments
			income security that has paid-down in part during its term. It always results in a SECMOVE + CASHMOVE.
MCAL	REDEMPTION	REDEMPTION	This is an early full redemption of an ABS (should also be used when making the final payment for a fixed income security that has paid-down in part during its term.) This will be handled via call schedule for all bonds but ABS. For ABS Bonds, this will be done by an update of the pool factor to 0 which generates a CA type REDEMPTION. It always results in a SECMOVE + CASHMOVE.

Configuration

2.1 Account Setup

Custody accounts:

The attribute XferAgentAccount must be defined with the reference of the account provided by the custodian (TAG :97A::SAFE// of the MT56X)

This reference is stored in the CA trade keyword CAAgentAccountId and is used to retrieve the corresponding Settle Account and SDI in the system.

NB: In case there's more than one account with this reference (AUTO ACCOUNT), the system will also parse the content of TAG:94F::SAFE// to retrieve the subcustodian country and compare the ISO with the value of the account property SubcustodianISO. The subcustodian country is also stored in the agent details of the CA and propagated to the CA trades in the keyword SubcustodianISO.

Cash accounts:

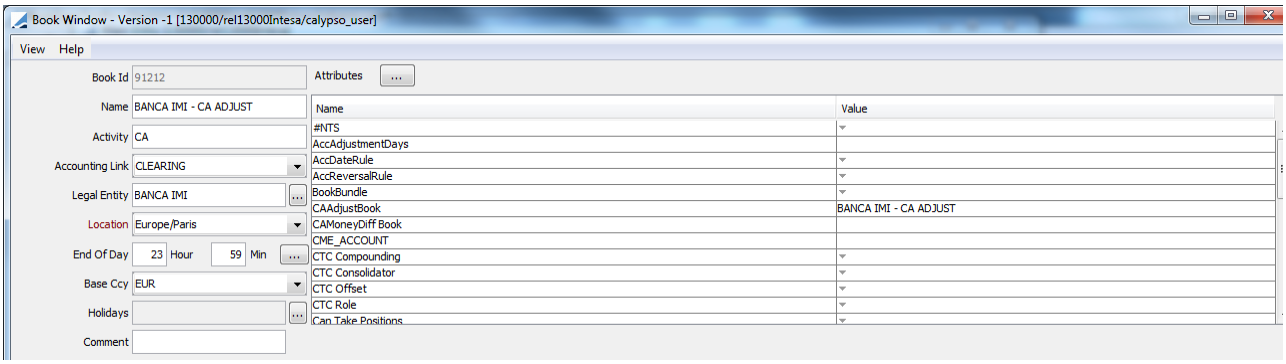
The attribute XferAgentAccount must be defined with the reference of the account provided by the custodian (TAG :97A::SAFE// of the CASHMOVE sequence of the MT56X)

This reference is stored in the CA trade keyword CAAgentCashAccountId and is used to retrieve the corresponding Settle Account and SDI in the system.

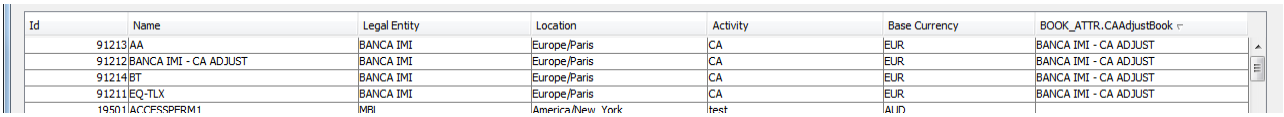
2.2 CAAdjustBook Definition

MT566 must be matched with a global transfer i.e. transfer that represents the amount settled with the agent. To do so, the system generates additional CA trade on a CAAdjustBook.

Define this adjustment book as follow:



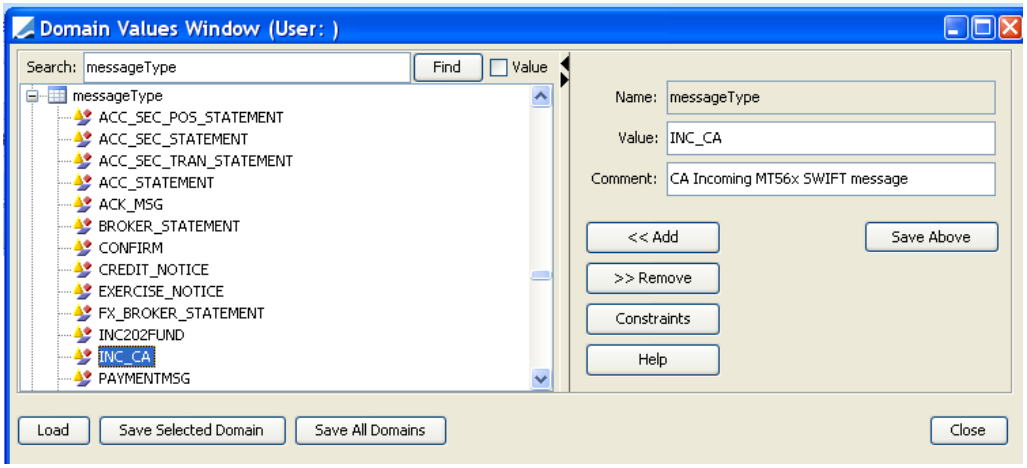
Then add the attribute CAAdjustBook on all trading books.



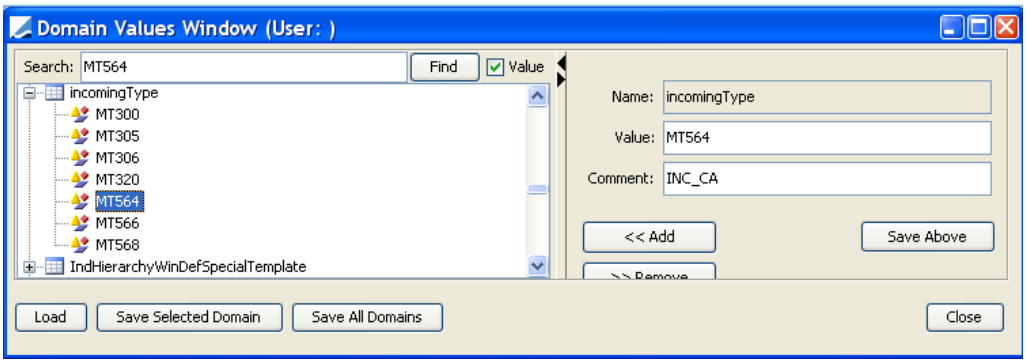
Id	Name	Legal Entity	Location	Activity	Base Currency	BOOK_ATTR.CAAdjustBook
91213	AA	BANCA IMI	Europe/Paris	CA	EUR	BANCA IMI - CA ADJUST
91212	BANCA IMI - CA ADJUST	BANCA IMI	Europe/Paris	CA	EUR	BANCA IMI - CA ADJUST
91214	BT	BANCA IMI	Europe/Paris	CA	EUR	BANCA IMI - CA ADJUST
91211	EQ-TLX	BANCA IMI	Europe/Paris	CA	EUR	BANCA IMI - CA ADJUST
19501	ACCESSPERM1	IMBI	America/New_York	test	AUD	

2.3 Message Type Configuration

The scheduled task MESSAGE_MATCHING can be used to import MT564/MT566 in the system. BO messages are saved with message type defined in the domain as shown below.



The Domain name 'IncomingType' is used to assign to MT564, MT566, and MT568 the BO message type INC_CA. The BO message type is specified in the comment field of the domain value (INC_CA).

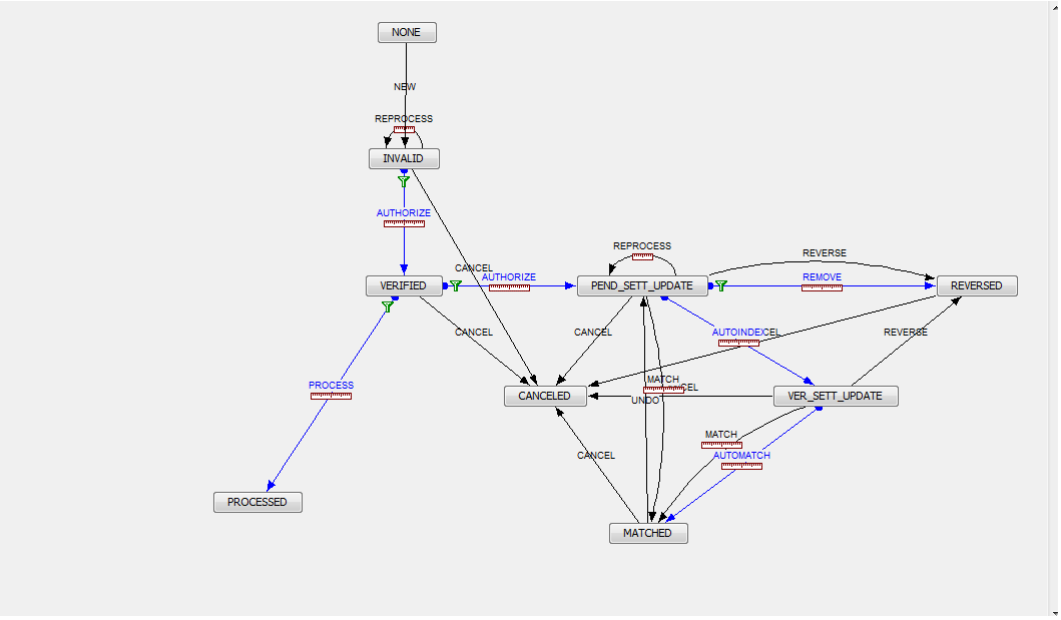


2.4 Message Workflow

Message workflow rules description:

- CheckIncomingFormat (property Validate SWIFT format must be true): this rule will block the incoming message in case its format is not SWIFT compliant.
- CheckIncomingProcess: this rule will block the incoming message in case there's a processing error. There's a processing error in case the message is not indexed to a CA (indexing issue) or in case at least one field of the message doesn't match the CA product (matching issue).

- ReprocessIncoming rule try to reprocess a BO message. Let suppose a static data is missing in the system. The MT56x will stay in “INVALID” status. After inserting the missing static data you can use the REPROCESS action to execute the ReprocessIncoming rule and reprocess the message
- ReverseLinkedMessage rule is used to apply the action REVERSE on the linked message in case MT566 REVR is matched.
- SetXferMessageRef rule is used to store the message id as transfer attribute when incoming MT566 is successfully indexed. Action UPDATE must be available.



Subtype = INC_CA

Orig Status	Action	Resulting Status	Use STP	Rules	Filter
INVALID	AUTHORIZE	VERIFIED	true	CheckIncomingFormat	CA MT566
INVALID	CANCEL	CANCELED	false		
INVALID	REPROCESS	INVALID	false	ReprocessIncoming	
MATCHED	UNDO	PEND_SETT_UPDATE	false		
NONE	NEW	INVALID	false		
PEND_SETT_UPDATE	AUTOINDEX	VER_SETT_UPDATE	true	SetXferMessageRef	
PEND_SETT_UPDATE	MATCH	MATCHED	false	MatchIncomingSecurity	

Orig Status	Action	Resulting Status	Use STP	Rules	Filter
PEND_SETT_UPDATE	REMOVE	REVERSED	true	ReverseLinkedMessage	REVR Messages
PEND_SETT_UPDATE	REPROCESS	PEND_SETT_UPDATE	false	ReprocessIncoming	
PEND_SETT_UPDATE	REVERSE	REVERSED	false		
PEND_SETT_UPDATE	RETRY	CANCELED	false	RegenerateIncoming	
VERIFIED	AUTHORIZE	PEND_SETT_UPDATE	true	CheckIncomingProcess	MT566
VERIFIED	CANCEL	CANCELED	false		
VERIFIED	PROCESS	PROCESSED	true	CheckIncomingProcess	MT564
VERIFIED	RETRY	CANCELED	false	RegenerateIncoming	
VER_SETT_UPDATE	AUTOMATCH	MATCHED	true	MatchIncomingSecurity	
VER_SETT_UPDATE	MATCH	MATCHED	false	MatchIncomingSecurity	
VER_SETT_UPDATE	REVERSE	REVERSED	false		

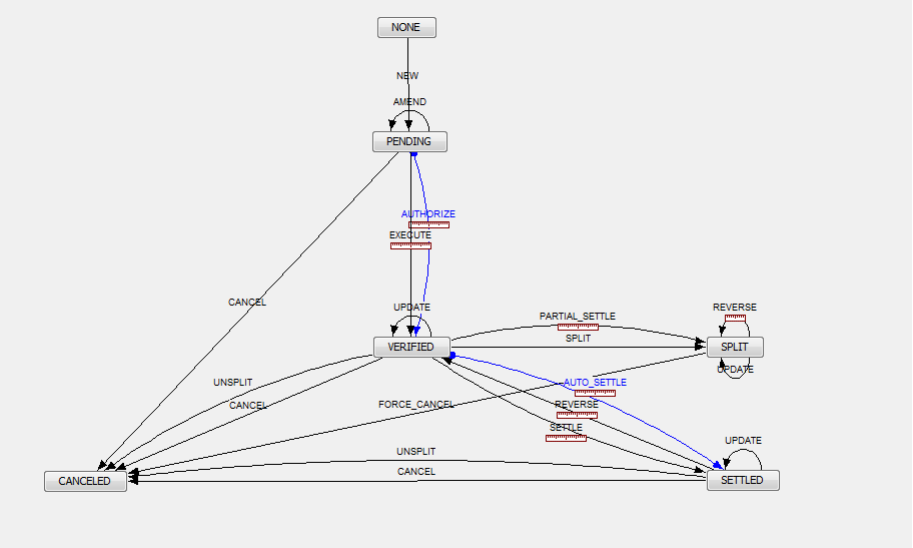
RegenerateIncoming - Rule to produce an event PSEventReProcessMessage that is consumed by the Import Message engine to regenerate the incoming message.

You need to add PSEventReProcessMessage to the domain eventClass. It needs to be consumed by the Import Message engine. If the Import Message engine is only used to process those types of events, you can run it with the engine parameter “config = noconfig”.

2.5 Transfer Workflow

Transfer workflow rules description:

- UpdateCAAdjustBookLinkedXfer rule is used to apply the same action (SETTLE or REVERSE in this transfer workflow) on transfers attached to the trading books trades.
- UpdateCASecurityLinkedXfer rule is used to apply the same action on security transfer of the CAAdjustBook. Thus the security transfer is automatically SETTLED (or CANCELED in case of REVR) once a MT566 is matched with the cash CA transfer.
- UnSplitXfer rule is used to REVERSE the transfers coming from a PARTIAL_SETTLE in case REVR MT566 is matched.
- UpdateINDEVALLinkedXfer - It should be set on the SETTLE action. It applies the SETTLE action on transfers coming from CA trades with Agent INDEVAL.



Orig Status	Action	Resulting Status	Use STP	Rules	Comment	Filter
NONE	NEW	PENDING	false			
PENDING	AMEND	PENDING	false			
PENDING	AUTHORIZE	VERIFIED	true	CheckNetting		
PENDING	CANCEL	CANCELED	false			
PENDING	EXECUTE	VERIFIED	false	SetKnownFlag		
SETTLED	REVERSE	VERIFIED	false	UpdateCAAdjustBookLinkedXfer, UpdateCASecurityLinkedXfer		
SETTLED	UNSPLIT	CANCELED	false			
SETTLED	UPDATE	SETTLED	false			
SPLIT	REVERSE	SPLIT	false	UpdateCAAdjustBookLinkedXfer, UpdateCASecurityLinkedXfer, UnSplitXfer		
SPLIT	UPDATE	SPLIT	false			
VERIFIED	AUTO_SETTLE	SETTLED	true	CheckToBeSettled, UpdateCAAdjustBookLinkedXfer, UpdateCASecurityLinkedXfer		
VERIFIED	CANCEL	CANCELED	false			

Orig Status	Action	Resulting Status	Use STP	Rules	Comment	Filter
VERIFIED	PARTIAL_SETTLE	SPLIT	false	UpdateCAAdjustBookLinkedXfer, UpdateCASecurityLinkedXfer		
VERIFIED	SETTLE	SETTLED	false	UpdateCAAdjustBookLinkedXfer, UpdateCASecurityLinkedXfer, CheckToBeSettled		
VERIFIED	SPLIT	SPLIT	false			
VERIFIED	UNSPLIT	CANCELED	false			
VERIFIED	UPDATE	VERIFIED	false			

2.6 Message Attributes

It is possible to add the following fields in the domain MsgAttributes. These attributes are used to ease the automatic incoming processing and can also be used in the security matching window or any report to ease the manual reconciliation/matching.

- CAReference: ID of the corporate action product
- Process Issue: attribute used to store the reason why the MT56X processing has failed
- Process_Status: content of TAG25D::PROC//
- Format Issue: attribute used to store the reason why message is not swift compliant
- Message_Function: content of TAG23G
- ISIN: content of TAG 35B
- Swift_Event_Code: content of TAG22F::CAEV//
- Payment_Date: content of TAG98A::PAYD// (else 98A::EFFD//)
- Ex_Date: content of TAG98A::XDTE// (else 98A::MATU//)
- Record_Date: content of TAG98A::RDTE//
- AgentRef: content of TAG20C::SEME//
- CA_Agent_Ref: content of TAG20C::CORP//
- Period_Rate: content of TAG92A::INTP//
- Fixed_Rate: content of TAG92A::INTR//
- Quantity Type: used to identify the subsequence CASHMOVE/SECMOVE
- Del Type: used to identify the direction of the movement (DEBIT/CREDIT)
- Nominal Amount: content of TAG19B::PSTA//
- Ccy: settlement currency
- Settle Date: Content of TAG:98A::POST//

- Amort_Rate: Content of the field :92A:RATE//
- Pool_Factor: Content of the field :92A:NWFC//
- Redm_Price: Content of the field :90A:OFFR//

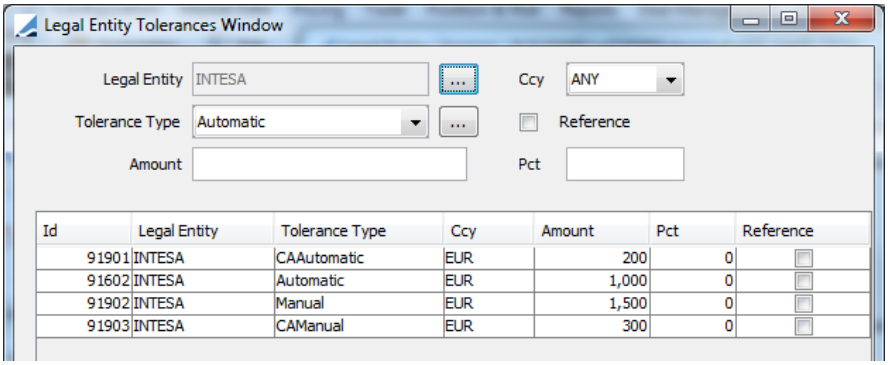
2.7 Trade Keywords

Following trade keywords must be added in the domain tradeKeyword:

- CAAgentAccountId: ID of the custodian settle account
- CAAgentCashAccountId: ID of the cash settle account
- CAReference: ID of the CA product
- CAFailedTransfer: ID of the failed transfer for which a CA claim is needed
- CAClaimReason: Reason why a CA claim trade is needed
- CASource

2.8 LE Tolerance

Tolerance can be defined at subcustodian, Agent or PO level (to be retrieved in this order). It can be an absolute (ccy is mandatory in that case) or relative amount. The Legal Entity Tolerance Window can be reached from the Legal Entity window.



Two types of tolerance are necessary:

- CAAutomatic: case where the amount difference is acceptable and transfer should be matched automatically when processing the incoming MT566
- CAManual: case where the amount difference is not acceptable for automatic matching but acceptable for Manual matching (from the Security Matching window)

In both cases, the system will apply the action PARTIAL_SETTLE on the transfer. The action SETTLE is applied on transfers attached to all "trading book" CA trades but settle amount is not updated. Settle date should still reflect the real settle date on these underlying transfers. The action MATCH is applied on the MT566.

MT56X processing

3.1 MT56X indexation to CA product

When MT56X messages are imported, the system tries to index the bo_message to existing CA, using the following logic:

(These two first checks are valid only in case a MT56X has already been imported for this CA)

- When a MT56X message is imported, the system stores the CA Id (calypso reference) as a message attribute (CAReference) of the MT56X bo message. It means that if a MT56X is imported and TAG 20C::RELA is present, the system will retrieve the CA Id from the previously sent MT56X.
- If TAG 20C::RELA is not present, the system can use the CA Agent Reference to link MT56X to a CA Id. The CA AGENT REFERENCE is provided in TAG 20C::CORP of the MT56X. The value of this TAG can be matched with the CA AGENT REFERENCE coming from the Agent details, the agent details being linked to the CA product in calypso. (NOTE: This CA AGENT REFERENCE was stored at the time former MT564 was imported).

Logic in case this is the first MT564 with these references:

- If the system wasn't able to retrieve the CA product using TAG 20C::RELA or TAG 20C::CORP, then the system will look if a CA exists using several criteria:
  - the ex date (stored on CA product): Content of the field:98A::XDTE// (or 98A::MATU//)
  - CA Bond Model and subtype (stored on CA product): Content of the field 22F::CAEV//
  - ISIN (stored on Bond definition): Content of the field:35B:ISIN

When receiving the MT564 with field :23G:NEWM/COPY, the system updates the CA product with “Function of the Message”.

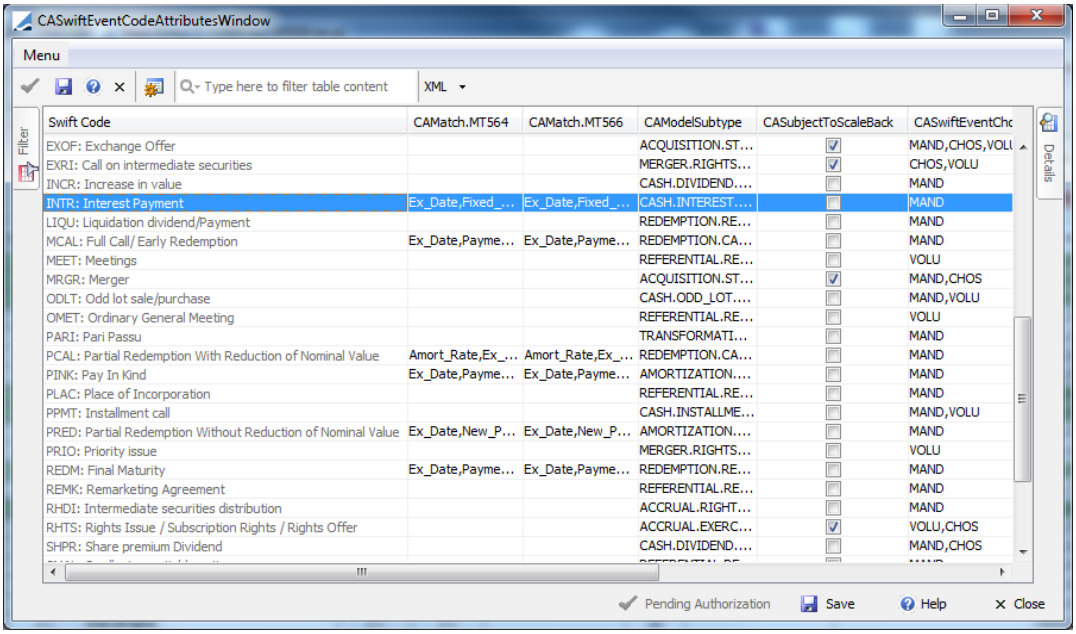
3.2 MT56X matching logic with CA product

Consistency checks are implemented to make sure the CA is in synch with the MT56X. If a check fails, the MT56X is still indexed to the CA but the MT56X is blocked in the workflow and exception is raised. The action REPROCESS with rule ReprocessIncoming can be used once the matching issue is fixed.

Swift code	CA TYPE	CA SUBTYPE	Checks
ANY	ANY	ANY	Record date (stored on CA product): Content of the field:98A::RDTE//

Swift code	CA TYPE	CA SUBTYPE	Checks
ANY	ANY	ANY	Payment date (stored on CA product): Content of the field:98A::PAYD//. If not present match with 98A::EFFD//
INTR	CASH	INTEREST	Fixed_Rate (stored on Bond definition): Content of the field :92A::INTR//0,00 CFP_Period_Rate (Computed from bond cash-flows): Content of the field :92A::INTP//0,00
PCAL	REDEMPTION	PRINCIPAL	Amort Rate (column in the bond cash-flows): Content of 92A:RATE// Redemption price (CA Amount): Content of 90A::OFFR
PCAL	REDEMPTION	CALL_REDEMPTION	Amort Rate (column in the bond cash-flows): Content of 92A:RATE// Redemption price (CA Amount): Content of 90A::OFFR
PRED	AMORTIZATION	AMORTIZATION	Previous pool factor (Bon definition): Content of :92A::PRFC// New pool factor (Bon definition): Content of :92A::NWFC//
PRED	PAYDOWN	PAYDOWN	Previous pool factor (Bon definition): Content of :92A::PRFC// New pool factor (Bon definition): Content of :92A::NWFC//
REDM	REDEMPTION	REDEMPTION	Redemption price (Bond definition) :Content of 90A::OFFR//

The fields to be matched when importing MT56X can be configured in the CASwiftEventCodeAttributes window:



3.3 MT566 NEWM Indexation to CA Trade/Transfer

The incoming MT566 is linked to an existing CA trade. The system retrieves a CA trade with the following conditions:

- Trade counterparty = SENDER of the message
- Trade counterparty role = Agent
- Trade status NOT CANCELED
- Trading book = CAAdjustBook (NB: the CAAdjustBook book attribute is mandatory and must be unique by PO)
- Same CAReference (ie same CA ID)
- Same custodian account. Agent account reference is provided in TAG 97A of the MT566 and can be used to retrieve the correspondent Settle Account in the system.

If a unique CA trade is found, the system will index the MT566 to a CA transfer, and apply the action SETTLE or PARTIAL_SETTLE if the amount is within the tolerance amount. The real settle date of the SETTLED transfer is updated with the content of TAG 98a::POST.

If no (or multiple) CA trade is found the bo_message is indexed to the CA id, but not to the trade id. The user will have to manually match a transfer with the message using the SecurityMatching window.

3.4 MT566 REVR Processing

In case of reversal MT566 (23G = REVR), the system retrieves the previous message reference from TAG 20C::PREV (mandatory as per conditional rule C2 of swift doc) in order to link the reversal with the new MT566 (cf field Msg Linked Id).

In case of reverse of perfect match, the system (MatchIncomingSecurity rule) applies the action REVERSE on the SETTLED transfer to move it back to VERIFIED.

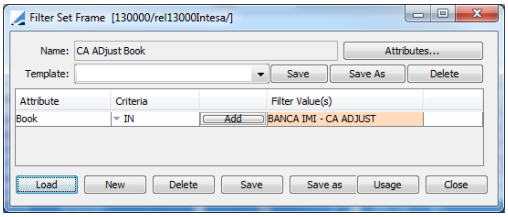
In case of reverse of partial settlement (mismatch), the system applies the action REVERSE (rule UnSplitXfer) on the SPLIT transfer. As a result, the SETTLED/FAILED child transfers are all CANCELED and the system creates a new transfer for the full amount.

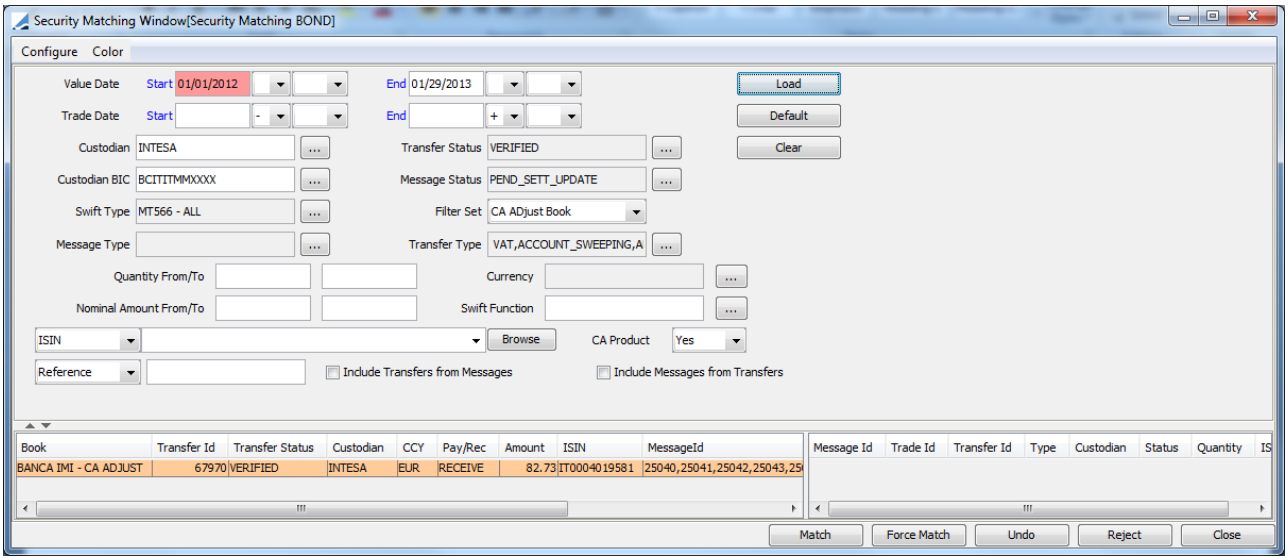
In case previous NEWM MT566 is not indexed to a transfer, the system (rule ReverseLinkedMessage) applies the action REVERSE on the MT566 to move both NEWM and REVR in REVERSED status.

3.5 Security Matching Window and Manual Match

In case no eligible trade/transfer was found, the user will have to manually match incoming MT566 with CA transfers.

FilterSet with book = CAAdustBook must be defined to only load relevant transfers.





MATCH button - within CAManual tolerance: It is possible to define a CAManual tolerance type for manual matching. Manual tolerance can be defined at subcustodian, Agent or PO level (to be retrieved in this order).

FORCE MATCH button: the user also has the possibility to force the match of incoming MT566. In that case, the system doesn't perform any consistency check. The system will apply the action MATCH on the message and SETTLE the selected transfer. The access permission **Manual Match** must be granted to the user.