Nasdaq Calypso

Corporate Cash Management

Version 18

Revision 1.0
February 2024
Approved

© 2025 Nasdaq, Inc. All Rights Reserved.

Private and Confidential

Copyright © 2025, Nasdaq, Inc. All rights reserved.

All content in this document is owned, or licensed, by Nasdaq, Inc. or its affiliates ('Nasdaq'). Unauthorized use is prohibited without written permission of Nasdaq.

While reasonable efforts have been made to ensure that the contents of this document are accurate, the document is provided strictly "as is", and no warranties of accuracy are given concerning the contents of the information contained in this document, including any warranty that the document will be kept up to date. Nasdaq reserves the right to change details in this document without notice. To the extent permitted by law no liability (including liability to any person by reason of negligence) will be accepted by Nasdaq or its employees for any direct or indirect loss or damage caused by omissions from or inaccuracies in this document.

Document History

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

This guide gives instructions and provides examples of how to configure and use the corporate cash management module in Calypso It covers the following features:

- **Integration and processing of incoming MT940, including Automatic Cash Pooling process**
- **Automatic generation of FX and MM consolidation deals**
- **Cash Position, Bank Account Activity, Proof of Cash and Process Status Reports**
- **Active and passive bank account sweeping**
- **Subsidiary internal messages**
- **MT101**
- **Forecasts and In-house Transactions (payment on behalf of, intercompany transactions – single currency or cross-currency - intracompany transactions – s single currency or cross-currency, market transactions, subsidiary interest bearing, etc.)**

Table of Contents

Processing of Incoming Swift MT940 5
1.1 Static Data Configuration 5
1.1.1 Processing Organization Bank Account and Related SDI 5
1.1.2 Subsidiary Internal Account and Related SDI 9
1.1.3 Account Mapping 13
1.1.4 Subsidiary Not Managed Account and Related SDI 19
1.1.5 Subsidiary Payment Factory SDI 19
1.1.6 SetBOValidationDate Trade Workflow Rule 19
1.1.7 Additional Static Data 20
1.2 Matching Framework 21
1.2.1 Matching Context for Cash Management 21
1.2.2 Matching Monitor for Cash Management 35
1.3 BO Cash Settlement trade and Automatic Cash Posting from Bank Account Statement (MT940) 44
1.4 BO Position Update 47
1.4.1 Overview 47
1.4.2 Configuration 48
1.5 Scheduled Task MESSAGE_MATCHING 48
1.5.1 Scheduled Task Attributes 48
1.5.2 Message workflow SWIFT_TRADE_RECON 49
1.6 Scheduled Task INC_CASH_STATEMENT 51
1.6.1 Processing Organization Legal Entity Attributes 51
1.6.2 Scheduled Task Attributes 52
1.6.3 Automatic Cash Pooling Trades 54
FX & MM Consolidation Deals 56
2.1 Account Configuration 56
2.1.1 FX Consolidation 56
2.1.2 MM Consolidation 57
2.2 Scheduled Task ACCOUNT_CONSOLIDATION 57
2.2.1 FX Consolidation deal generation process 58
2.2.2 MM Consolidation deal generation process 59
BO Positions and Process Status Reports 61
3.1 BO Position 61
3.2 Account Activity Report 61
3.3 BO Position Reconciliation Report 62

3.3.1 Scheduled Task INVENTORY_SNAPSHOT 62
3.3.2 BO Position Reconciliation Report 64
3.4 Process Status Report 68
3.4.1 Scheduled Task CHECK_INC_STATEMENT 68
3.4.2 Report 69
Account Sweeping 71
4.1 Active Account Sweeping Process from BO Position 71
4.2 TARGET_BALANCE Active Account Sweeping 72
4.3 Passive Account Sweeping Process 73
Internal Messages 74
5.1 Scheduled Task ACCOUNT_STATEMENT 74
5.2 Subsidiary Internal MT940 75
5.3 Subsidiary Internal Bank Account Statement 76
5.4 Subsidiary Internal Interest Advice Message 78
5.5 Subsidiary Statement of Holdings Internal Message 80
Swift MT101 Payment Message 81
6.1 MT101 Setup 81
6.1.1 Individual MT101 81
6.1.2 Grouped MT101 82
Forecasts and In-House Transactions 87
7.1 Subsidiary Forecasts 87
7.2 In-House Transactions 88
7.2.1 Payment on behalf of 88
7.2.2 Intercompany Transactions 89
7.2.3 “In-House” Market Transactions with a Subsidiary 93
7.2.4 Interest Bearing Transactions 94
Appendix 98

Processing of Incoming Swift MT940

1.1 Static Data Configuration

To be able to integrate and process incoming MT940, you must initially configure the appropriate bank accounts and settlement instructions.

1.1.1 Processing Organization Bank Account and Related SDI

Define a SETTLE Account

Bank Account for which the processing organization receives and processes MT940 must be configured in Calypso as a standard SETTLE Account for that specific Processing Organization, Currency and Agent.

This type of account must then be attached to the related Processing Organization SDI.

The Account Name for this type of account is generic (for instance, as stated below, BARCLON-EUR for the EUR Account of the Processing Organization PARENT_COMPANY at Barclays Bank PLC Ldn).

Populate the Account Attribute XferAgentAccount

For this type of account, we store the value of Tag 25 used by the Agent when sending a MT940 as an Account Attribute “XferAgentAccount”.

To do so, once on the SETTLE Account window select the Properties/Attributes button and add the value of Tag25 as value of the account attribute XferAgentAccount (see below).

Populate the Account Attribute PaymentFactory

If the bank account is a Payment Factory account, there is no difference with a standard account except that we identify that type of account with a specific account attribute PaymentFactory = true.

For a standard bank account, this attribute is left blank. We only set the value to true to mark the account as a payment factory account.

Populate the Account Attribute GuaranteeFees

If the bank account is a Guarantee Fees account, there is no difference with a standard account except that we identify that type of account with a specific account attribute GuaranteeFees = true.

For a standard bank account, this attribute is left blank. We only set the value to true to mark the account as a guarantee fees account. When this is set to true, the SubStatement linked to such bank account will have a SubStatement.GuaranteeFee type. The attached trade, created automatically by the system, will be typed with a

keyword CashStatementProcess = Automatic Guarantee Fee. The related transfer will have a “GUARANTEE FEE” description instead of cash pooling.

Attach an Incoming Statement Configuration

These types of accounts are identified as ‘Account to reconcile’ based on their **Incoming Statement Configuration**.

Only SETTLE accounts defined with an incoming statement configuration are taken into account for the MT940 reconciliation process.

To add an Incoming Statement Configuration, go to the statement panel of the SETTLE Account, choose Statement type = Incoming, set an ‘active from’ date for that configuration (default is today) and select a date rule.

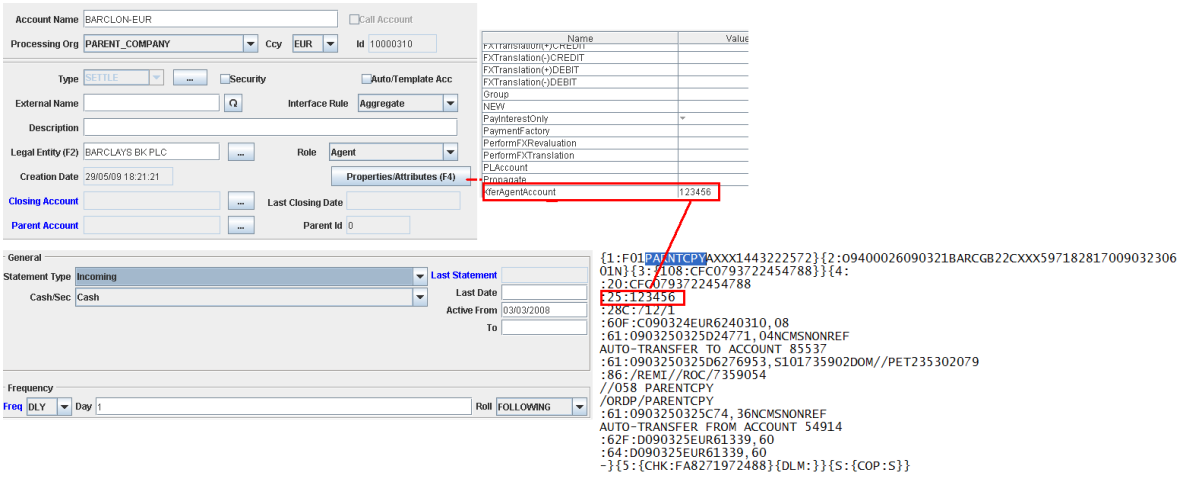
The Date Rule attached to the Incoming Statement Configuration in the Frequency panel is used to determine when we expect an Incoming MT940 for this specific account. We do not process an incoming MT940 if the date rule does not expect such a statement for a specific date; inversely, we raise an exception if a statement expected by the date rule is not received. See the Process Status Reporting section for more details.

Define the Processing Organization and Bank/Agent SWIFT Codes (LE Contact)

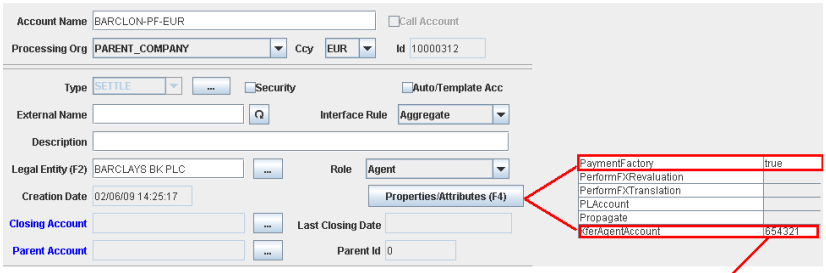
To be able to map an incoming MT940, you need to setup the related SWIFT Codes for the Processing Organization and Agent/Bank. As we integrate ‘outgoing’ SWIFT Messages, block 1 identifies the Receiver (Processing Organization) and block 2 the Sender (Bank/Agent). Thus, to be able to process the incoming MT940 you need to set in the LE Contact Window the SWIFT Address of the Processing Organization (to map block 1 swift address of the message receiver) and the SWIFT Address of the Agent (to map block 2 swift address of the message sender).

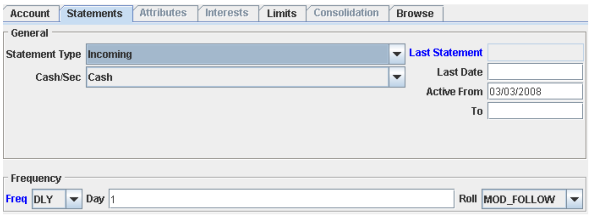
P.O. Bank Account to Reconcile - Setup Example

An example is presented below for a ‘standard’ bank account for the processing organization PARENT_COMPANY (Swift is PARNTCPY) with the agent BARCLAYS BANK PLC (BARCGB22) – second block displays the Account Statement Config:



Another example is presented below for the payment factory account of the processing organization PARENT_COMPANY with the same agent BARCLAYS BANK PLC:






```
{1:F01PARNTCPYAPAY1443222572}{2:O9400026090321BARCGB22CXXX597182817009032306
01N}{3:{108:CFC0793722454788}}{4:
:20:CFC0793722454788
:25:654321
:28C:712/1
:60F:C090324EUR6240310,08
:61:0903250325D24771,04NCMSNONREF
AUTO-TRANSFER TO ACCOUNT 85537
:61:0903250325D6276953,S101735902DOM//PET235302079
:86:/REMI//ROC/7359054
//058 PARENTCPY
/ORDP/PARENTCPY
:61:0903250325C74,36NCMSNONREF
AUTO-TRANSFER FROM ACCOUNT 54914
:62F:D090325EUR61339,60
:64:D090325EUR61339,60
-}{5:{CHK:FA8271972488}{DLM:}}{S:{COP:S}}
```

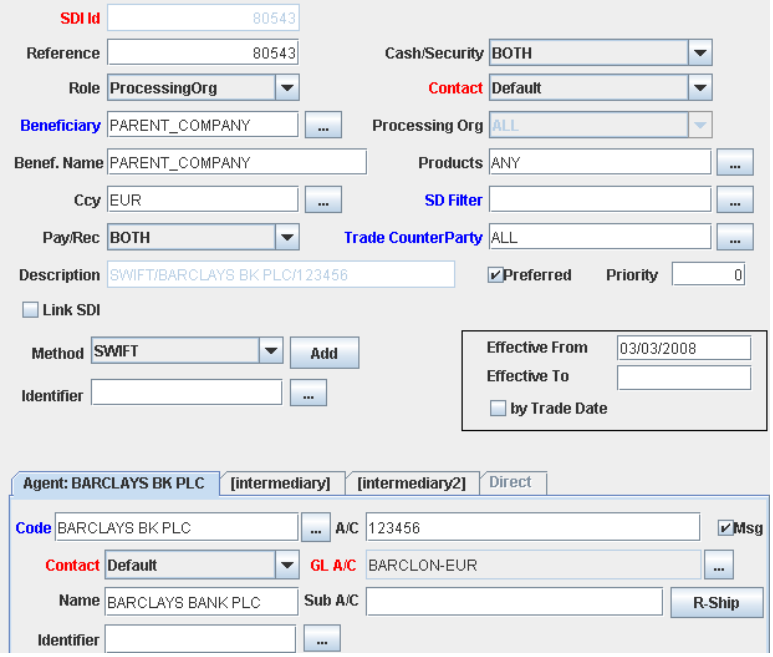
Same setup would be needed for a guarantee fees account, the only difference being to populate the GuaranteeFees account attribute to true to handle it properly.

Define the Related SDI

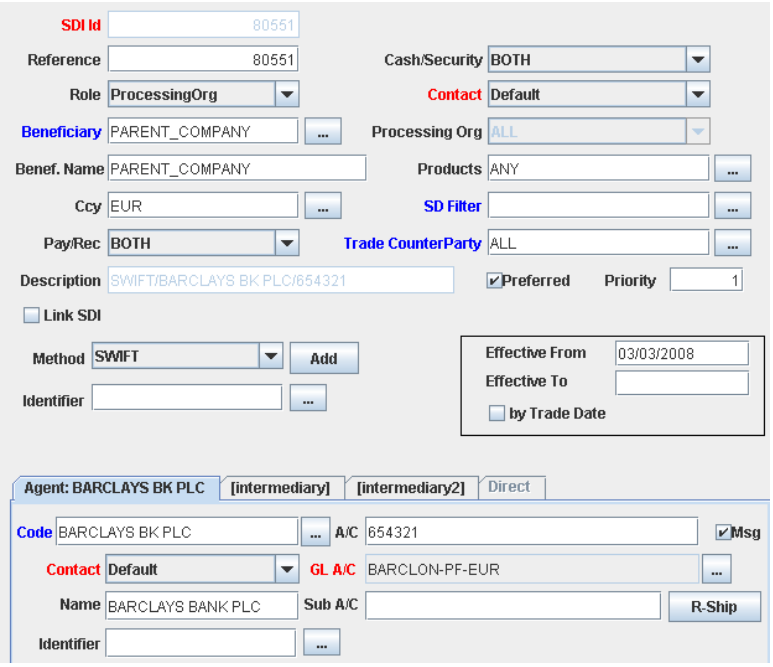
External PO SDI

Each processing organization bank account is attached to a SDI for the Role = Processing Organization and a SETTLE Method = SWIFT (for instance).

Below an example for the standard account of the processing organization PARENT_COMPANY with the agent BARCLAYS BANK PLC:



Below another example for the payment factory account of the processing organization PARENT_COMPANY with the same agent BARCLAYS BANK PLC:



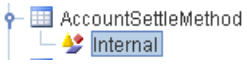
Internal PO SDI

In addition, we must define an internal processing organization SDI by currency.

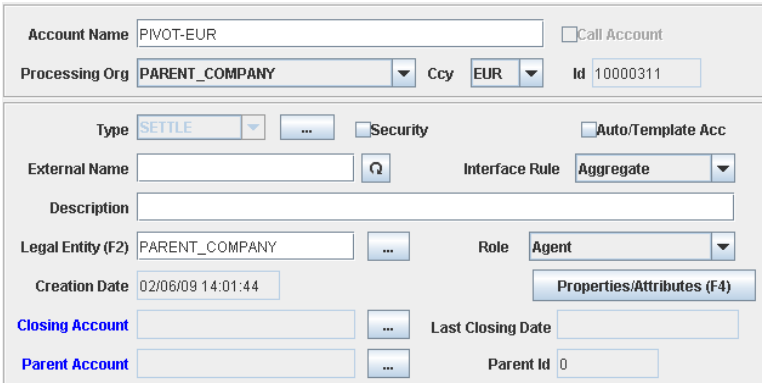
This SDI is only used for internal transactions: it is a ‘technical’ SDI picked by the system for cash pooling and inter-company transactions.

This SDI refers to a dummy 'PIVOT-currency specific' account previously defined with Agent = PO.

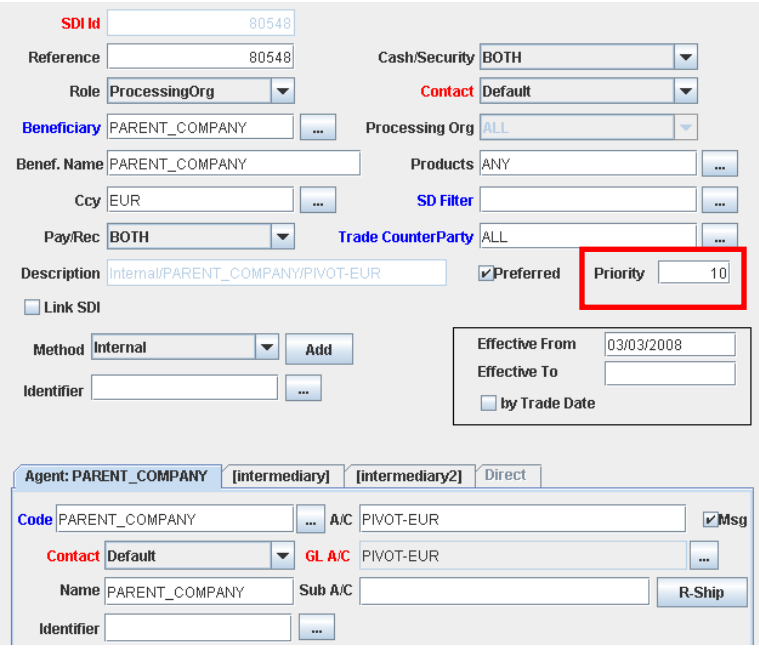
Before creating the 'Internal' SDI, go to **Configuration > System > Domain Values** – AccountSettleMethod and add the **Internal** settle method as shown below:



Then, create the PIVOT-"currency-specific" SETTLE Account with Agent = Processing Organization (example below with PIVOT-**EUR**):



Finally, add the internal processing organization SDI using that dummy 'PIVOT-CCY' account as shown below. Make sure to **assign a higher priority than the one set on the external SDIs.**

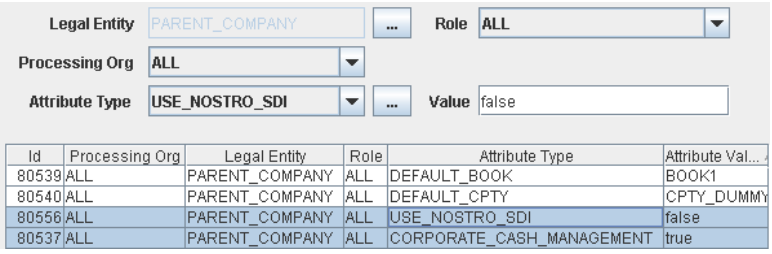


1.1.2 Subsidiary Internal Account and Related SDI

Internal company accounts are configured as **Call Accounts** in Calypso.

Before creating a call account, add the following **Processing Organization** attributes:

- CORPORATE_CASH_MANAGEMENT = true (this will create the call account SDI for Role AccountHolder with Product = ALL). This is needed to be able to select the AccountHolder SDI for any product (FX, Cash…)
- USE_NOSTRO_SDI = false (this will only create call account SDI for Role AccountHolder)



To activate the call account functions, go to the user environment, choose “add property” and add USE_CALL_ACCOUNT = true.

Please also add the SetCallAccountId trade workflow rule on a product-specific call account trade workflow to set a common reference (= CallAccountId) on call account related trade (customer transfer, interest bearing). For more information regarding call account statements, please refer to the Client Custody Management User Guide.

All ‘cash pooling’ movements from the subsidiary accounts which are confirmed by the MT940 will be mirrored on its related call account, defined by subsidiary, currency and processing organization.

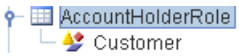
The processing organization reports the balance, produces bank account statements on a predefined frequency (or ad-hoc) and computes interest bearings for these internal accounts.

Each subsidiary has one unique call account by currency. Thus, movements in the same currency but reported by different incoming MT940 (different banks/different accounts) for the same subsidiary impact the same subsidiary call account.

Define Subsidiary as Internal ‘AccountHolder’

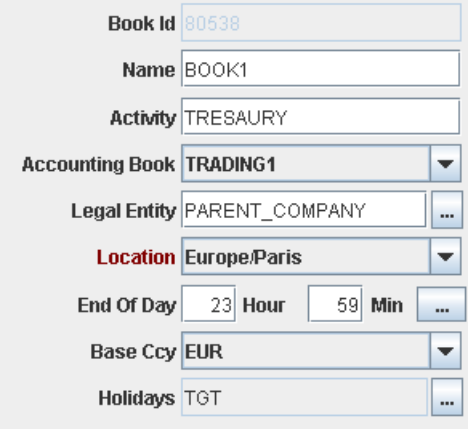
Subsidiaries must be defined in Calypso with the Role CounterParty and the Role = AccountHolder

To define the AccountHolder role, go to **Configuration > System > Domain Values** - AccountHolderRole and set the Role name. In our example below AccountHolder Role has been set as ‘Customer’, but it could be set to a different value (for instance Subsidiary):

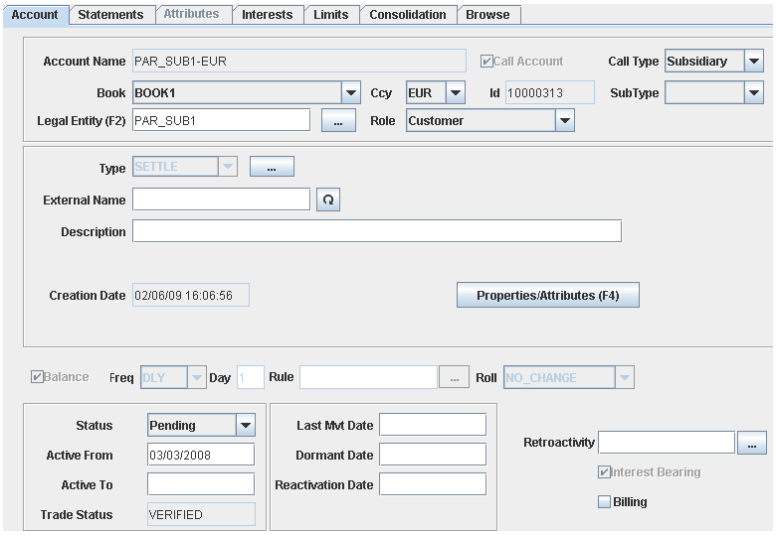


Define the Subsidiary Call Account

A call account is linked to a Book. Before defining a call account, define a book related to your processing organization (BOOK1 in our case, linked to the processing organization PARENT_COMPANY).

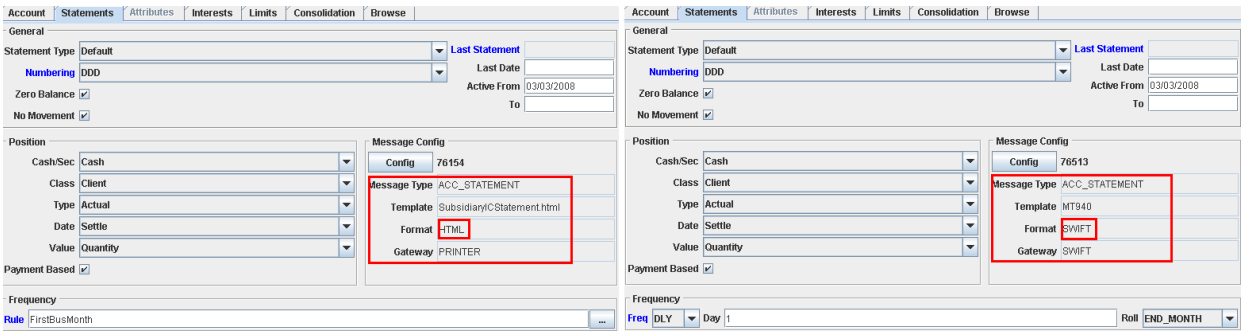


Then, to define a call account, go to **Configuration > Accounting > Accounts**, flag 'call account' and configure a subsidiary internal account as shown below for PAR_SUB1-EUR:



Subsidiary call accounts **must** be defined with statement configurations (at least one/usually more than one). An example is shown below:

- one to produce daily internal Swift MT940
- one to produce monthly bank account statement (HTML)

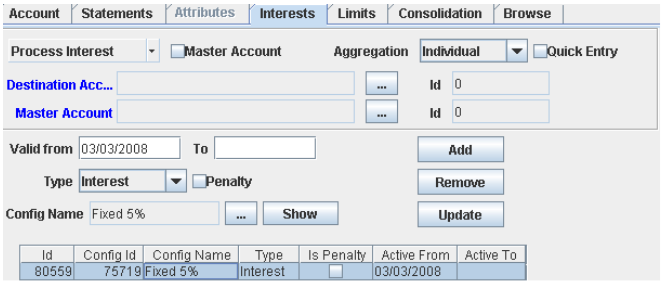


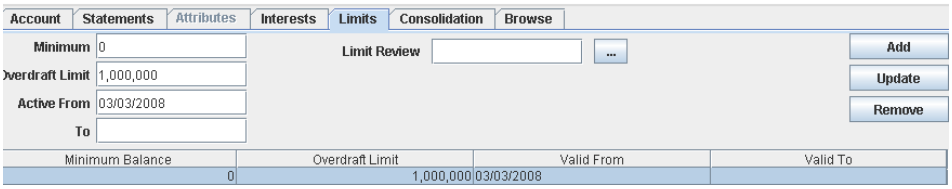
Calypso provides out-of-the-box different HTML Statement Templates:

- SubsidiaryICStatement.html provides an HTML Internal Bank Account Statement in Value Date
- SubsidiaryICStatementBookingDate provides an HTML Internal Bank Account Statement in Booking Date

The Swift Template MT940 is based on Trade or Transfer Booking Date, depending on your trade workflow setup (usage or not of the "SetBOValidationDate" trade workflow rule).

Subsidiary call accounts must also be defined with an interest bearing configuration (and limit information when a limit check must apply). An example is presented below:



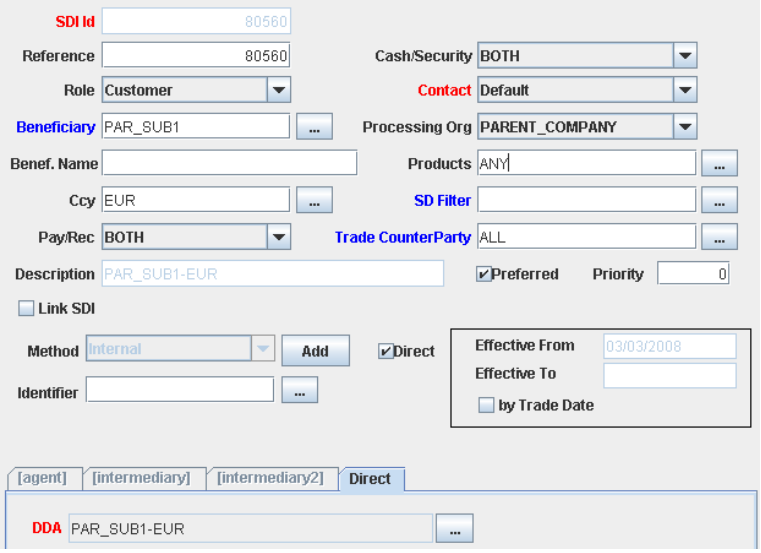


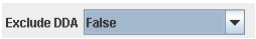
Subsidiary Automatic Call Account SDI

As soon as a subsidiary call account is saved, the system automatically generates the related 'internal' SDI.

When the attributes USE_NOSTRO_SDI = false and CORPORATE_CASH_MANAGEMENT = true for your processing organization(s), only **one** automatic SDI is created for call accounts linked to that processing organization.

This SDI is created for the Role Customer (or role set in the domain 'AccountHolderRole') and product = ANY, as shown below:



From the SDI browser, you need to set the ExcludeDDA to false 

to be able to display that SDI.

That SDI will be used for cash pooling (including guarantee fees accounts) deals and all in-house transactions.

1.1.3 Account Mapping

A new table 'Account Mapping' has been added to store the Bank Account Relationships and map the Incoming Statement Information.

Thus:

- To map the information contained in Tags 61 and/or 86;
- To define the dependencies between the processing organization bank accounts (and so the received MT940) and the related internal subsidiaries accounts,

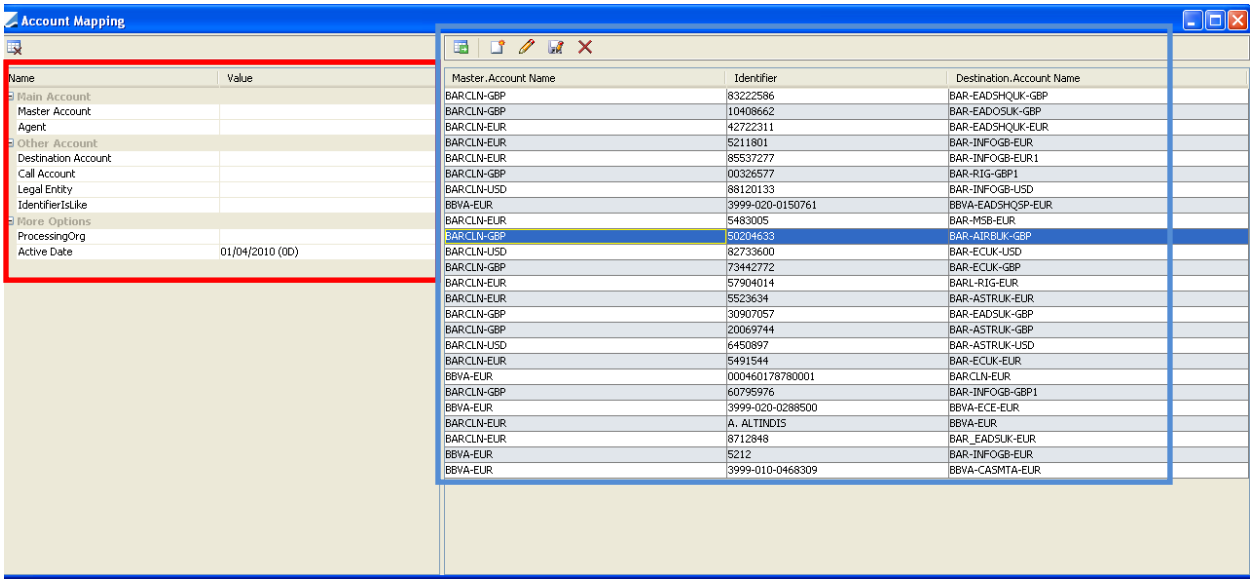
You need to define a mapping:

- Between the processing organization accounts (the so-called 'Master Account') and the internal subsidiaries (call) accounts
- Between two processing organization bank accounts (for automatic processing organization bank accounts sweeping)

This table also allows configuring automatic bank fees booking.

Account Mapping Overview

To access the Account Mapping, go to **Configuration > Accounting > Mapping**.



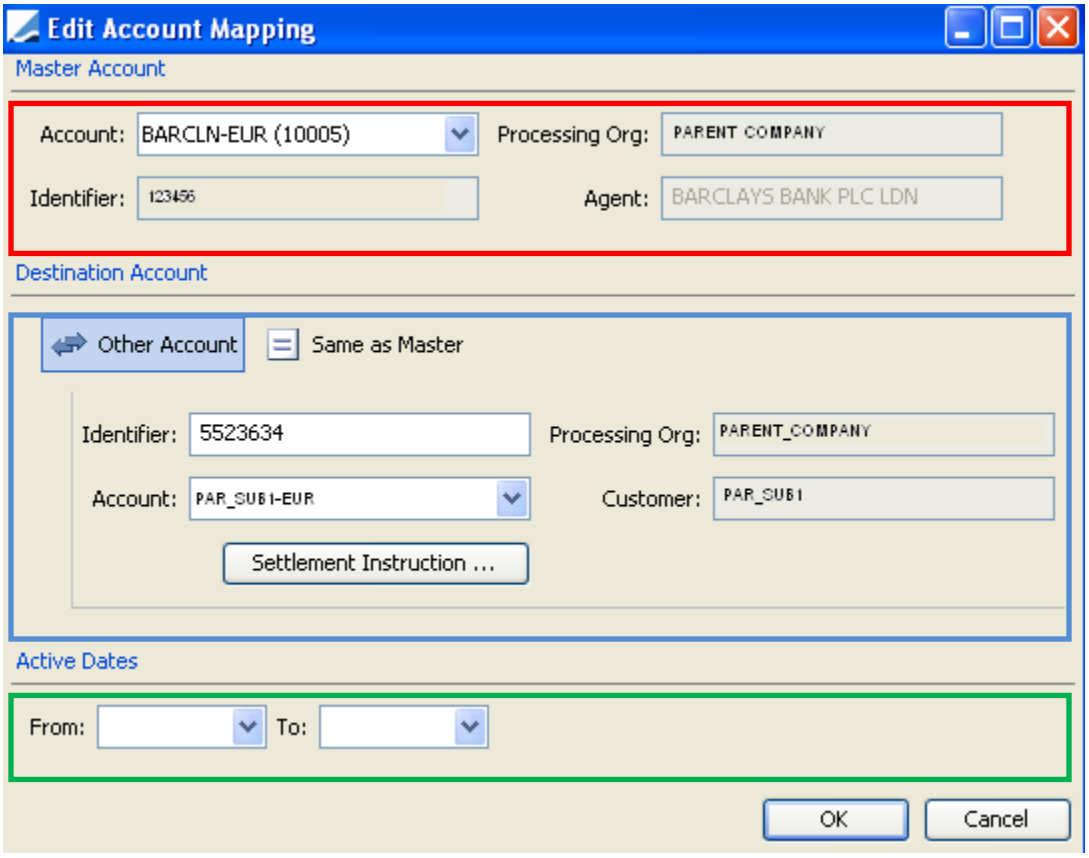
In red above, the Browser part of the Account Mapping Window. Selection criteria include:

- Master (P.O.) Account (= Statement Receiver) = to load only mapping rules for that specific P.O. Account
- Agent (= Statement Sender) = to load mapping rules involving only that Bank
- Destination Account = to load mapping rules for a specific destination account (either another or the same P.O. Account or a Subsidiary Call Account)
- Call Account = false to load Call Accounts and SETTLE Accounts - true to limit the load to call account mapping rules only
- Legal Entity = to load mapping rules for a specific Account Holder
- IdentifierLike = to load mapping rules based on one part of the identifier value (from the statement)
- Processing Organization = to load only mapping lines involving a specific processing organization
- Active Date = to load only active mapping rules based on tenor

In blue above, the Details part for the mapping rules defined. Options for that panel include:

- Load
- New mapping
- Edit (to show a mapping rule or modify an existing one)
- Duplicate
- Delete

Mapping for Automatic Cash Pooling



The above setup allows the automatic creation of customer transfer deals for the internal cash pooling movements impacting both the processing organization settle account (either standard, payment factory or guarantee fees) and the subsidiary internal account.

Step 1 “Master Account”

In red above, you must select the processing organization account for which you receive an incoming statement to process.

The selection is done by clicking on the Account field that pops-up the Account Selector window. The ‘Master Account’ is the processing organization SETTLE Account for which we receive an incoming statement (MT940).

By selecting the account, the system defaults the grey fields (Processing Org Name, Identifier = Account Attribute XferAgentAccount, Account Agent Name) from the account static data.

Step 2 “Destination Account”

In blue above, you must select the ‘OtherAccount’ button, enter the identifier from Tag 61 or 86 of your incoming statement (MT940) then select the **subsidiary call account** by clicking on the Account fields. The system then pops-up the Account Selector window. Once the subsidiary call account selected, the system defaults the grey fields (Processing Org and Customer from the Account Static Data).

From the call account, we also allow automatic creation of the CounterParty External SDI (the so-called subsidiary not managed account/SDI). To do so, once the call account is selected you just have to press the "Settlement Instruction" button. If no SDI exists for the 'MasterAccount' Agent and the 'Destination Identifier' Identifier, then the system automatically creates that external SDI for the Call Account Holder with Role = CounterParty, Currency = Call Account Currency and Product = ALL. This external SDI is used to populate the cash pooling transfer description with the Bank and Subsidiary External Account Number. That information is then displayed on the subsidiary internal bank account statement. See SDI Section below for details.

Please note that for standard cash pooling (not involving a Payment Factory Account) we advise to create the related Counterparty external SDI from the Account Mapping using the Settlement Instruction button.

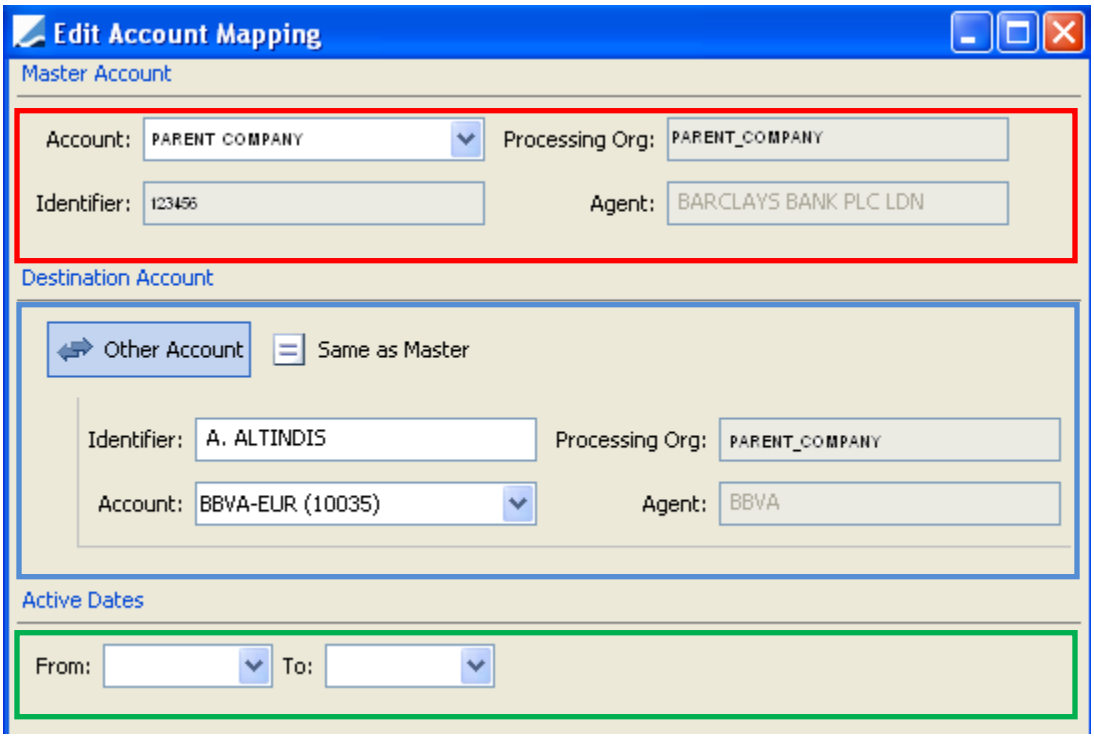
For Payment Factory Cash Pooling, we do not advise to create the related Counterparty external SDI from the Account Mapping using the Settlement Instruction button. Indeed, the identifier being a specific code (e.g. Subsidiary Kapis) we do not need a CounterParty SDI. Instead, a generic SDI, manually defined for the processing organization with the Role CounterParty, Agent = Processing Organization, Currency = Payment Factory Account Currency, will be picked by the system when it does not find an external CounterParty SDI with A/C field = Destination Account Mapping Identifier.

Step 3 "Active Dates"

In green above, you need to populate the "Active From/To" mapping Dates.

Default is blank but as mapping rules can evolve these fields allow putting a "To" and "From" Dates to the mapping rules.

Mapping for Automatic P.O. Bank Account Sweeping



The above setup allows the automatic creation of transfer agent deals for the processing organization bank accounts sweeping, when bank account are automatically swept by the bank on behalf of the processing organization. This movement is created in the system once we receive and process the incoming statement (MT940).

Step 1 “Master Account”

In red above, you must select the processing organization account for which you receive an incoming statement to process.

The selection is done by clicking on the Account field that pops-up the Account Selector window. The ‘Master Account’ is the processing organization SETTLE Account for which we receive an incoming statement (MT940).

By selecting the account, the system defaults the grey fields (Processing Org Name, Identifier = Account Attribute XferAgentAccount, Account Agent Name) from the account static data.

Step 2 “Destination Account”

In blue above, you must select the ‘OtherAccount’ button, enter the identifier from Tag 61 or 86 of your incoming statement MT940 then select the **processing organization account** by clicking on the Account fields. The system then pops-up the Account Selector window. Once the processing organization account selected, the system defaults the grey fields (Processing Org and Agent from the Account Static Data).

Step 3 “Active Dates”

In green above, you need to populate the “Active From/To” mapping Dates.

Default is blank but as mapping rules can evolve these fields allow putting a “To” and “From” Dates to the mapping rules.

Automatic Sweeping between two Accounts belonging to two different Processing Organizations.

It is possible to automatically sweep funds from a bank account belonging to a PO to a bank account belonging to another PO (member of the group).

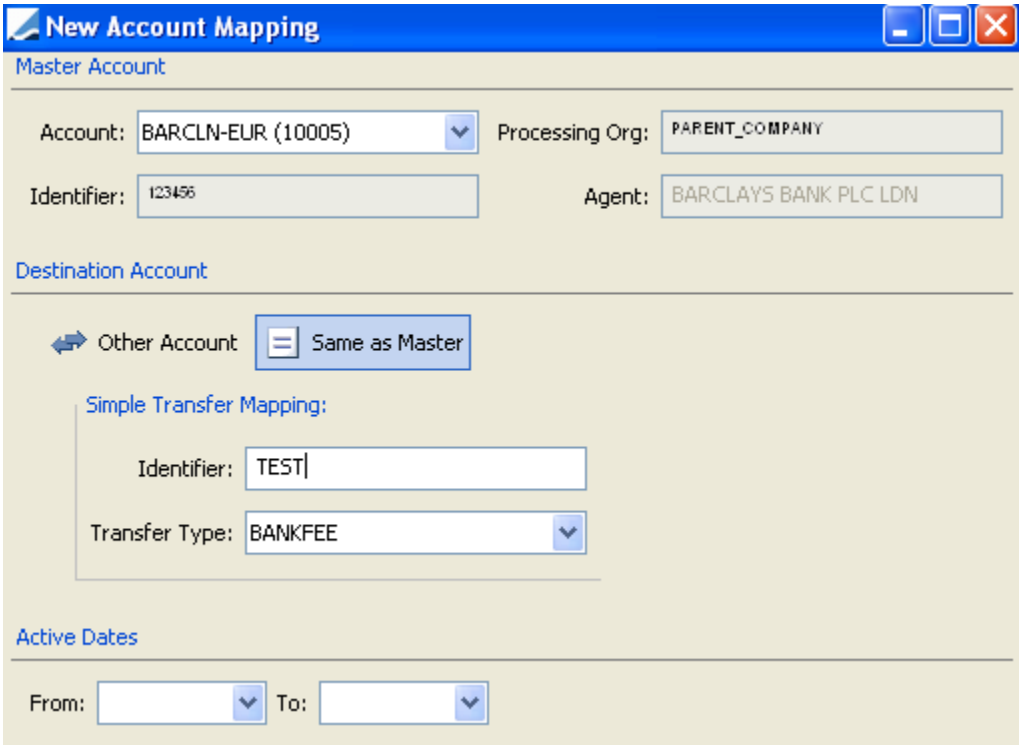
To do so, in the Destination field of the mapping table, press

and select – from the Account Browser – an account belonging to another PO than the ‘Master Account’. Fill the identifier field and Save.

Once receiving the MT940s, the system will create two Simple Transfer trades, saved between the DEFAULT_BOOK set on each PO as LE Attribute and the other PO with the Role CounterParty.

To enable that process, the PO attributes and CounterParty SDIs for each processing organization must be set in your system.

Mapping for Automatic Bank Fee Booking



The above setup allows the automatic creation of simple transfer deals (type BANKFEE) for the processing organization bank account, when this fee is confirmed by the bank statement (MT940) and not pre-booked.

Step 1 "Master Account"

In red above, you must select the processing organization account for which you receive an incoming statement to process.

The selection is done by clicking on the Account field that pops up the Account Selector window. The 'Master Account' is the processing organization SETTLE Account for which we receive an incoming statement (MT940).

By selecting the account, the system defaults the grey fields (Processing Org Name, Identifier = Account Attribute XferAgentAccount, Account Agent Name) from the account static data.

Step 2 "Destination Account"

In blue above, you must select the '**SameAccount**' button, enter the identifier from Tag 61 or 86 of your incoming statement MT940 then select the Transfer Type to create. The system will therefore create a simple transfer impacting that processing organization bank account each time it will find that specific identifier value.

Step 3 "Active Dates"

In green above, you need to populate the "Active From/To" mapping Dates.

Default is blank but as mapping rules can evolve these fields allow putting a "To" and "From" Dates to the mapping rules.

1.1.4 Subsidiary Not Managed Account and Related SDI

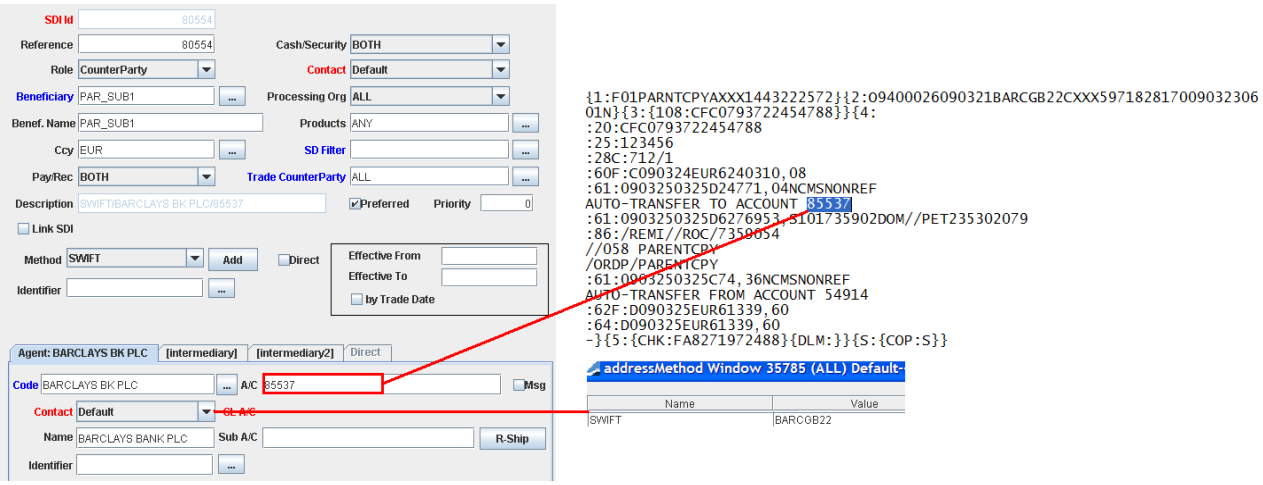
Each subsidiary must be defined with the Role = CounterParty.

CounterParty SDI can be defined for each subsidiary and sub-account. The information stored on this type of SDI can be used to map what is confirmed by the incoming MT940, provided the SDI is ticked as Preferred.

More precisely, the subsidiary not managed (sub-) account, as confirmed in Tag 61 and/or 86 of the incoming MT940, is mapped with the value saved in the field **A/C** of the CounterParty SDI for the MT940 sender (agent).

This SDI is usually automatically created from the Account Mapping window using the "Settlement Instruction" button.

An example is presented below for the subsidiary PAR_SUB1, EUR Not Managed Account at BARCLAYS BANK PLC (The Identifier from the incoming statement MT940 is "85537" which is the value set in the mapping rules for the Destination Account Identifier and the CounterParty SDI A/C field):



1.1.5 Subsidiary Payment Factory SDI

When a payment factory is used, subsidiary is not identified by its sub-account but by a specific code (eg. Kapis).

We can store that identifier in the field **A/C** of the CounterParty SDI but this information (only used for statements) is not really useful.

That's why, for Payment Factory Cash Pooling, we do not advise to create the related Counterparty external SDI.

Instead, **a generic SDI manually defined for the processing organization with the Role CounterParty, Agent = Processing Organization, Currency = Payment Factory Account Currency**, will be picked by the system to create the automatic cash pooling deals for payment factory accounts.

This generic SDI could also be used for standard cash pooling if you do not require the not managed accounts to be saved in the Calypso Static Data Referential.

1.1.6 SetBOValidationDate Trade Workflow Rule

The booking date confirmed by the statement for cash pooling entries is automatically populated as the transfer booking date.

For other internal transactions (intercompany settlements and in-house market transactions), the booking date to confirm on the internal account statement is set – on DDA/Client transfers only – by the transfer engine with today's date for back value movements.

The user can still change that 'default' booking date using the trade workflow rule SetBOValidationDate.

This rule compares the 'validation date' (= action date when we apply the SetBOValidationDate rule) stored as a trade keyword with the value date of all related cash flows for that trade:

- If cash flow value date >= validation date (action) then transfer booking date = value date
- If cash flow value date < validation then booking date = validation (action) date

1.1.7 Additional Static Data

The additional following static data are used for the reconciliation process and are populated by the default upgrade script provided by Calypso. If the script has not been executed, you will need to add them to use the reconciliation process.

Message Attribute

CashStatementProcess which indicates the type of "substatement" for the sub-MT940 message automatically split from the global MT940.

Trade Keywords

- CashStatementProcess which indicates the type of trade created automatically from the statement;
- CrossCheck (TransferAgent only) which indicates that both legs of the trade (two incoming swifts) have been received;
- TradeSource (SimpleTransfer only) populated to CashStatement and used in SimpleTransfer SDI selection to select the correct processing organization SDI from the statement;
- TargetAccountId (SimpleTransfer only) indicating the GL Account Id for the processing organization, derived from the statement and used in SimpleTransfer SDI selection to retrieve the correct processing organization SDI.

Transfer Attribute

ExpectedStatus which is checked by the rule CheckToBeSettled to move the transfers marked with that attribute automatically to SETTLED. This status identifies the reconciled movements and automatically updates the settled/reconciled cash position.

1.2 Matching Framework

The reconciliation process relies on the Calypso Matching Framework.

The Matching Framework being part of a module, you'll first need to build and deploy that module.

To do so,

- Get the matchingrel.jar (or build it: type ant matching or bld matching from you calypso directory). You will create a matching.jar and a matchingx.jar in the main calypso jars directory. The matching.jar contains the core Matching Framework. The matchingx.jar contains the calypsox part (sample of rules to create a workflow and sample of SwiftFormatter to generate Chaser).

These jars need to be added, if not already, in your classpath for the functionality to be accessible.

- If matching windows are not defaulted in **Configuration > Message & Matching**, add the windows linked to the matching process through **Utilities > Main Entry Customizer** thanks to the following actions:
  - Matching Alias Definition: `refdata.MatchingAliasWindow`
  - Matching Context Definition: `refdata.MatchingContextConfigurationWindow`
  - Matching Window: `reporting.MatchingUILauncher Matchable`

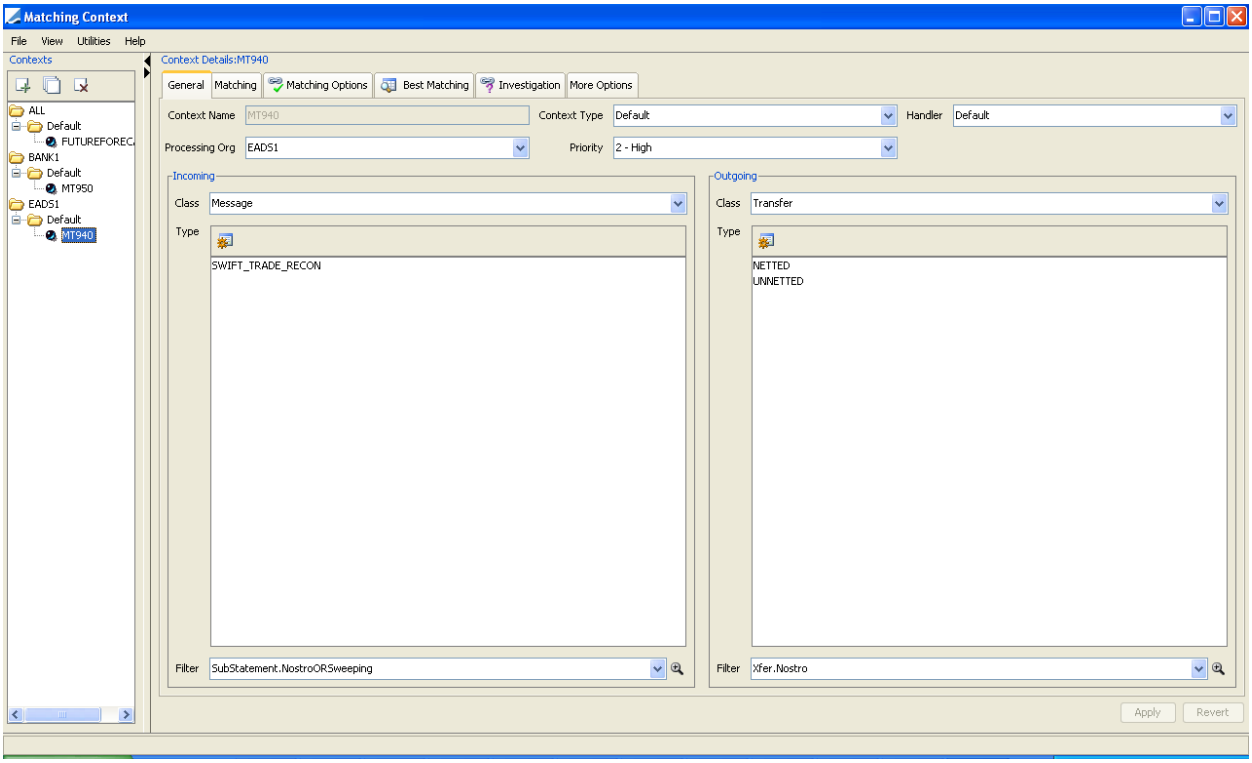
  To use a specific template, you can use the following action: `reporting.MatchingUILauncher Matchable <template name>`.
- Add the necessary Engines:
  - MatchableBuilderEngine: `apps.startup.StartMatchableBuilderEngine`
  - MatchingEngine: `apps.startup.StartMatchingEngine`

1.2.1 Matching Context for Cash Management

Matching Context MT940 versus Transfers

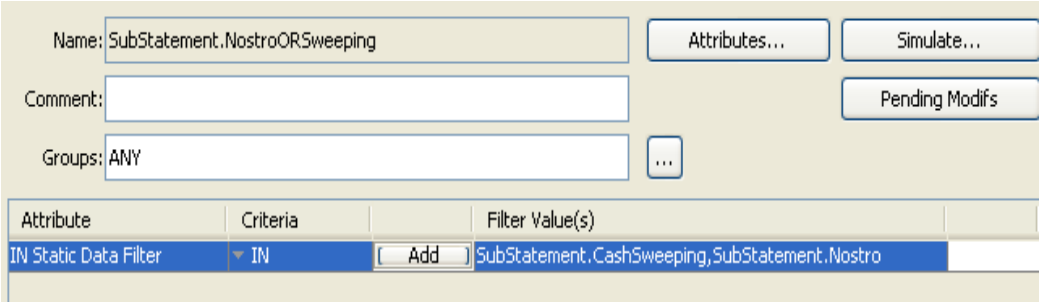
To handle the matching between the 'incoming' MT940 sub-statement and the 'outgoing' transfer, we need to add a new context type 'MT940'.

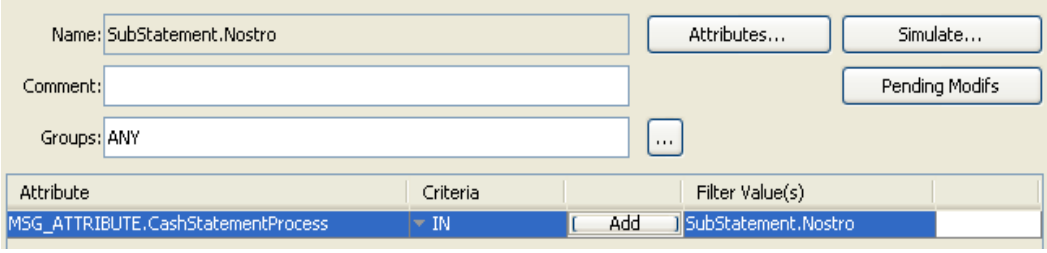
To do so, go to **Configuration > Message & Matching > Matching Context** and, in the General panel, add a new context named (for instance) MT940 as shown below:

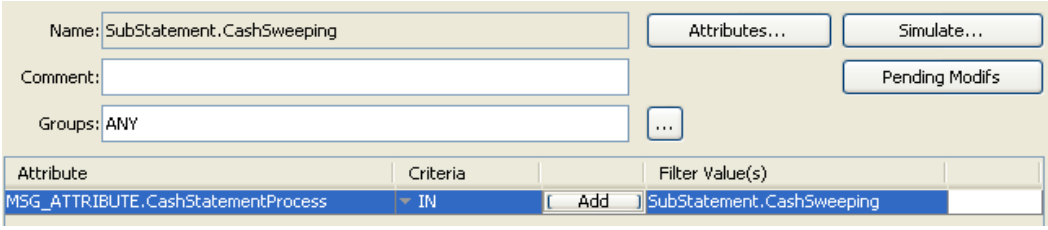


Set the General context elements as follows:

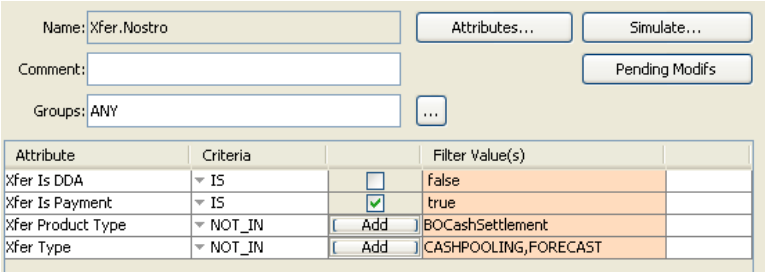
- Context Type = Default;
- Handler = Default;
- Processing Org. = ALL (Context can also be defined by Processing Organization for complex organizations);
- Priority = High;
- **Incoming Class = Message** (we want to match incoming MT940 sub-statements saved into Calypso as BO_MESSAGE Type = SWIFT_TRADE_RECON);
  - **Type =** choose the sub-statement message type **SWIFT_TRADE_RECON;**
  - **Filter =** define a Static Data Filter to only select sub-statement marked with **CashStatementProcess = Statement.Nostro or SubStatement.CashSweeping**, as shown below:



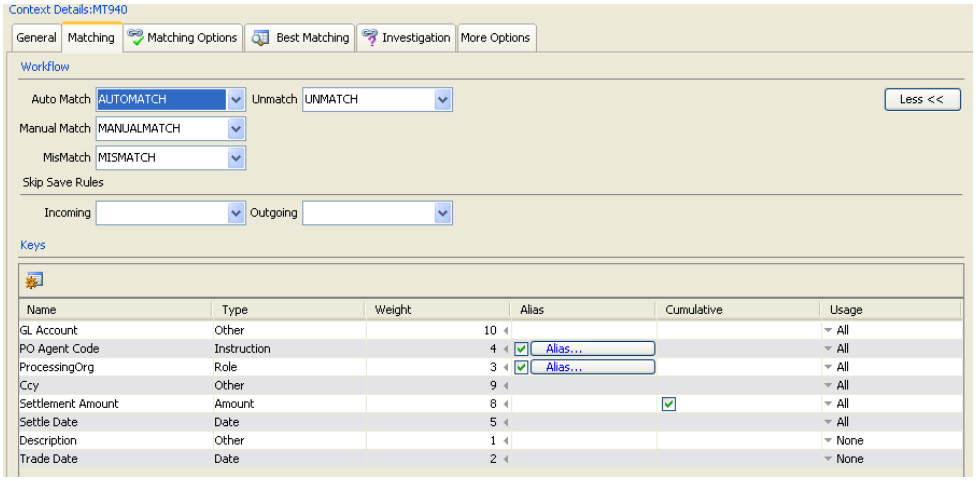




- Outgoing Class = Transfer (we want to match incoming MT940 sub-statements saved into Calypso with the Calypso "Nostro" Transfers, including CASH SWEEPING transfers);
  - Type = choose the transfer types; the transfer types are derived from your transfer workflow subtypes. By default, Calypso proposes NETTED/UNNETTED but there is the possibility through SD Filters to add new workflow transfer types. Please make sure to select the appropriate ones.
  - Filter = define a Static Data Filter to only select transfers which are not client, forecasts and/or cash pooling; the system must only perform automatic matching for 'nostro' transfers versus incoming sub-messages and must exclude subsidiary internal transfers impacting the subsidiary call account. You must also exclude from the matching the transfers type FORECAST which impact the nostro but that must be reconciled through another context with CASHPOOLING. No need to reconcile them with sub-message. You must also exclude transfers associated to the 'BOCashSettlement' technical deal (see 1.3).

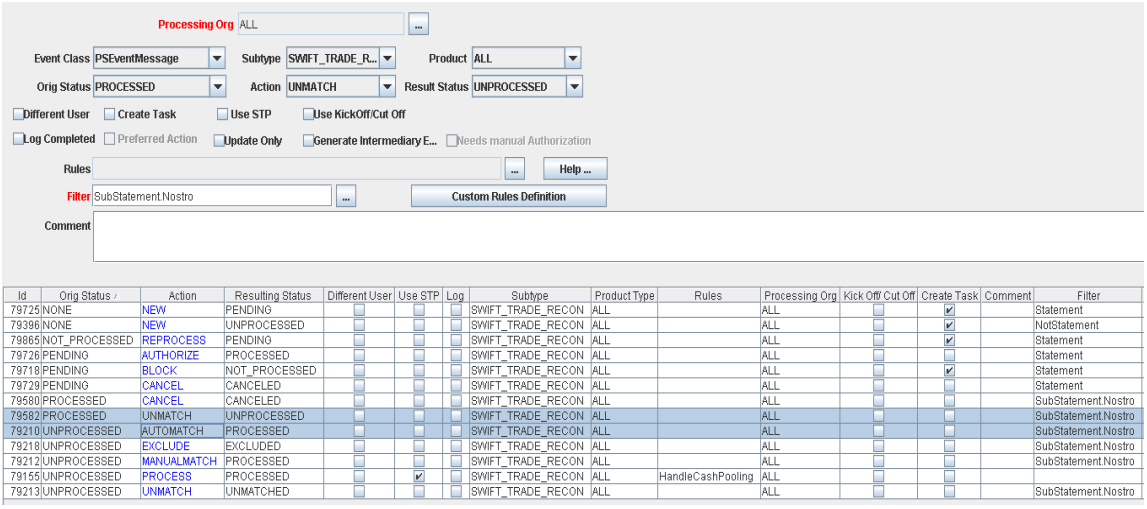


Set the Matching panel as follows:



» Define the underlying message (type SWIFT_TRADE_RECON) and transfer (nostro) workflow **actions** that the system will apply on these objects when matching automatically (AUTOMATCH), manually (MANUALMATCH), de-matching (UNMATCH) or mismatching (MISMATCH). These **actions must be available as message and transfer actions**.

Below an example of the related SWIFT_TRADE_RECON Message Workflow:

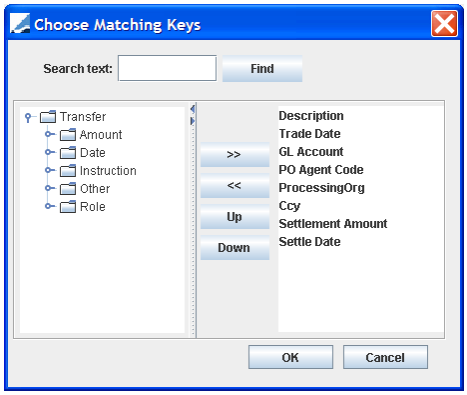


Id	Orig Status	Action	Resulting Status	Different User	Use STP	Log	Subtype	Product Type	Rules	Processing Org	Kick Off/ Cut Off	Create Task	Comment	Filter
79725	NONE	NEW	PENDING				SWIFT_TRADE_RECON	ALL		ALL		☑		Statement
79396	NONE	NEW	UNPROCESSED				SWIFT_TRADE_RECON	ALL		ALL		☑		NotStatement
79865	NOT_PROCESSED	REPROCESS	PENDING				SWIFT_TRADE_RECON	ALL		ALL		☑		Statement
79726	PENDING	AUTHORIZE	PROCESSED				SWIFT_TRADE_RECON	ALL		ALL				Statement
79718	PENDING	BLOCK	NOT_PROCESSED				SWIFT_TRADE_RECON	ALL		ALL		☑		Statement
79729	PENDING	CANCEL	CANCELED				SWIFT_TRADE_RECON	ALL		ALL				Statement
79580	PROCESSED	CANCEL	CANCELED				SWIFT_TRADE_RECON	ALL		ALL				SubStatement.Nostro
79582	PROCESSED	UNMATCH	UNPROCESSED				SWIFT_TRADE_RECON	ALL		ALL				SubStatement.Nostro
79210	UNPROCESSED	AUTOMATCH	PROCESSED				SWIFT_TRADE_RECON	ALL		ALL				SubStatement.Nostro
79218	UNPROCESSED	EXCLUDE	EXCLUDED				SWIFT_TRADE_RECON	ALL		ALL				SubStatement.Nostro
79212	UNPROCESSED	MANUALMATCH	PROCESSED				SWIFT_TRADE_RECON	ALL		ALL				SubStatement.Nostro
79155	UNPROCESSED	PROCESS	PROCESSED		☑		SWIFT_TRADE_RECON	ALL	HandleCashPooling	ALL				
79213	UNPROCESSED	UNMATCH	UNMATCHED				SWIFT_TRADE_RECON	ALL		ALL				SubStatement.Nostro

And the same related actions to be available on the Transfer Workflow:

79585	CANCELED	UNMATCH	CANCELED				ALL	ALL		TEST_EADS				
79444	MATCHED	UNMATCH	UNMATCHED				ALL	ALL		TEST_EADS		☑		
79584	SETTLED	UNMATCH	VERIFIED				ALL	ALL		TEST_EADS				
79466	UNMATCHED	UNMATCH	UNMATCHED				ALL	ALL		TEST_EADS		☑		
79583	VERIFIED	UNMATCH	VERIFIED				ALL	ALL		TEST_EADS		☑		
79471	VERIFIED	MANUALMATCH	SETTLED				ALL	ALL		TEST_EADS				
79453	VERIFIED	AUTOMATCH	SETTLED				ALL	ALL		TEST_EADS				

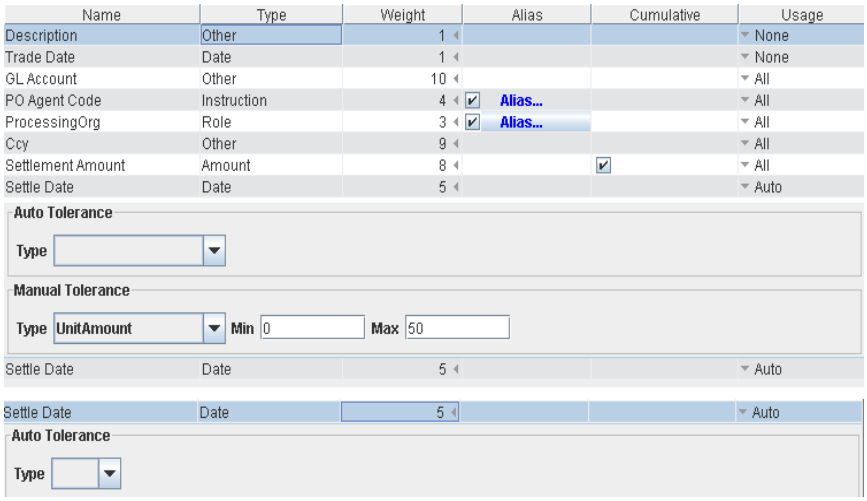
» Then define the matching Keys clicking on the Keys icon; this opens the following selection window:



» Then, for each key define their type, weigh (this also defines the order of the columns of each key in the matching monitor), alias (only available for some key such as Processing Org., Agent and must be set to true/false), cumulative aspect (only available for some keys such as Settlement Amount) and the usage (All, Auto, Manual, None);

» For some keys, we also allow the definition of Tolerance for Automatic and/or Manual Matching;

A proposed setup for the matching keys is presented below (this setup must be defined and adjusted in terms of keys, usage and tolerance rules for each implementation):



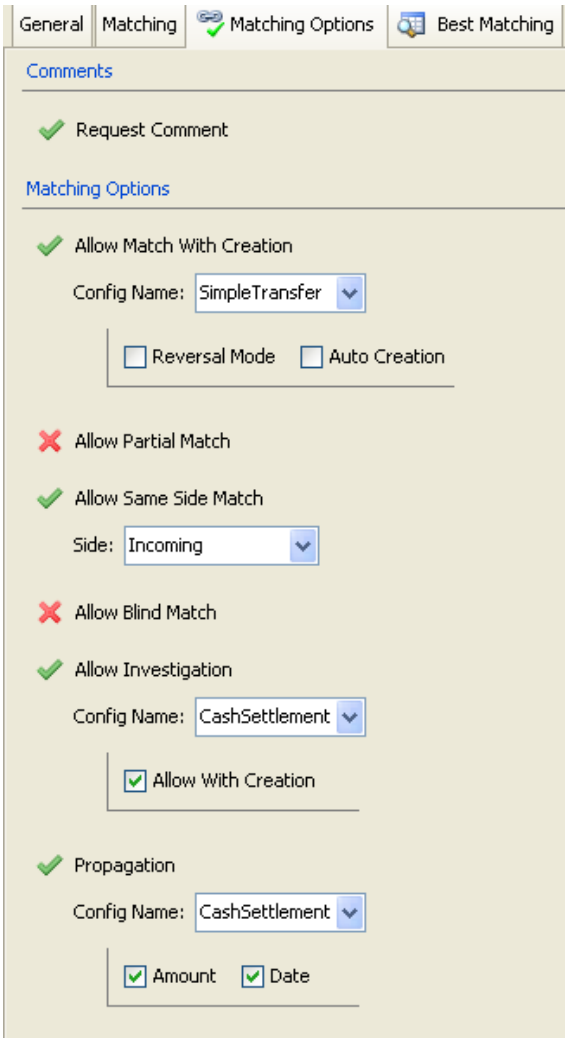
In that example:

- Description and Trade Date have been defined with Usage = None; this means these keys are there for information purpose only but are not used for matching check;
- GL Account and Ccy have been defined with Usage All meaning that we do not allow to match (automatically or manually) a sub-message and a transfer having different GL Account or Ccy;
- PO Agent Code and Processing Org. are by default aliasable and are also defined as mandatory matching keys through Usage = All. This means we do not allow to match (automatically or manually) a sub-message and a transfer having different processing organizations and agents;
- Settlement Amount is defined as Cumulative as we allow matching n sub-messages with m transfers, the cumulative sum of settlement amount being computed automatically by the system. Settlement amount is a key which also allows applying tolerance types (%, Amount, UnitPer...) that could be different for Auto or Manual Matching. In the example above, we do not allow tolerance for

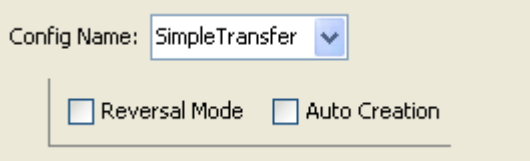
automatic matching (= strict match) but we allow a 50 UnitAmount in the transfer currency when matching manually;

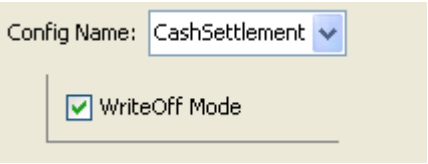
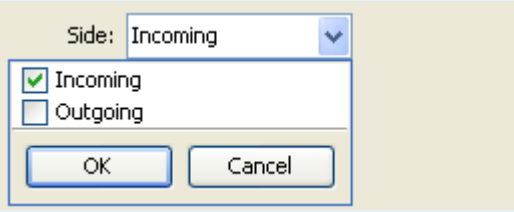
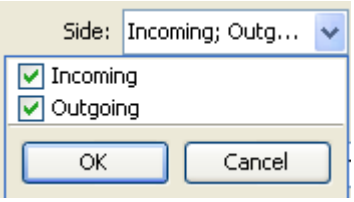
- Settle Date is defined with Usage = Auto. This means settle date is only checked for automatic matching. We allow, for manual matching, a difference between the sub-message date and the transfer.
- All these keys are user-configurable and the setup presented above is only an example.

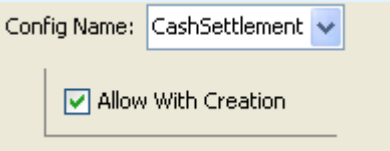
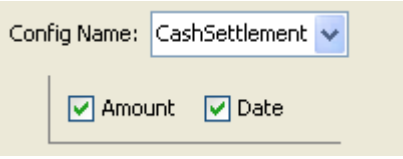
Set the Manual Matching panel as follows:



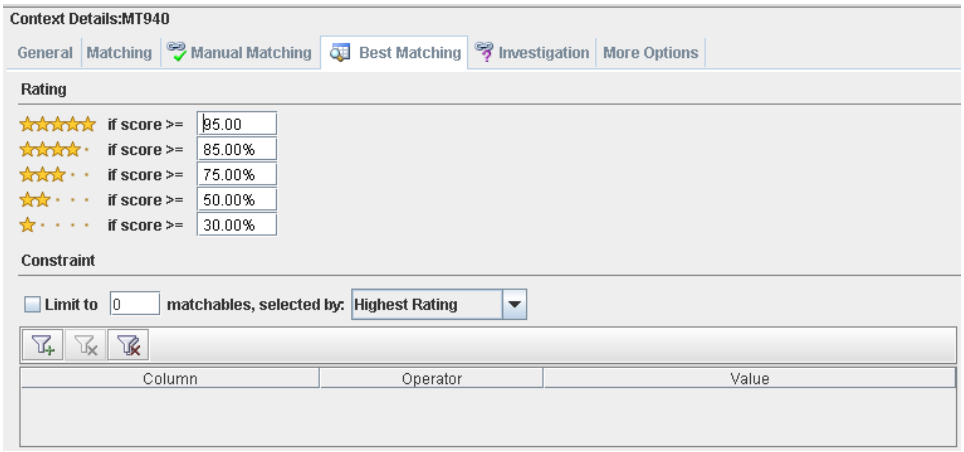
We allow the user to define the manual matching options that he can access when using the Matching Monitor to manually reconcile unmatched items:

Credit Facility / Additional Functions	
Comments	
Request Comment	If the user requests a comment to be added when manually matching an item, then he will have to select the Request Comment in that window. Calypso allows entering a text comment or attaching a file (any format) to the matching object. The comment will be displayed on the matched object window column “user comment”; the file will be displayed on the matched object window through the Show File option (right-click).
Matching Options	
Allow Match With Creation	If the user allows the manual creation of missing movement, he will have to tick the “Allow Match With Creation” in the Matching Options. This will activate a “Match With Creation” button in the Matching Monitor that will add the possibility to manually create and type a specific movement that is missing in the Nostro balance; the system will create a SimpleTransfer with Flow Type = “Type set from the Matching Monitor”.  The system allows two additional modes for the ‘Match With Creation’: Reversal Mode: this mode may be used to adjust the theoretical and actual balances when expected value dates differ from the statement/bank confirmed value date. When Reversal Mode is activated, the system will create one simple transfer per date to reverse your expected entry and a new one per date to pass that movement as confirmed by the bank. Auto Creation: when activated, that mode allows the automatic creation of a missing entry which is confirmed by the bank account statement. The system initially tries to find a matching transfer based on the matching criteria; if it cannot find any the system automatically creates a Simple transfer to impact the balance with the missing movement coming from the statement.
Allow Partial Match	A Partial Match mode can also be used. This mode allows doing a (manual) partial settlement of a transfer when only one part of the flow has been confirmed by the bank statement. The system, when manually choosing the ‘Partial Match’ function, will SPLIT the existing transfer and will create two new: one going to SETTLED for the part confirmed by the statement, another one going

Credit Facility / Additional Functions	
	to FAILED for the part remaining to be confirmed. The FAILED transfer will still appear in the pending items to reconcile in the Matching Monitor. This is based on the transfer workflow setup. Please see the annex for more details.  WriteOff Mode: when activated, that mode allows automatic (or manual) booking of the difference between your transfer and your message which is within the tolerance as a writeoff. The write off mode will not impact your position but can trigger specific postings. When applying the write off logic (means the amount difference between your transfer and message is within the settlement amount tolerance), the system automatically SPLIT the existing transfer and will create two new: one going to SETTLED for the part confirmed by the statement, another one going to WRITEOFF for the difference. Globally with the SETTLED + WRITEOFF transfer, the position remains unchanged but the WRITEOFF transfer allows booking the difference in specific accounts. This is based on the transfer setup. Please see the annex for more details.
Allow Same Side Match	When activated, the system will allow matching n items within the same category (n incoming messages/n outgoing transfers) to remove them from the list of items to reconcile (with a comment). We usually apply the Same Side Matching only on Incoming (Messages) to remove some of them from the list of items to reconcile. This function will check that the sum of amount = 0 and will only allow the matching when that condition is verified. It is allowed only within the same category. 
Allow Blind Match	This function should not be activated for cash management. It is designed for confirmation matching and allows matching without any control. 

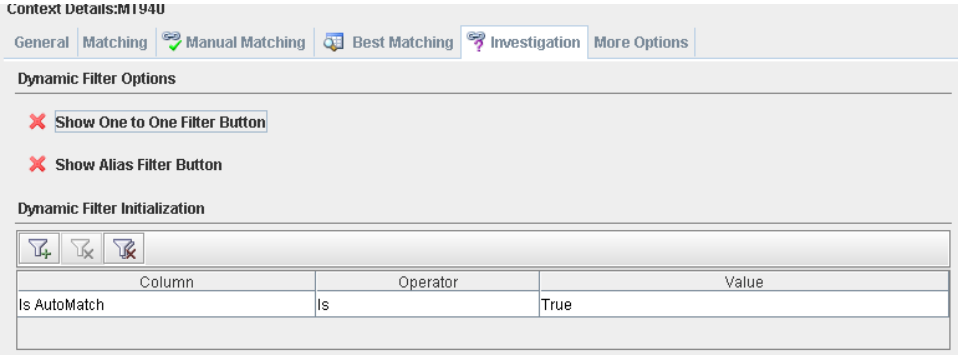
Credit Facility / Additional Functions	
Allow Investigation	Some items, needing additional investigation and/or action from a third party, cannot be reconciled quickly. In some cases, these items (either incoming message and/or outgoing transfer) need to be identified as items under investigation and users must be able to store and consult the audit/history of actions/documents...associated to them. When the mode 'Allow Investigation' is activated, the user has the possibility to link and mark these items with a specific status (eg. Investigate). He can perform action such as adding a comment, a file and keep the complete 'reconciliation' history on it. The items marked with the 'Investigate' status will still be eligible to automatic matching. It is possible to mark as Investigate (to link them) several incomings together, several outgoings together, n to m incoming/outgoing  Allow With Creation Some unmatched items are due to our errors. When a back-office user cannot receive a correction on time, he/she has the possibility to use an 'Investigate with creation' function that will impact the theoretical position only (no postings/no actual position). When activating that option in the matching context, the user will have to select the default Xfer Type that will be used on the Simple Transfer for the impact on the theoretical position only (eg. POSITION_ADJUSTMENT in the following of that document). The Investigate with creation will follow the same rules as the Match with Creation but will only be used when we are waiting for a correction from our side but need to impact the theoretical position before receiving the correction for timing issues.
Allow Propagation	This mode cannot be activated in conjunction with 'Match With Creation / Reversal Mode'. When selected (Amount and Date are ticked) the system – when automatically or manually matching message and transfer which have a difference of amount and/or date within the tolerance – will propagate as Real Settle Amount of the transfer the amount as confirmed by the bank statement and as Settle Date of the transfer the value date as confirmed by the bank statement. 

Set the Best Matching panel as follows:



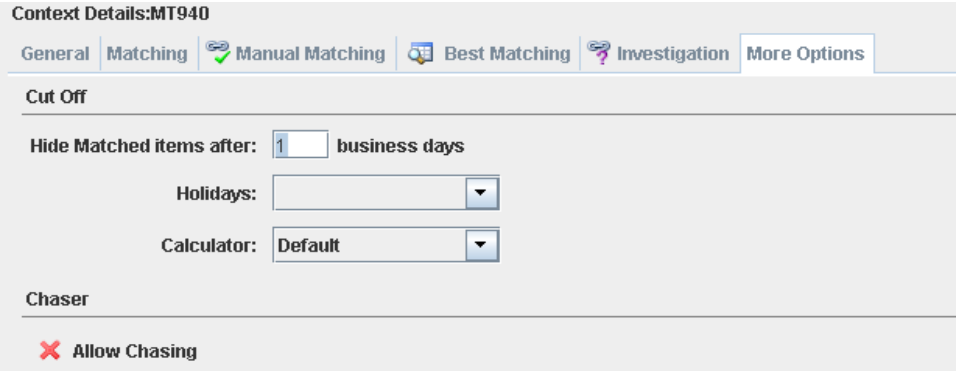
- The Best Matching part is only used for information
- No specific constraint is necessary

Set the Investigation Matching panel as follows:



- One to One Filter button and Show Alias Filter is not needed for Cash Management
- May be activated later based on end-user business practice
- Dynamic Filter must be set to IsAutoMatch Is True to select only automatchable items, as shown above

Set the More Options panel as follows:



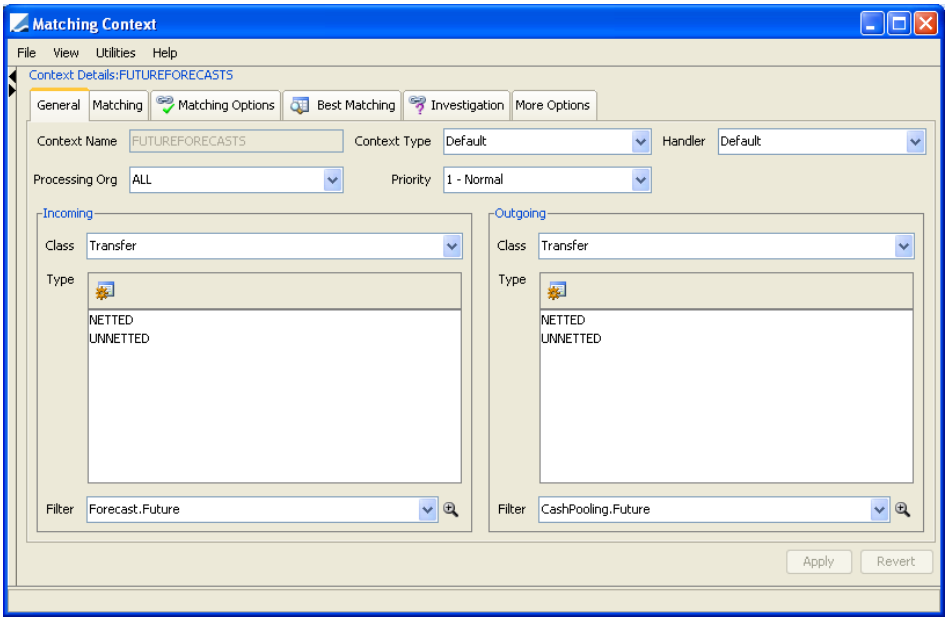
- Hide Matched Items after = number of business days to hide the matched items from the matched panel of the Matching Monitor
- The Matched panel is used to Unmatch Items
- We propose a default value of 1 to avoid polluting the matched panel with automatic or manual old matched items. This is subject to change based on end-user business practice
- Holidays = Add a calendar based on business dates to define into the system the + 1 business days to hide matched items
- Calculator must be Default
- Chaser = not activated for now for cash management

Matching Context Future Forecast (Substitution)

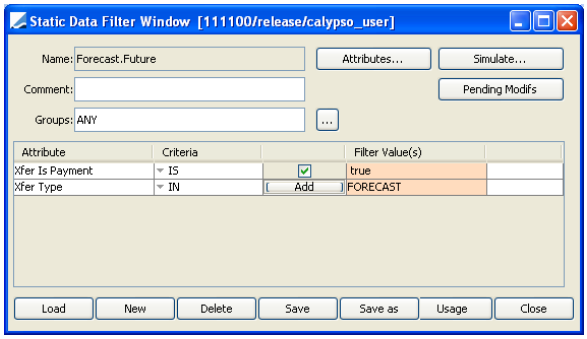
Some agents are confirming future cash pooling movements in their MT940.

We have therefore added the possibility to automatically (if 1 to 1) or manually (n to m) remove the related future dated forecasts based on the result of the matching of the future dated cash pooling entries with the forecast.

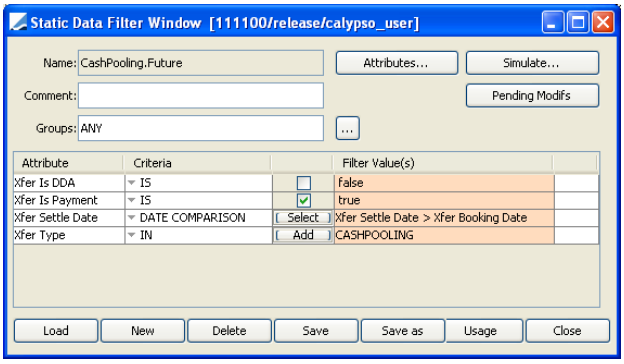
To do so, you need to configure a new matching context as shown below:



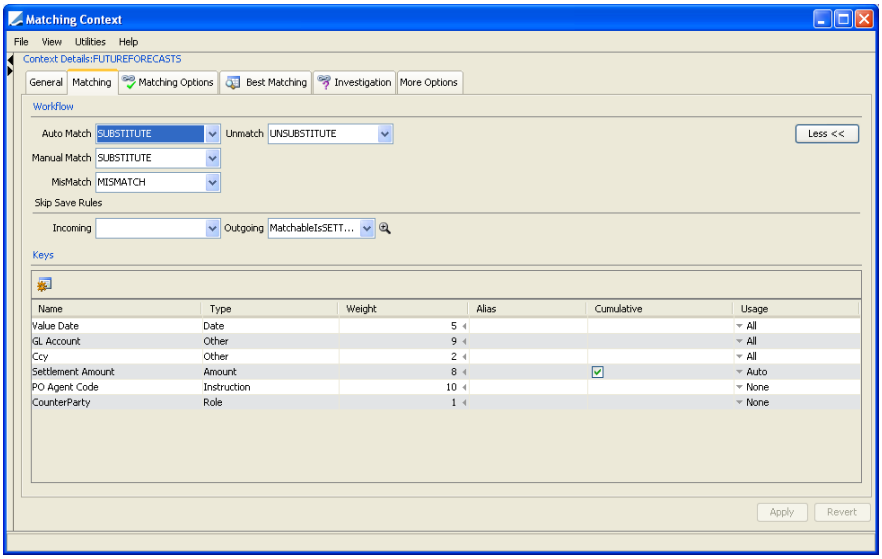
- Incoming Class = Transfer type FORECAST (we want to match transfer type FORECAST with transfer type CASHPOOLING);
  - Type = choose the transfer types; the transfer types are derived from your transfer workflow subtypes. By default, Calypso proposes NETTED/UNNETTED but there is the possibility through SD Filters to add new workflow transfer types. Please make sure to select the appropriate ones.
  - Filter = define a Static Data Filter to only select transfer type FORECAST, as shown below:



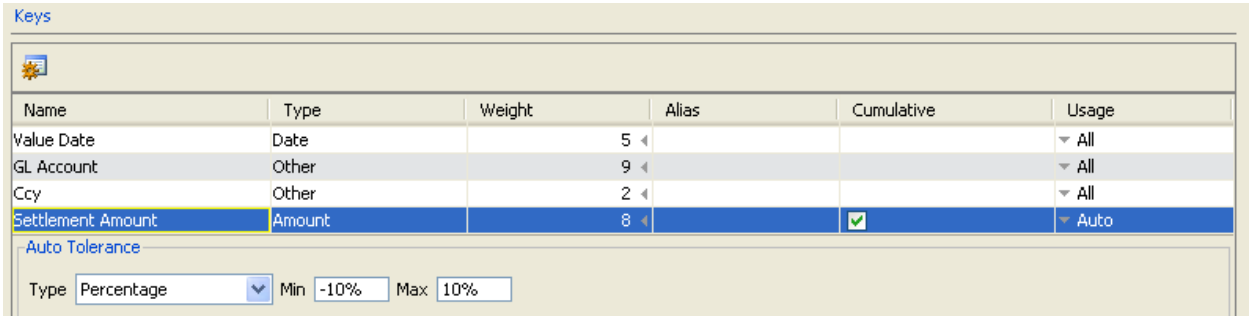
- Outgoing Class = Transfer (we want to match transfer type FORECAST with transfer type CASHPOOLING);
  - Type = choose the transfer types; the transfer types are derived from your transfer workflow subtypes. By default, Calypso proposes NETTED/UNNETTED but there is the possibility through SD Filters to add new workflow transfer types. Please make sure to select the appropriate ones.
  - Filter = define a Static Data Filter to only select CASHPOOLING transfer, nostro side (exclude the DDA/Client which does not need to be reconciled) and which are future dated (Xfer Settle Date > Xfer Booking Date), as shown below:



Then define the Matching actions and keys for that specific context, as shown below:



As forecast never matches the cash pooling amount strictly, we advise to apply a large tolerance for the automatic matching, as shown below:



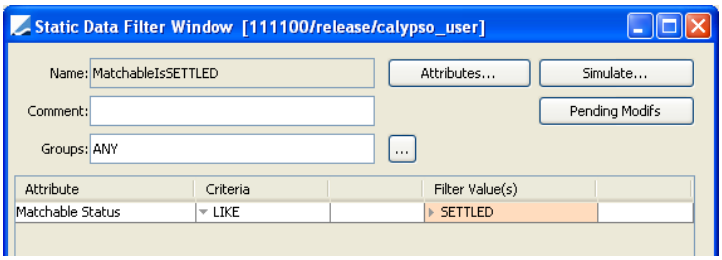
Settlement Amount is only used for Usage = Auto with a tolerance of +/- 10%. For manual matching, we do not apply any check on the amount (with that configuration).

Note that:

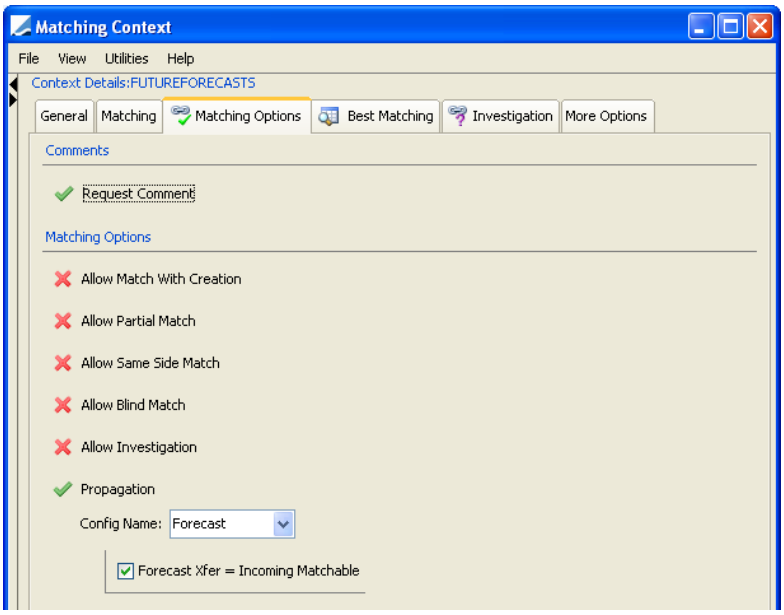
CASHPOOLING transfers, created from the incoming MT940, are automatically SETTLED.

When reconciling a SETTLED CASHPOOLING transfer with a 'not settled' FORECAST transfer, we will apply the automatch or manual match actions only on the FORECAST transfer and will skip saving that action on the CASHPOOLING transfer (the outgoing object in our example).

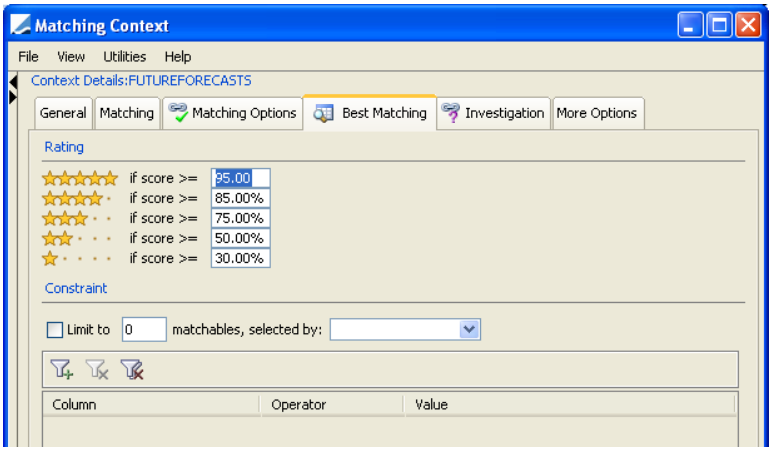
This is why you need to set – for the outgoing (=CASHPOOLING) in our example – the **Skip Save Rules** based on the below static data filter.

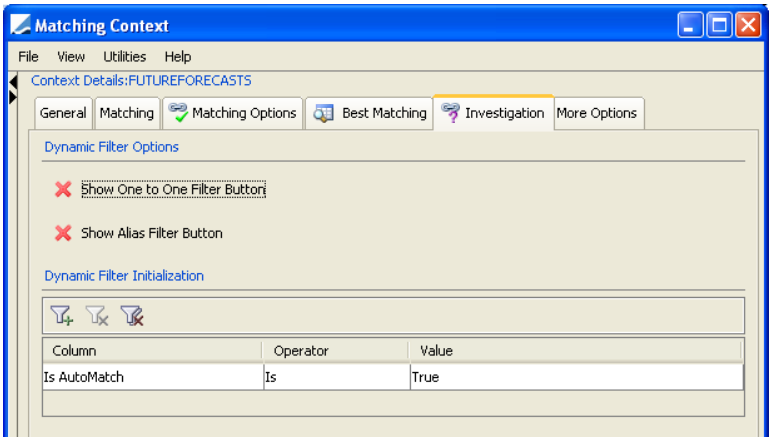


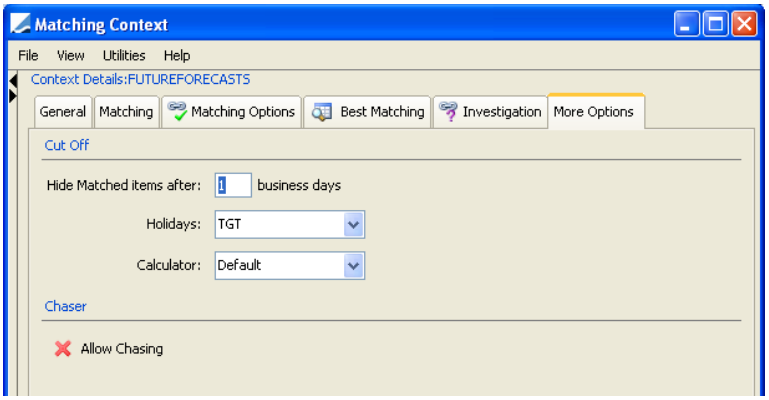
Then, define the matching options as shown below. Only Request Comment and Propagation should be activated to ensure the user enters a comment when manually matching a forecast with a cash pooling, and to propagate as real settle amount of the forecast which is reconciled with a future dated cash pooling the amount of the cash pooling transfer.



Then set the Best Matching, Investigation and More Options panel as shown below and choose Apply to save your context.







1.2.2 Matching Monitor for Cash Management

From the context “sub-MT940 versus transfer”, the **Investigate** panel gives a summary view of:

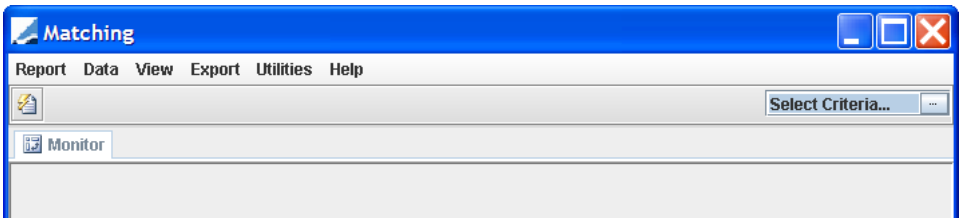
- Un-reconciled incoming sub-statements in the top-part
- Un-reconciled calypso transfers in the bottom-part

From the context “FUTUREFORECAST”, the **Investigate** panel gives a summary view of:

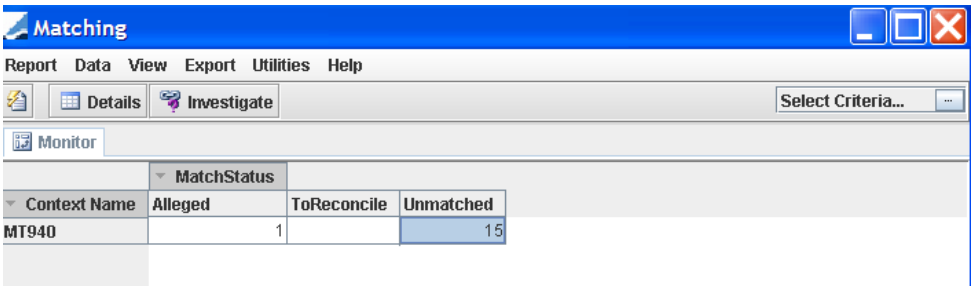
- Un-reconciled incoming Forecasts in the top-part
- Un-reconciled future Cash Pooling in the bottom-part

Matching Monitor Layout Configuration

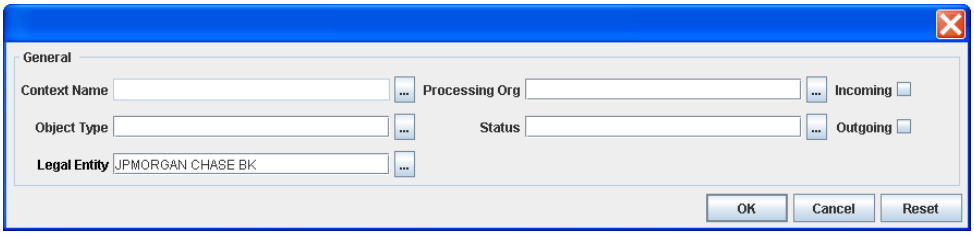
To configure the **Investigate** view, go to **Configuration > Message & Matching > Matching Window**. This opens the Matching Monitor as shown below:



Then press the button on the left-hand side of the screen to load the "matchable" objects, as shown below:

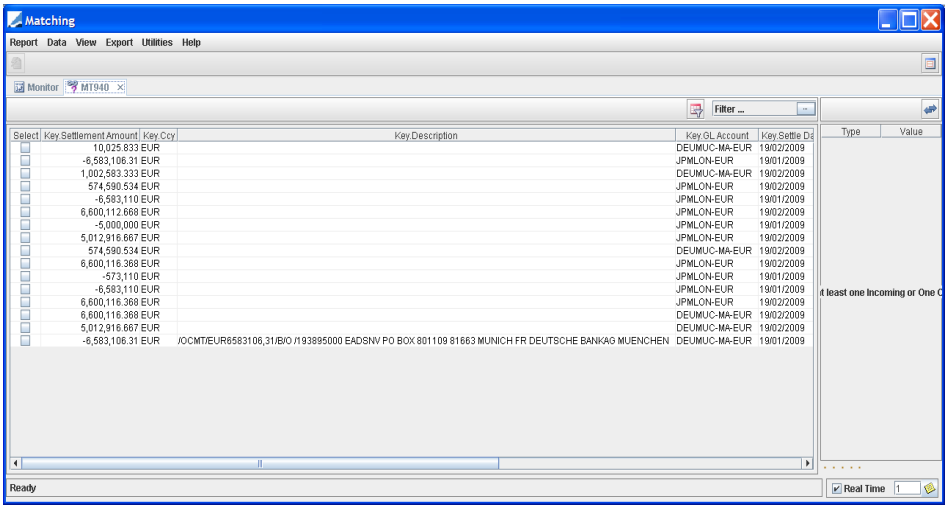


In addition, you can use the 'Select Criteria' to get a synthetic view of un-reconciled objects for specific criteria (e.g. one specific Agent only).



Click the **Investigate** button for a specific context to get the default view of **all** 'packed' un-reconciled items. Then, split and filter your table using the **Split** and **Filter** functions available in that panel.

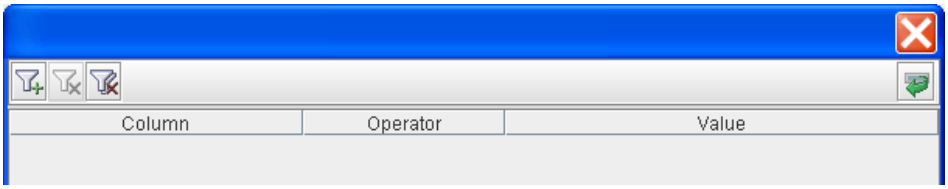
Without panel split and filter all unmatched items are mixed as shown below:



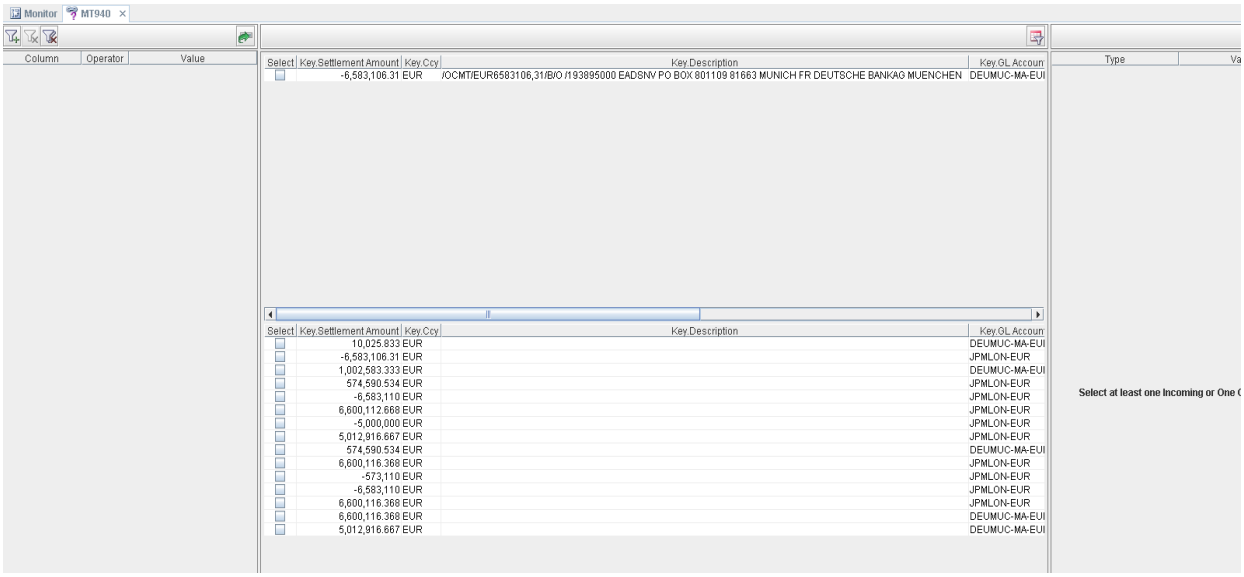
To configure the default layout of that screen, press the split panel icon . This will display the 'incoming' un-reconciled items at the top and the 'outgoing' un-reconciled items at the bottom (as shown below).

Then select the filtering icon

. This pops-up the following window:



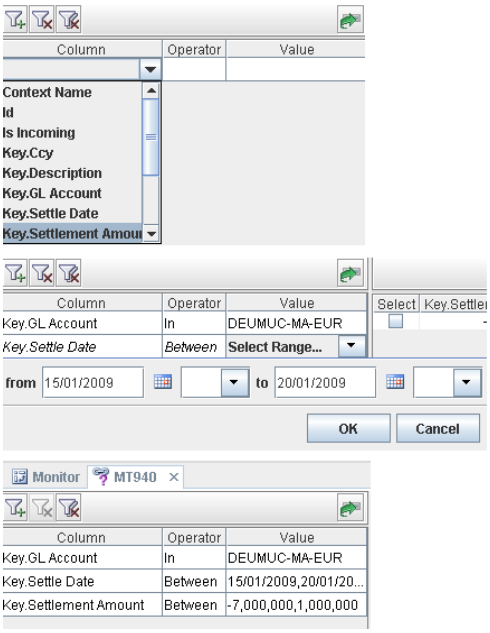
Press the icon to move the filtering capabilities as a permanent element of the Investigate panel, as shown below:



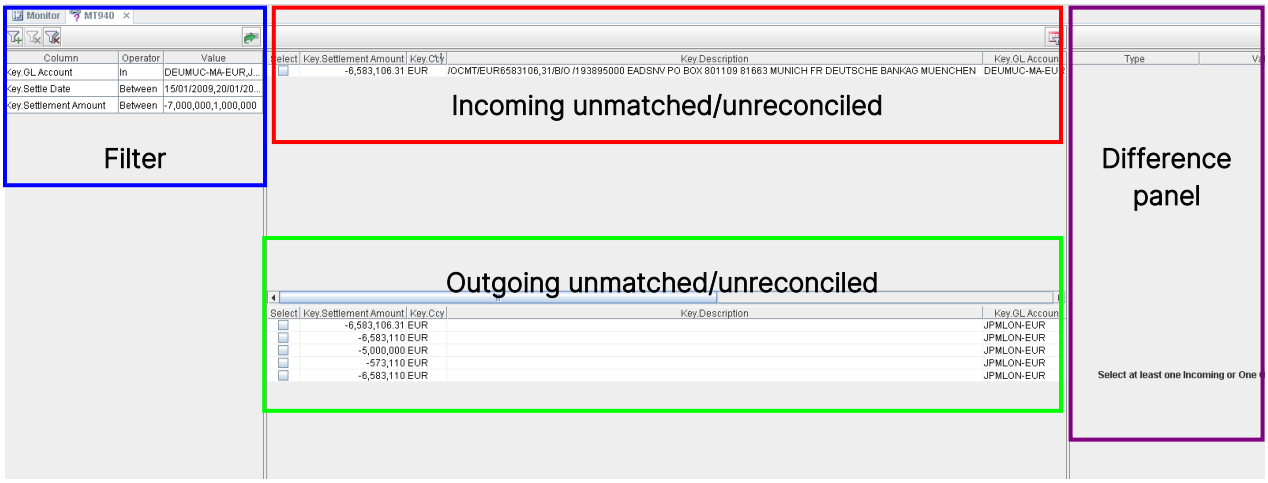
Once done, the user must **save that layout for that context** going to **Report > Save As Template** and then **define that template as the default** view using **Report > Set Default Template**.

Then, pressing the icons (green to add a filter, red to remove a filter), the user can add its own filters picking among the matching keys defined in the related matching context to limit the selection of un-reconciled items on which he has to work.

An example of filter definition is presented below:



An example of the final 'proposed' view to monitor un-reconciled items is presented below. Please note that this configuration of the investigate panel view has to be done for each context:



Same Investigate panel layout should be defined for the "FUTUREFORECAST" matching context.

Contextual Matching Actions for sub-MT940 versus transfer

From the investigate panel, the user can select and match one or n incoming sub-MT940(s) (the system compute the sum of settlement amount for the appropriate keys) versus one or m outgoing transfer(s);

The system will display all or only difference keys in the right panel (depending if the user has selected the icon which determines if we only show the difference or all keys);

Actions proposed are contextual and based on the Manual Tolerance and Usage Set defined for each key in the matching context;

Thus, if you have set a Tolerance for Manual Matching the system allows the user the **Match** (the system does not impact the difference) or **Match with Creation** (the system impacts the difference) when one or n incoming sub-MT940(s) versus one or m outgoing transfer(s) are selected:

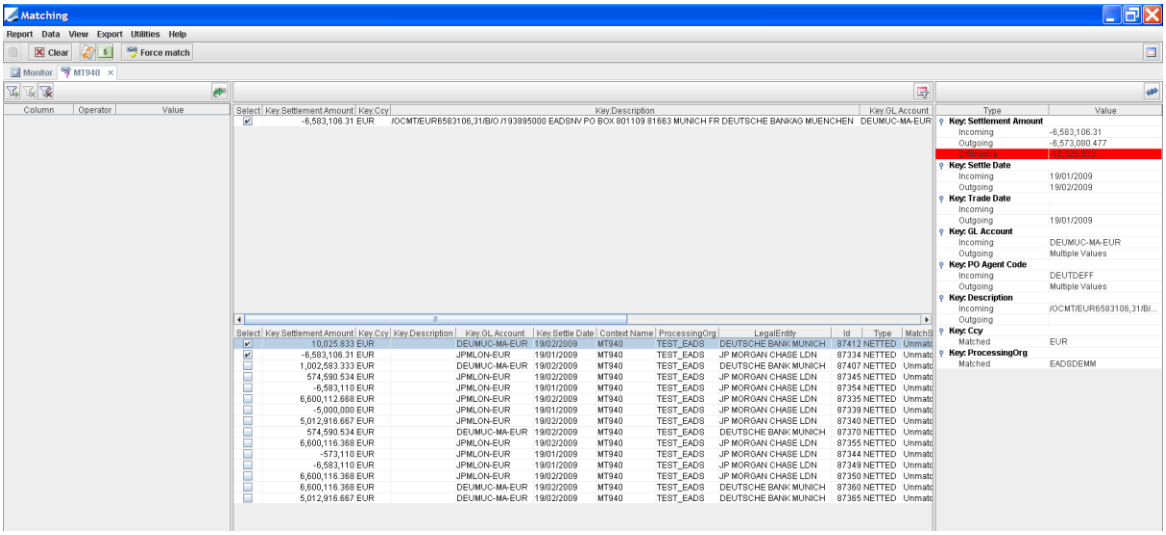
- When choosing **Match**, the system asks for a comment to be added on the Manual Match (if requested in the context), applies the MANUALMATCH workflow action on the underlying message(s) and transfer(s) but does not impact the difference on the Cash Balance;
- When choosing **Match With Creation**, the system asks for a comment to be added on the Manual Match (if requested in the context), applies the MANUALMATCH workflow action on the underlying message(s) and transfer(s) and creates an additional deal type **SimpleTransfer** to impact the difference to the Cash Balance;
- When creating the difference, the system asks the user to choose the Transfer Type (eg. BANKFEE, INTEREST, CASH_ADJUSTMENT. This is extensible by the user);
- The **Match With Creation** is also available when one or n incoming **only** is(are) selected. In this case - where the flow is missing in the system but is confirmed by the statement line - the system allows creating and 'typing' the missing movement from the manual matcher, attaching a comment or file on that object.

If you have also, for instance, allow a Partial Match and Investigate Function, then the system will propose these actions when possible in the matching monitor.

Examples of layout & functions are presented below.

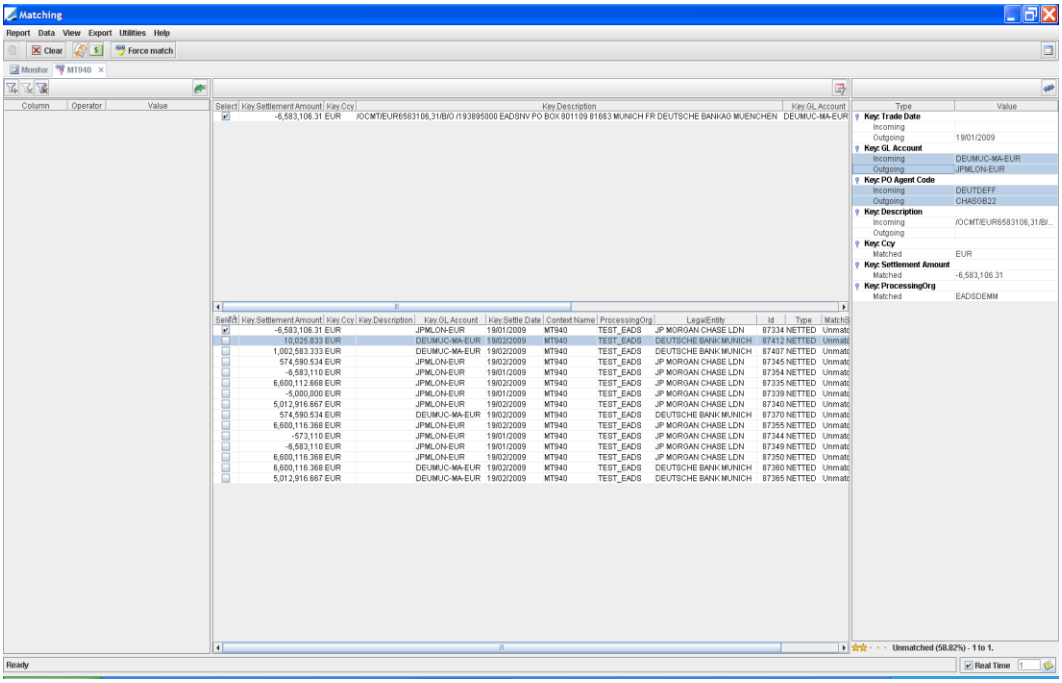
Example 1

In the example below, the difference exceeds the tolerance set on the settlement amount key and, as such, appears in red. In addition the GL Account shows 'Multiple Values', meaning different bank accounts have been selected by the user. The system does not allow matching different accounts and therefore only proposes a **Force Match** action. Please note that the Force Match can be de-activated through access permission.



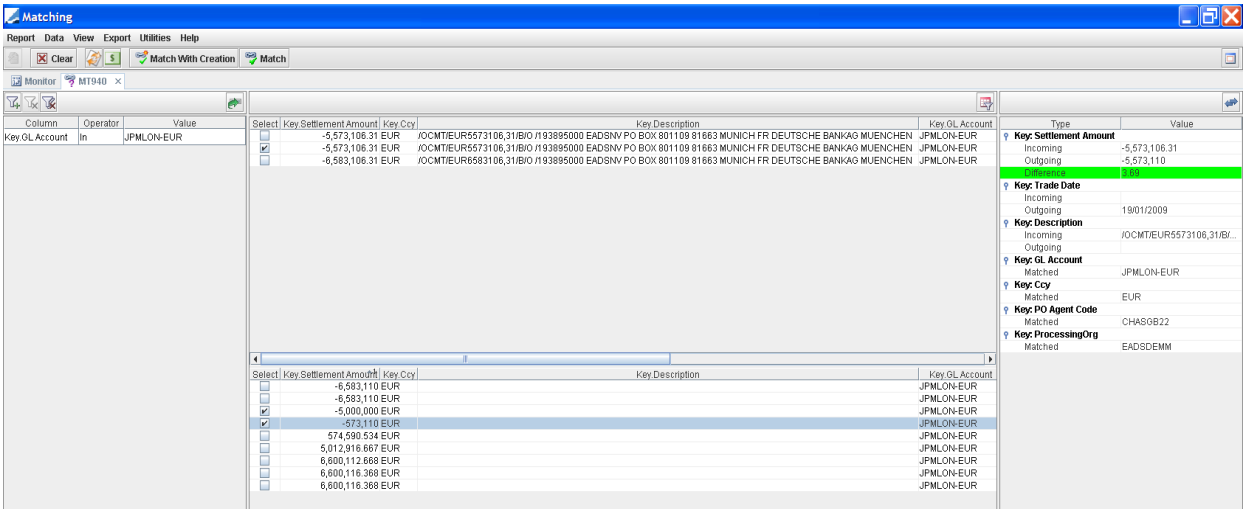
Example 2

In the example presented below, the amount matches but the GL Account and Agent are different. There again, the system does not allow matching different accounts/banks and therefore only proposes a **Force Match** action. Force Match action is subject to an access permission as only few persons in the organization should be allowed to process a ForceMatch.

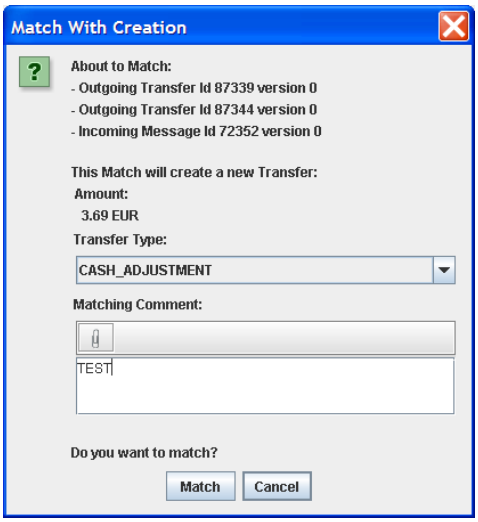


Example 3

In the example presented below, the user has activated the filter to only display un-reconciled items for a specific Bank (GL) Account thanks to the filter functions available/displayed on the left-hand side of the panel. For the selected items (incoming versus outgoing) the system computes and shows the amount difference for the selected items. As it falls within the tolerance set in the context for the Settlement Amount key, the system proposes a **Match** or **Match With Creation** functions.



If the user chooses the Match With Creation, the system pops-up the following window to enter a comment (if requested), and/or attach a file and define the Transfer Type to create to impact the balance (CASH_ADJUSTMENT in our example):



In this example, when clicking **Match With Creation**, the system links the incoming message with the two outgoing transfers selected + the new transfer associated with the Simple Transfer trade created to impact the difference. Thus, in that example matching is done between 1 incoming sub-message and 3 outgoing transfers.

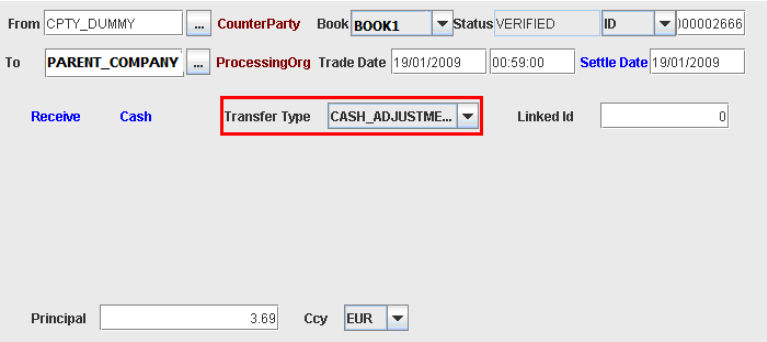
The simple transfer is created as follows:

- Transfer Type = CASH_ADJUSTMENT (selected by the user)

- Specific Trade Keyword are automatically set by the system to select the right SDI and Account, and filters that types of 'adjustment' trades

Name	Value
TargetAccountId	10000210
TradeSource	SimpleTransferMatchWithCreationHandler

- Transfers associated to the trade go directly to SETTLED to impact both the theoretical and actual positions



The system refers to specific processing organization **attributes** to create this type of SimpleTransfer:

- DEFAULT_BOOK = "YourDefaultBook" for that processing organization (BOOK1 in our example);
- DEFAULT_COUNTERPARTY = "YourDefaultCounterParty" for that processing organization (CPTY_DUMMY in our example).

Please define these attributes on your processing organization.

Depending on the matching actions and tolerance on matching keys set in the Matching Context, other actions will be **contextually** proposed by the system. These actions are:

- Match
- Match With Creation
- Partial Match
- Investigate
- Investigate With Creation
- Same Side Matching

Drilldown Functions

From the Investigate panel, the user can always drill-down:

- For incoming messages to the underlying BO_MESSAGE and Advice Document;
- For outgoing transfers to the underlying trade and related transfers.

When a file is attached, we store the file as a Generic Comment accessible from the trades(s) and matchable object.

Matching Information

Matching information is also available from the **BO Trade Window**, where we have added a **Matching panel**.

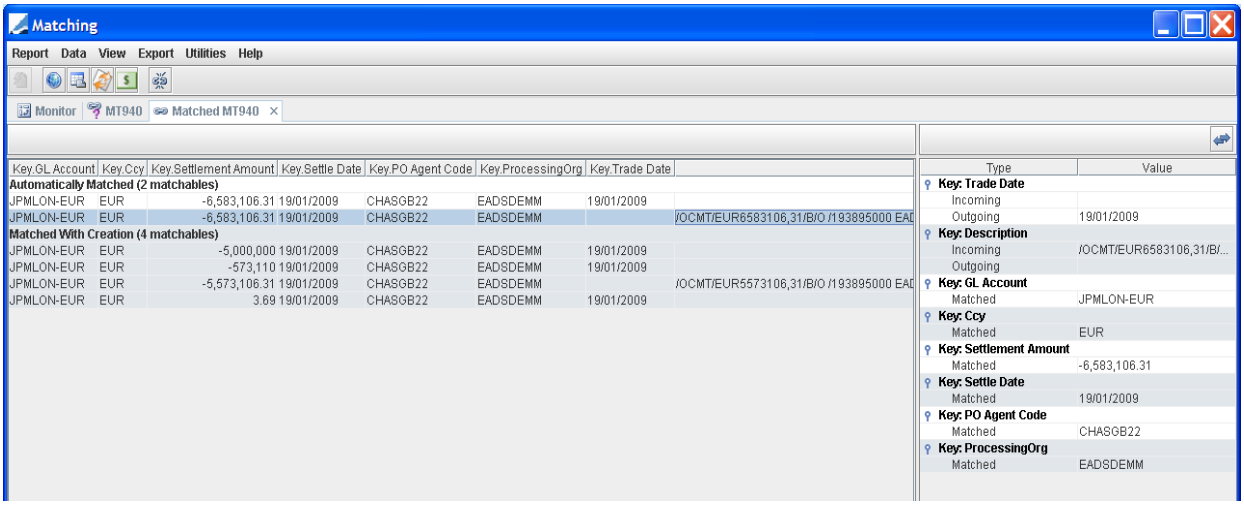
This panel displays the matching information with the same level of details as what is available in the matching monitor.

This information is available for a 1 to 1, 1 to n, n to 1, n to m matching. For n to m matching, all BO Windows for transfers (& related trades) involved in matching display the same global matching information for the matching group.

Unlink Wrong Reconciliations

This layout is also available in the **Linked** Panel of the Matching Monitor to be able to check the matched objects. The user will have to go to the Linked panel of the Matching Monitor to un-link wrong matching.

- The Linked panel is accessible from the Monitor panel;
- From a specific context, righ-click the “Linked” function;
- This pops-up the following layout with the possibility, using the icon, to un-match wrongly matched items.
- There again, the same drill-down functions exist and a specific default template must be defined to get the best possible view.



Unlinking matched objects is allowed if you have set the UNMATCH action in your Transfer and Message workflow, behind the SETTLED (transfer) and MATCHED (Message) status. If you want to unlink a Match With Creation (for which a Simple Transfer would have been created), you would also need to CANCEL to transfer (settled) associated to that simple transfer. This could be done by adding the CANCEL action on your transfer workflow after the SETTLED status, but limiting that CANCEL to that type of Simple Transfer (marked with a specific keyword) with a static data filter.

1.3 BO Cash Settlement trade and Automatic Cash Posting from Bank Account Statement (MT940)

We have added the possibility to automatically book the cash settlement postings from the bank account statement (MT940) regardless of the reconciliation status.

This allows having a cash account in our books matching what the bank has confirmed.

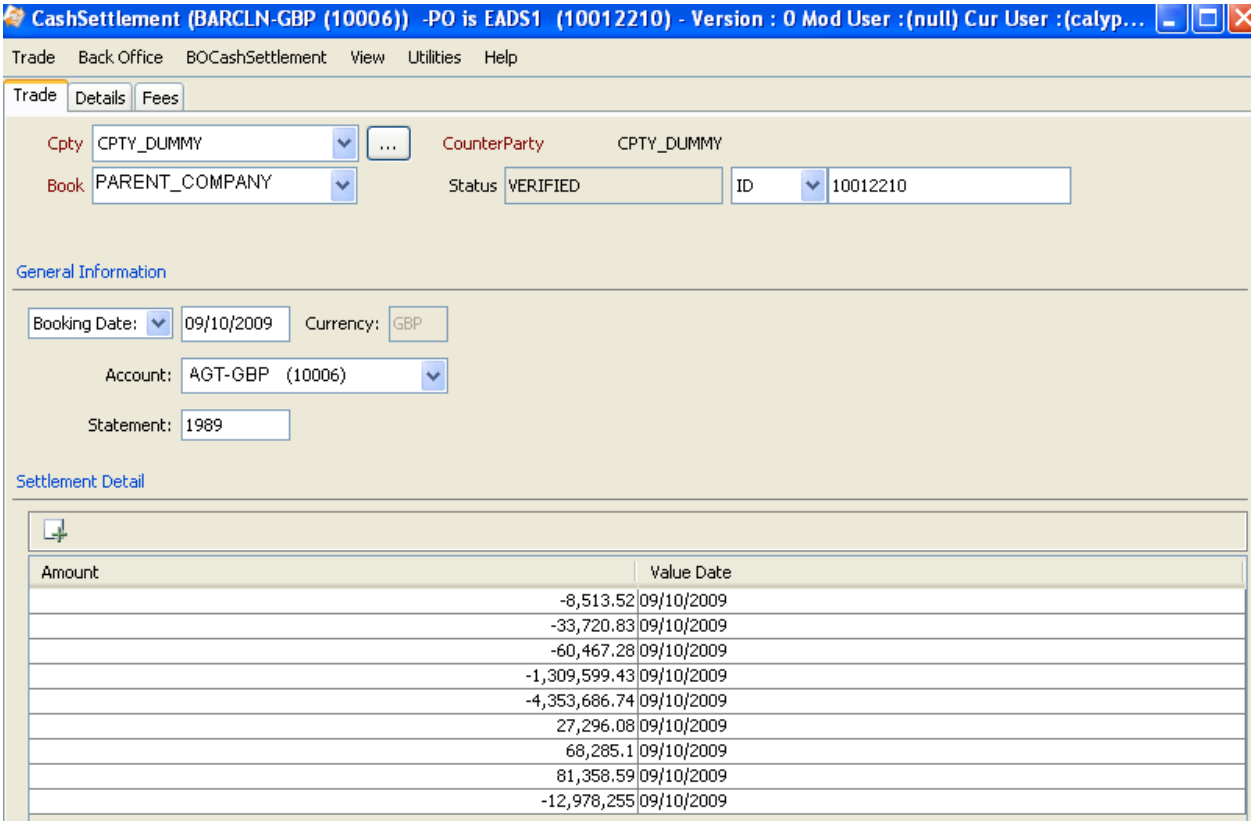
A specific Suspense Account – per Agent or Agent/Account – would be also fed by this posting to isolate the differences between my expected cash and my confirmed cash.

This type of accounting event is only optional. Depending on your business practice you may or may not use it.

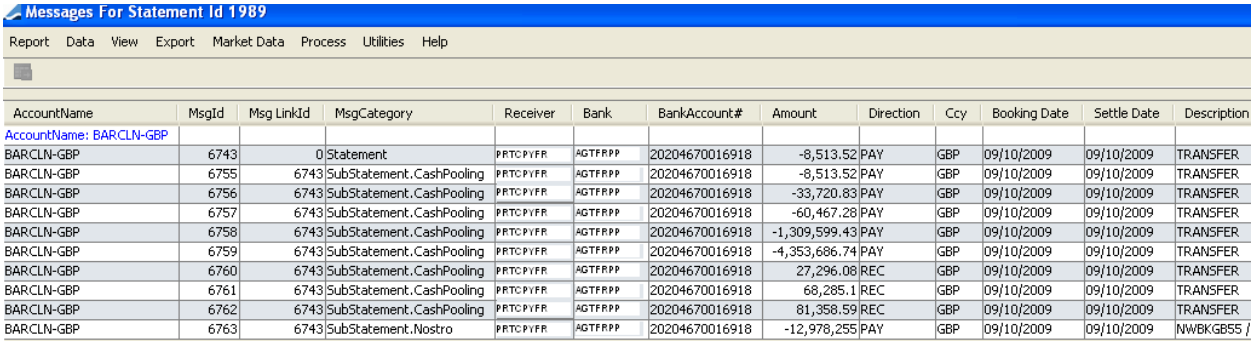
The process is the following:

- The Scheduled INC_CASH_STATEMENT – in addition to perform consistency checks, create automatic deals (cash pooling, cash sweeping, etc.) also creates a technical deal in the system on a product named BOCashSettlement.
- There is one BOCashSettlement trade per statement.
- The trade on BOCashSettlement saves, from a MT940, all individual movement as confirmed by the statement for the statement booking date. Potentially, a statement will confirm movements back-value or future-dated. In that case, the transfer engine will create one transfer per value date of the statement, all attached to the BOCashSettlement trade created for that specific statement/booking date.
- These transfers are created to impact an EXTERNAL – ACTUAL – SETTLE inventory cash position (only).
- This EXTERNAL – ACTUAL –SETTLE position can be used to generate the Interest Bearing Trades for Nostro Account and compare – using the daily balance as confirmed by the bank – if the expected interests match the figures confirmed by the bank.
- The BOCashSettlement trade can also trigger a BD_CST accounting event that will post the cash settlement entries as confirmed by the bank versus a suspense account (by bank account) triggered on its side by the expected cash entries from our trades/transfers. This setup is not mandatory but could be used in the case where you do not want to wait for the reconciliation to be performed before booking your cash entries, as confirmed by the bank.

Example of BOCashSettlement Trade



- BOCashSettlement is done between the Processing Organization DEFAULT_BOOK (PO Attribute DEFAULT_BOOK) and the PO DEFAULT_CPTY (PO Attribute DEFAULT_CPTY).
- This trade is created from the statement and is marked with a keyword TradeSource = CashStatement.
- The General Information part shows the Account, the related Statement Id, the Currency and the Date (Booking or Settle).
- The Settlement Detail part shows all individual movements as confirmed by the MT940 for that specific booking (=statement booking date) or settle date (= statement value date)
- The menu BOCashSettlement/ShowStatement Messages allows displaying the related MT940 'sub-entries', as shown below:



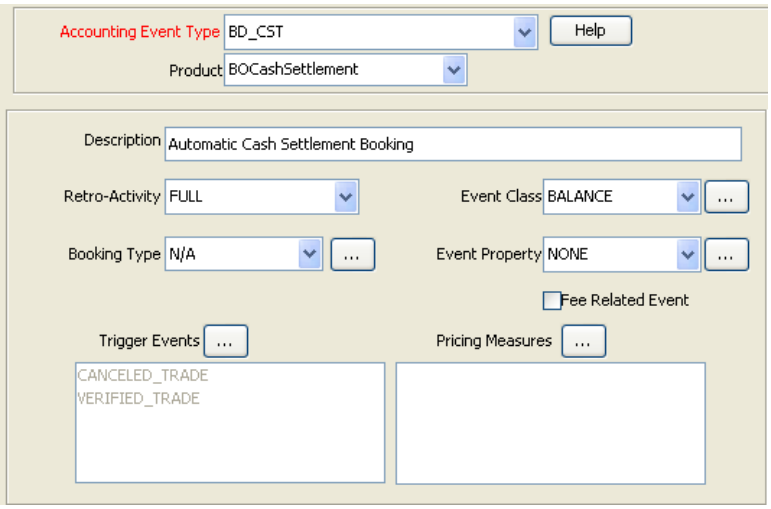
AccountName	MsgId	Msg LinkId	MsgCategory	Receiver	Bank	BankAccount#	Amount	Direction	Ccy	Booking Date	Settle Date	Description
AccountName: BARCLN-GBP												
BARCLN-GBP	6743	0	Statement	PRTCPYFR	AGTFRPP	20204670016918	-8,513.52	PAY	GBP	09/10/2009	09/10/2009	TRANSFER
BARCLN-GBP	6755	6743	SubStatement.CashPooling	PRTCPYFR	AGTFRPP	20204670016918	-8,513.52	PAY	GBP	09/10/2009	09/10/2009	TRANSFER
BARCLN-GBP	6756	6743	SubStatement.CashPooling	PRTCPYFR	AGTFRPP	20204670016918	-33,720.83	PAY	GBP	09/10/2009	09/10/2009	TRANSFER
BARCLN-GBP	6757	6743	SubStatement.CashPooling	PRTCPYFR	AGTFRPP	20204670016918	-60,467.28	PAY	GBP	09/10/2009	09/10/2009	TRANSFER
BARCLN-GBP	6758	6743	SubStatement.CashPooling	PRTCPYFR	AGTFRPP	20204670016918	-1,309,599.43	PAY	GBP	09/10/2009	09/10/2009	TRANSFER
BARCLN-GBP	6759	6743	SubStatement.CashPooling	PRTCPYFR	AGTFRPP	20204670016918	-4,353,686.74	PAY	GBP	09/10/2009	09/10/2009	TRANSFER
BARCLN-GBP	6760	6743	SubStatement.CashPooling	PRTCPYFR	AGTFRPP	20204670016918	27,296.08	REC	GBP	09/10/2009	09/10/2009	TRANSFER
BARCLN-GBP	6761	6743	SubStatement.CashPooling	PRTCPYFR	AGTFRPP	20204670016918	68,285.1	REC	GBP	09/10/2009	09/10/2009	TRANSFER
BARCLN-GBP	6762	6743	SubStatement.CashPooling	PRTCPYFR	AGTFRPP	20204670016918	81,358.59	REC	GBP	09/10/2009	09/10/2009	TRANSFER
BARCLN-GBP	6763	6743	SubStatement.Nostro	PRTCPYFR	AGTFRPP	20204670016918	-12,978,255	PAY	GBP	09/10/2009	09/10/2009	NWBKGB55 /

- BOCashSettlement trade triggers transfers which only impact the EXTERNAL – ACTUAL – SETTLE position.

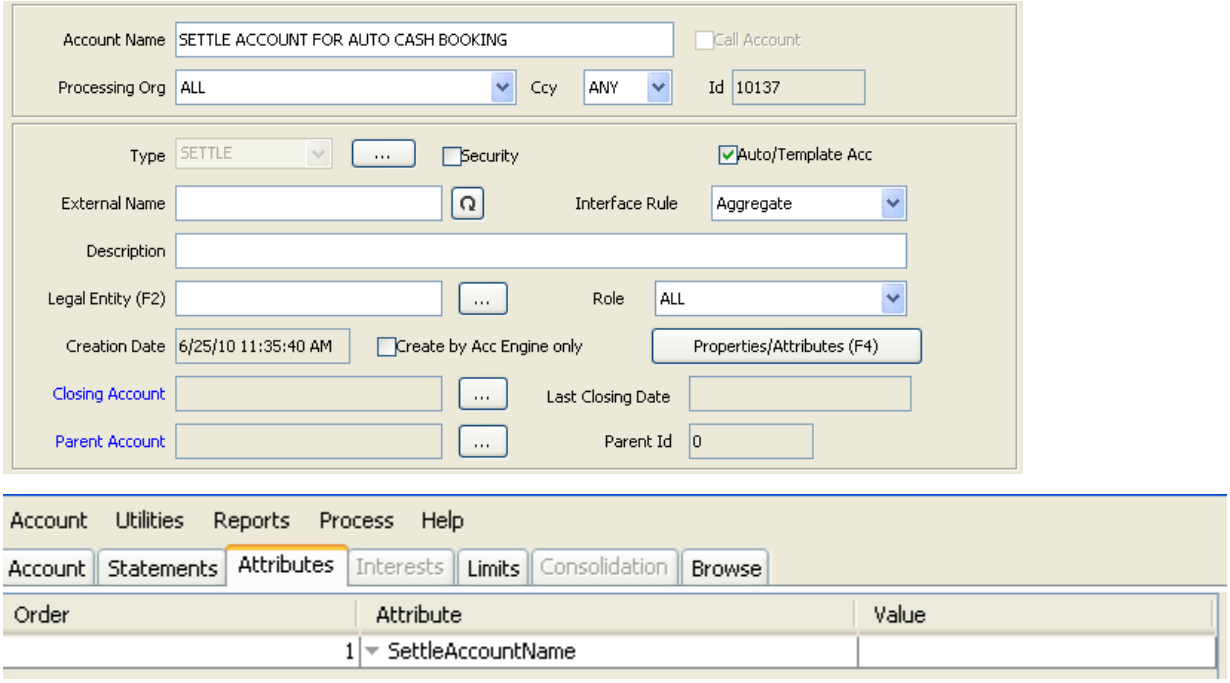
- There is one BOCashSettlement per Bank Account Statement, potentially with movements at different settle date but same booking date (= statement booking date).
- When back-value (and/or future-dated value) movements are confirmed in the same statement, the system creates only one BOCashSettlement trade for that statement but n transfer per value date for that Statement/BOCashSettlement trade.
- Transfer linked to BOCashSettlement trade must be excluded from the Nostro matching. There is no need to reconcile this type of transfer which is only created to feed the EXTERNAL position (= bank confirmed position).

BOCashSettlement Accounting Event

To book automatically the cash settlement postings as confirmed by the bank, you can use the BD_CST accounting event, as shown below:



- BD_CST is triggered by Trade events for the product BOCashSettlement.
- BD_CST will have a booking date = trade booking date and an effective date = trade settle date.
- BD_CST should impact an automatic and generic settle account created with the Attribute SettleAccountName.
- Thus, for each account/statement, BD_CST will debit or credit the related settle account for that SettleAccountName coming from the BOCashSettlement trade.



1.4 BO Position Update

1.4.1 Overview

BO Cash position is monitored through the BO Position:

- The Theoretical position type represents the Funded or Projected Cash Position;
- The Actual position type represents the Realized or Reconciled Cash Position;
- The Not Settled position type shows the difference between the two above-mentioned cash positions.

Theoretical Position takes FORECAST transfers into account until these transfers are excluded from that position and substituted by CASH POOLING deals. FORECAST transfer types never impact the Actual BO Position.

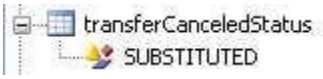
The general logic to substitute FORECAST with CASHPOOLING and update the BO Position accordingly is the following:

- As soon as a MT940 is processed for a specific Agent (Sender), Processing Org (Receiver) and Account (Tag25), the system cleans the related Theoretical Position for that Statement Date (Tag62F).
- This means that the system moves the FORECAST transfers having a Settle Date <= Statement Date for that specific Agent/Account into a status which is excluded from the Theoretical (and, of course, Actual) Position.
- This is achieved applying a specific “ForecastAction’ on the Trade and related Transfer (eg. SUBSTITUTE) set on the INC_CASH_STATEMENT Scheduled Task.

  The FORECAST type Simple Transfer trades and related transfers are thus moved to a status (SUBSTITUTED) excluded from the Inventory Position.

1.4.2 Configuration

Go to **Configuration > System > Domain Values**, select the domain transferCanceledStatus and add the transfer status to be excluded from the Inventory Position.



Then select the domain ignoreTradeStatus and add the trade status to be ignored, as shown below:

Then make sure to add the SUBSTITUTE Action on both the trade (for Simple Transfer with Flow Type FORECAST) and transfer workflows to allow the system performing this action on both underlying objects.

1.5 Scheduled Task MESSAGE_MATCHING

The import of the MT940 files is done through the MESSAGE_MATCHING scheduled task.

This scheduled task must be linked (using the Next Id) to the INC_CASH_STATEMENT scheduled task that implements the consistency checks and MT940 split process.

Thus, the MT940 integration process is a two-step process:

- We first save the complete incoming statements MT940 (using the MESSAGE_MATCHING);
- We then check the consistency and completeness of these statements, split the MT94O into sub-statements by entry, create the cash pooling deals and exclude forecasts from the cash position (using the INC_CASH_STATEMENT).

1.5.1 Scheduled Task Attributes

The integration of incoming MT940 is performed by the scheduled task MESSAGE_MATCHING.

The scheduled task criteria are summarized below.

Attributes

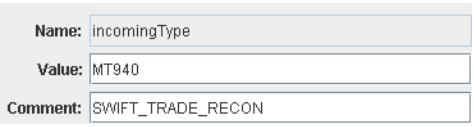
Attribute	Value
Swift Message Delimiter	
Swift File	20090325_BARC_EUR_0052.txt
InputDir	c:\temp
File Rename	False
Gateway	
ExternalMessageType	

- Swift Message Delimiter: Enter the delimiter between messages in the text file as specified in the environment property CALYPSO_SWIFT_LINE_SEPARATOR (should be blank if the property is not used)
- Swift File: Specify the Swift File Name (eg.20090325_BARC_0052.txt) to integrate one specific file. To integrate **all** incoming MT940, the user has to set the value ***.txt** on that field. The system will then try to save as 'bo_messages' all .txt files that are located under the InputDir
- InputDir: Specify the Swift File Path and Directory where MT940.txt files are saved
- File Rename: Select true/false. When set to true add a timestamp to the Swift File once processed

Gateway/ExternalMessageType: If the default implementation for processing MT940 is not satisfying, the CashStatementHandlerFactory tries to retrieve a Handler respecting the order Gateway – ExternalMessageType as specified in these attributes.

The MESSAGE_MATCHING Scheduled Task does not perform any validation; it only saves the incoming MT940 as a Calypso BO_MESSAGE type "SWIFT_TRADE_RECON". If not already populated, this message type can be added through **Configuration > System > Domain Values** – messageType.

One can rename that BO_MESSAGE Type as 'MT940' using the domain name incomingType and specifying as value MT940 and as comment SWIFT_TRADE_RECON as shown below:



The incoming SWIFT_TRADE_RECON message types saved by the MESSAGE_MATCHING Scheduled Task are marked with a specific message attribute "CashStatementProcess" = Statement.

This statement is a simple copy of each complete MT940 file text.

1.5.2 Message workflow SWIFT_TRADE_RECON

To be able to save the incoming MT940, one must define a Message Workflow for the message type SWIFT_TRADE_RECON.

If necessary, Calypso provides a workflow file that can be uploaded to create another default SWIFT_TRADE_RECON message workflow (actions, rules and static data filters listed below are created automatically).

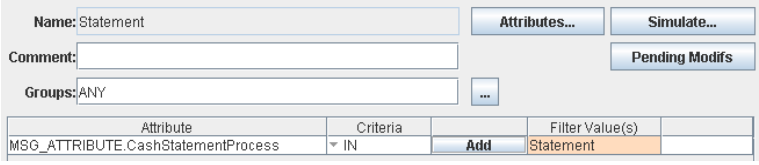
Please note that you need to add in an STP message workflow transition the HandleCashPooling rule to generate automatically the cash pooling deals.

SWIFT_TRADE_RECON Message Workflow

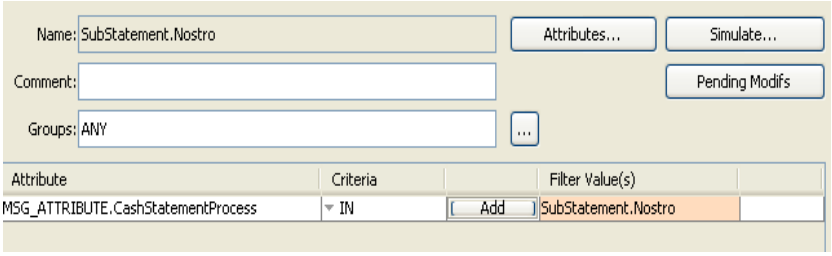
Id	Orig Status	Action	Resulting Status	Different User	Use STP	Log	Subtype	Rules	Processing Org	Kick Off/ Cut Off	Create Task	Filter
3237	IMPORTED	CANCEL	CANCELED	false	false	false	SWIFT_TRADE_RECON		ALL	false	false	
2817	NONE	NEW	PENDING	false	false	false	SWIFT_TRADE_RECON		ALL	false	true	Statement
12102	NONE	NEW	UNMATCHED	false	false	false	SWIFT_TRADE_RECON		ALL	false	true	SubStatement.Nostro
2818	NONE	NEW	UNPROCESSED	false	false	false	SWIFT_TRADE_RECON		ALL	false	true	SubStatement<>Nostro
3229	NOT_PROCESSED	CANCEL	CANCELED	false	false	false	SWIFT_TRADE_RECON		ALL	false	false	Statement
2822	NOT_PROCESSED	REPROCESS	PENDING	false	false	false	SWIFT_TRADE_RECON		ALL	false	true	Statement
3206	PENDING	AUTHORIZE	IMPORTED	false	false	false	SWIFT_TRADE_RECON		ALL	false	true	Statement
2820	PENDING	BLOCK	NOT_PROCESSED	false	false	false	SWIFT_TRADE_RECON		ALL	false	true	Statement
2821	PENDING	CANCEL	CANCELED	false	false	false	SWIFT_TRADE_RECON		ALL	false	false	Statement
2823	PROCESSED	CANCEL	CANCELED	false	false	false	SWIFT_TRADE_RECON		ALL	false	false	
13301	PROCESSED	REPROCESS	PROCESSEDAGAIN	false	false	false	SWIFT_TRADE_RECON	HandleCashPooling	ALL	false	true	
12107	PROCESSED	UNMATCH	UNMATCHED	false	false	false	SWIFT_TRADE_RECON		ALL	false	true	
12104	UNMATCHED	AUTOMATCH	PROCESSED	false	false	false	SWIFT_TRADE_RECON		ALL	false	true	
12105	UNMATCHED	CANCEL	CANCELED	false	false	false	SWIFT_TRADE_RECON		ALL	false	false	
12106	UNMATCHED	MANUALMATCH	PROCESSED	false	false	false	SWIFT_TRADE_RECON		ALL	false	true	
12108	UNPROCESSED	CANCEL	CANCELED	false	false	false	SWIFT_TRADE_RECON		ALL	false	false	
2828	UNPROCESSED	PROCESS	PROCESSED	false	true	false	SWIFT_TRADE_RECON	HandleCashPooling	ALL	false	true	SubStatement<>CashSweeping
12103	UNPROCESSED	PROCESS	UNMATCHED	false	true	false	SWIFT_TRADE_RECON	HandleCashPooling	ALL	false	true	SubStatement.CashSweeping

Using the specific Static Data Filters defined below:

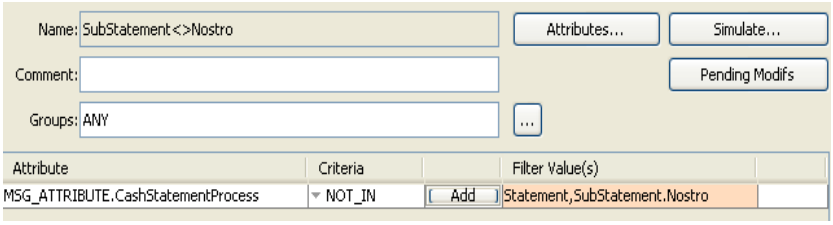
- Statement



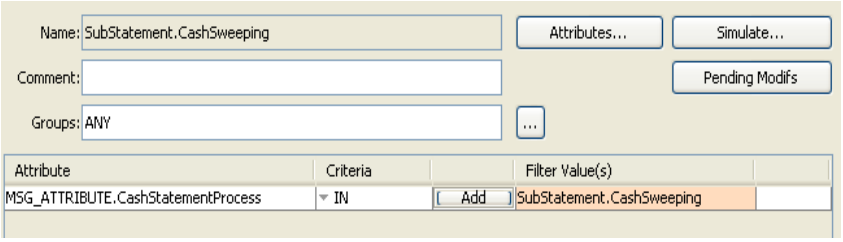
- Substatement.Nostro



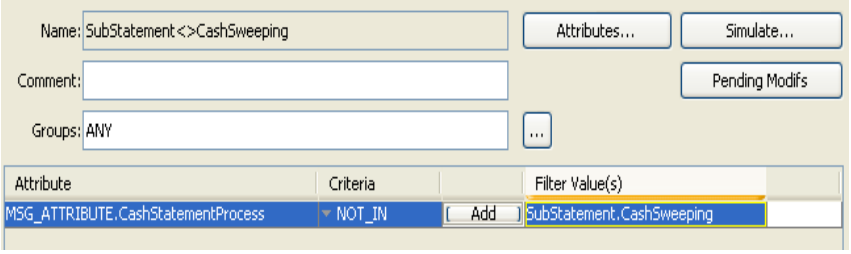
- SubStatement<>Nostro



- SubStatement.CashSweeping



- SubStatement <> CashSweeping



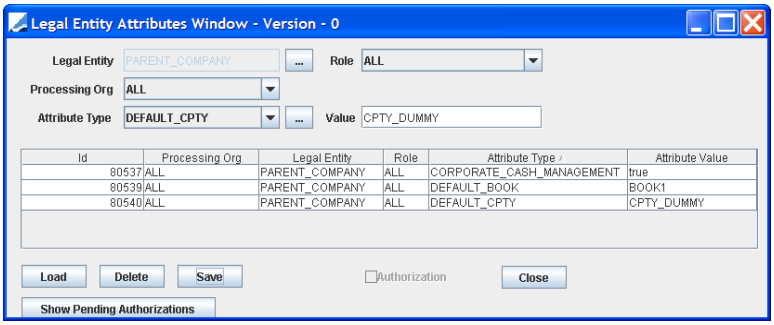
1.6 Scheduled Task INC_CASH_STATEMENT

Once the Scheduled Task MESSAGE_MATCHING is reported as successful, the system must run the INC_CASH_STATEMENT scheduled task to process the incoming MT940. This processing is only done for Processing Organizations marked with a specific attribute CORPORATE_CASH_MANAGEMENT = true.

1.6.1 Processing Organization Legal Entity Attributes

Define the following LE Attributes for your processing organizations:

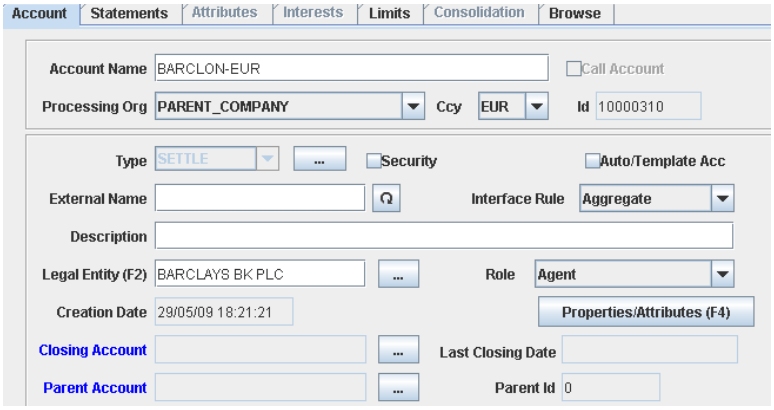
- CORPORATE_CASH_MANAGEMENT = true
- DEFAULT_BOOK = “YourDefaultBook” for that processing organization
- DEFAULT_COUNTERPARTY = “YourDefaultCounterParty” for that processing organization

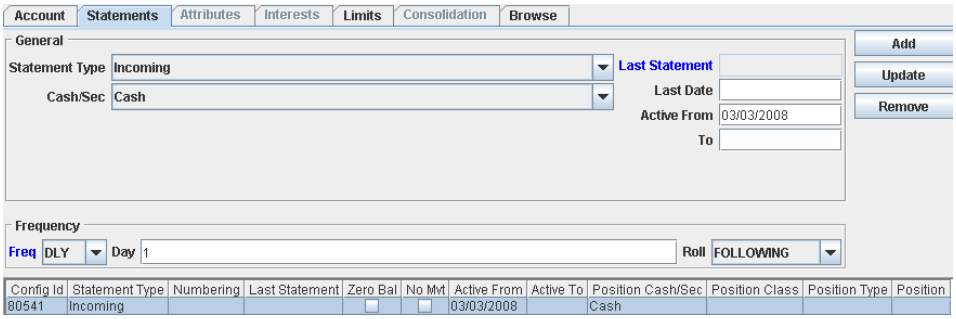


The INC_CASH_STATEMENT process is only done for Processing Organizations marked with the attribute CORPORATE_CASH_MANAGEMENT = true.

Once the Scheduled Task MESSAGE_MATCHING is reported as successful, the system runs the ‘linked’ INC_CASH_STATEMENT scheduled task to process the incoming MT940.

To be able to save and split a MT940, one must attach an Incoming Statement Configuration to the SETTLE Account(s) for which the processing organization is expecting a MT940, as shown below for a EUR Bank Account at Barclays London:





The Date Rule attached to the Incoming Statement Configuration in the Frequency panel is used to determine **when** we expect an Incoming MT940 for this specific account:

- When missing, the scheduled task reports an exception;
- When a statement is received for an unexpected date (not part of the date rule), the scheduled task reports an exception.

1.6.2 Scheduled Task Attributes

The INC_CASH_STATEMENT is a two-step process:

Step 1: Process Incoming Cash Statements:

- Load all "statement" SWIFT_TRADE_RECON messages (initially saved by the MESSAGE_MATCHING) applying the "Statement statuses restriction" attribute for the Scheduled Task **Valuation Date** (= Tag62F statement date);
- Group the "Statement" SWIFT_TRADE_RECON messages by GL (SETTLE) Account;
- The correspondence between the MT940 Tag 25 for a specific Processing Organization (Receiver BIC) + Agent (Sender BIC) and the Calypso GL Account is done using the GL Account Attribute "**XferAgentAccount**" where we store the value of Tag 25;
- Check the Group consistency (Sequence Numbering/Intermediary Balances/Statement Opening Balance = Previous Statement Closing Balance, etc.);
- Create Account Statement summarizing the group information;
- Create Sub-Statement Messages (with specific types) and Cash Pooling Trades;
- Create BOCashSettlement Trade for each valid statement;
- Apply the "Valid/Invalid Statements Action" on Account Statement summarizing the group information:
  - SWIFT_TRADE_RECON Message Workflow Valid Action is applied on complete and consistent grouped statements
  - SWIFT_TRADE_RECON Message Workflow Invalid Action is applied on incomplete and/or inconsistent grouped statements

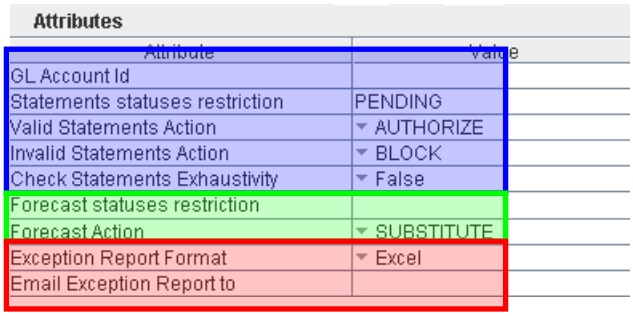
Step 2: Exclude FORECAST Transfer Types from the Cash Position for the Processing Organization Bank Accounts successfully processed in Step 1:

- Load all simple transfer trades with transfer type = FORECAST and related transfers type FORECAST having a Settle Date <= Scheduled Task Valuation Date for the GL Accounts in the group applying the "Forecast statuses restriction" attribute;

- Apply the “Forecast Action” set as attribute on the selected Simple Transfer trades type FORECAST and related transfers;
- The “Forecast Action” is an action that must be added to your trade and transfer workflows (for instance, VERIFIED – SUBSTITUTE – SUBSTITUTED);
- This resulting status (SUBSTITUTED in our example) must be set in the transferCanceledStatus and ignoreTradeStatus to make the system treat them as a CANCEL and exclude transfers with that specific status from the BO Position. This also allows building specific trade and transfer reports including or excluding these forecasts.

These ‘actions’ are driven by the attributes set on the INC_CASH_STATEMENT Scheduled Task.

INC_CASH_STATEMENT Attributes



- GL Account Id: Specify the Processing Organization SETTLE account for which you want to run the process. If left blank, the process is run for all the processing organization SETTLE accounts initially saved by the MESSAGE_MATCHING scheduled task.
- Statement statuses restriction: Specify the “SWIFT_TRADE_RECON” message status to take into account for the processing (in the above example, only SWIFT_TRADE_RECON statements in PENDING status are processed by that scheduled task) for the Scheduled Task Valuation Date (= Tag62F statement date);
- Valid Statements Action: Specify the SWIFT_TRADE_RECON workflow action to apply on complete and consistent grouped statements. In the above example, the system applies the AUTHORIZE action on complete and consistent incoming MT940.
- Invalid Statements Action: Specify the SWIFT_TRADE_RECON workflow action to apply on uncomplete and inconsistent grouped statements. In the above example, the system applies the BLOCK action on uncomplete and/or inconsistent incoming MT940.
- Check Statements Exhaustivity: true/false. When set to true, the system reports ‘missing statements’ as exceptions in the exception file (eg. a statement is expected for a specific bank account according to the incoming statement configuration date rule but the related MT940 is not received). When set to false, the system only reports ‘exceptions’ for received statements, without reporting missing statements (type missing page, wrong opening balance, etc.).
- Forecast statuses restriction: Specify the trade and transfer status to take into account to load the FORECAST trades/transfers. The system will load all Simple Transfer trades with type = FORECAST and related transfers type FORECAST which have a Settle Date <= Scheduled Task Valuation Date for that/these status(es) (eg. no restriction in the above example).
- Forecast Action: Specify the trade and transfer workflow action that the system must apply on the selected FORECAST trades and transfer (eg. SUBSTITUTE trade and transfer action will be applied on FORECAST

transfers and Simple Transfer Trades with Transfer Type = FORECAST in the above example). The "Forecast Action" is a trade and transfer action that must be added to your trade and transfer workflows (eg. VERIFIED – SUBSTITUTE – SUBSTITUTED for this type of trader/transfer using SD Filters or Trade and Xfer workflow types). Then, one must add the resulting status in **Configuration > System > Domain Values** – transferCanceledStatus and ignoreTradeStatus to make the system treat that status as a CANCEL and, as such, exclude transfers with that specific status from the BO Position:

- Exception Report Format: The Scheduled Task generates an Exception report which allows having a summary of all the issues encountered during the processing. Define the format of that exception file in the "Exception Report Format" attribute. Available format include: excel, html, csv, pdf.
- Email Exception Report to: This exception file can be sent by email. Specify the email address or alias using the "Email Exception Report to" attribute.

Consistency prerequisite checks performed by the INC_CASH_STATEMENT

The consistency checks performed by the INC_CASH_STATEMENT scheduled task are the following:

- Check if all start-balances match the previous end-balances;
- Check that all intermediate balances for an account are matching (Tag 60M/62M for a n-page statement);
- Check that opening + movements = closing balance;
- Check statement completeness and order (number of pages and existence of Tag 62F);
- Check if all accounts exist in MT940 and vice-versa;
- Check if date of end balance is not in the future;
- Check if there is only one end-balance (and so MT940) per day.

1.6.3 Automatic Cash Pooling Trades

The INC_CASH_STATEMENT scheduled task creates automatically cash pooling deals. These cash pooling deals are of different categories:

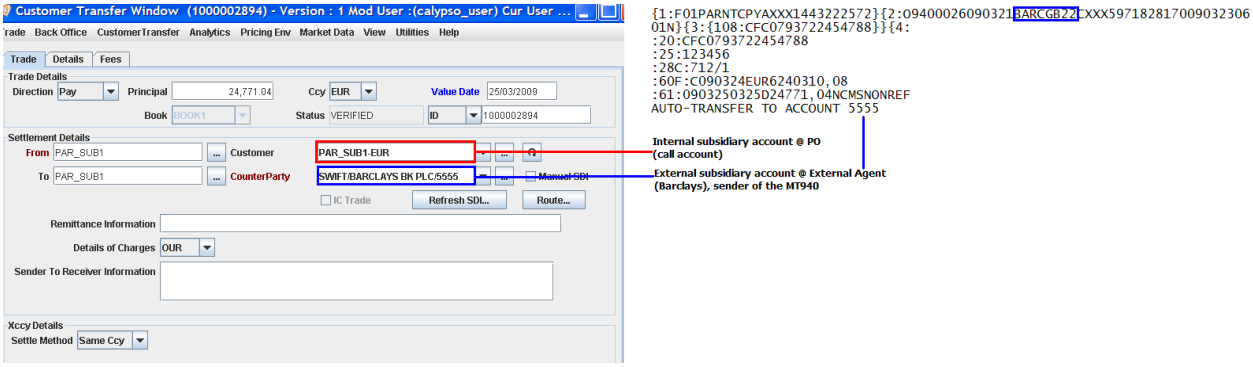
- Customer Transfer Trades
- Transfer Agent and Simple Transfer for cross-PO sweeping

Customer Transfer Trade

From the incoming MT940,

- When the processing organization bank account is a standard or guarantee fees account
- When the system can find a mapping rule (in the table Account Mapping) with Account Identifier confirmed in Tag 61 and/or 86 of the statement = CounterParty SDI A/C field or, if such an SDI does not exist or is not preferred, if the system can find a generic SDI for the statement agent/sender and with beneficiary = PO with Role = CounterParty.

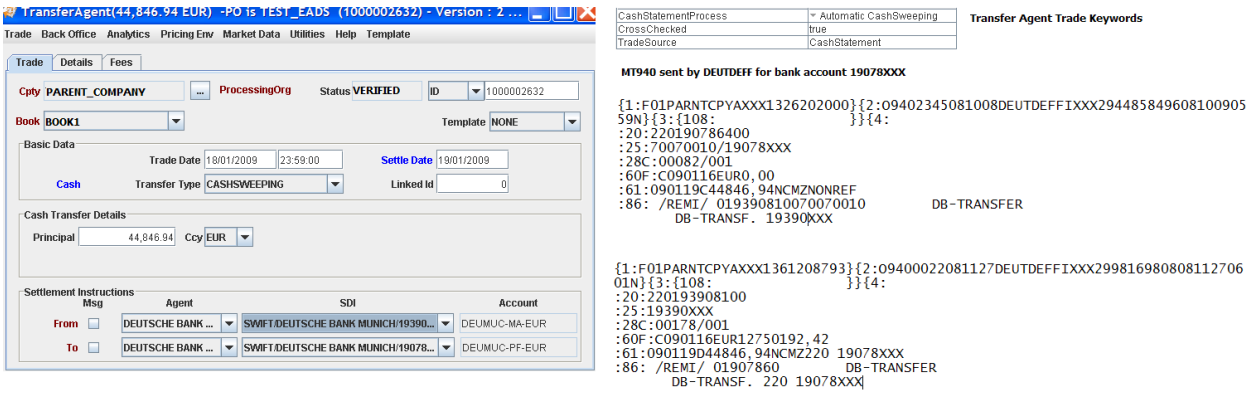
Then, provided the related subsidiary has a call account SDI defined with Agent = processing organization (= MT940 Receiver), the system automatically creates a cash pooling deal on the product customer transfer, as shown below:



From the incoming MT940, when the processing organization bank account is a payment factory account, if the system finds an Account Mapping Rule but cannot map the account identifier confirmed in Tag 61 and/or 86 of the statement with the CounterParty SDI A/C field, then the system retrieves a generic SDI defined with Role CounterParty for the LE = Master Account Processing Organization (from the Account Mapping Table) for that account currency and, provided the subsidiary has a call account SDI defined with Agent = processing organization (= MT940 Receiver), the system automatically creates a cash pooling deal on the product customer transfer between the Subsidiary and the P.O. as CounterParty.

Transfer Agent Trade

From the incoming MT940, when the system can find an Account Mapping Rule between two processing organization bank accounts and when there is no existing 'matching' transfer, the system automatically creates the cash sweeping deal using the product transfer agent, as shown below:



A cross-checked keyword is set to true only when both statements confirming both legs (pay/receive) are received and mapped with the trade related transfers.

When the mapping exists for accounts belonging to two different processing organizations, the system creates automatically two Simple Transfers.

Detailed mapping algorithm is provided in Appendix.

FX & MM Consolidation Deals

Once cash pooling deals have been created by the INC_CASH_STATEMENT scheduled task, the system must run the Scheduled Task called ACCOUNT_CONSOLIDATION.

This scheduled task creates FX Spot and Cash MM deals to zero balance some subsidiary call accounts.

The computation of the FX Spot and/or Cash deals depends on the subsidiary call account balance **after integration and processing of MT940**, the cash pooling deals generated from the statement impacting the subsidiary call account.

The ACCOUNT_CONSOLIDATION scheduled task must be linked (NextId) to the INC_CASH_STATEMENT scheduled task.

Once the INC_CASH_STATEMENT is reported as “successful”, the system automatically runs the ACCOUNT_CONSOLIDATION to create the FX and Cash deals zero-balancing some subsidiary call accounts.

The deal creation is done **in sequence**. Thus, the system first creates the FX Spot deals for subsidiary call accounts that are not allowed to carry an FX position. Then the system simulates the impact of these FX Spot related transfers on the subsidiary call account balance for the currency of consolidation of that account. Based on that simulation, the scheduled task then creates the Cash deals to move this subsidiary call account balance to zero.

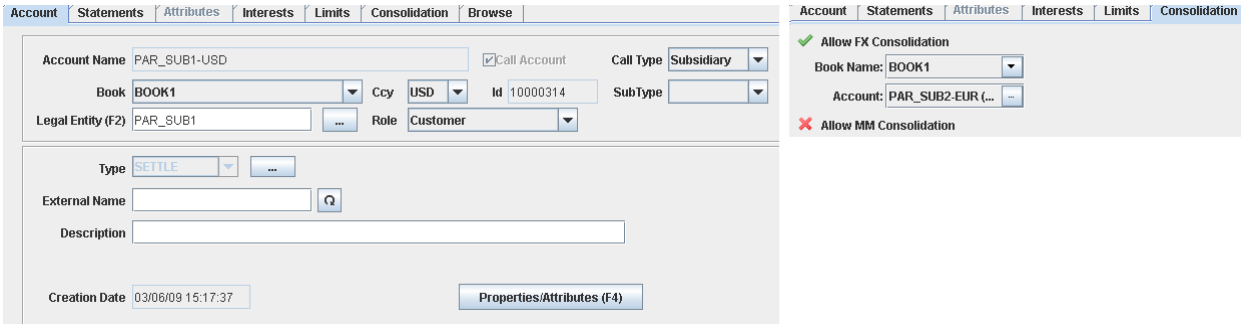
2.1 Account Configuration

2.1.1 FX Consolidation

To flag a call account as eligible to the FX consolidation process, go to the **Consolidation** panel of the call account, activate the ‘Allow FX Consolidation’ flag and specify the book and the FX consolidation call account. The FX consolidation call account must have a different currency than the subsidiary call account. The consolidation currency = FX consolidation call account currency.

In the example below, the internal subsidiary call account PAR_SUB1-USD is consolidated in EUR using the PAR_SUB2-EUR call account.

This means that PAR_SUB1-USD cannot carry a USD position. Any USD position on that call account will be automatically converted in EUR using the PAR_SUB2-EUR call account.

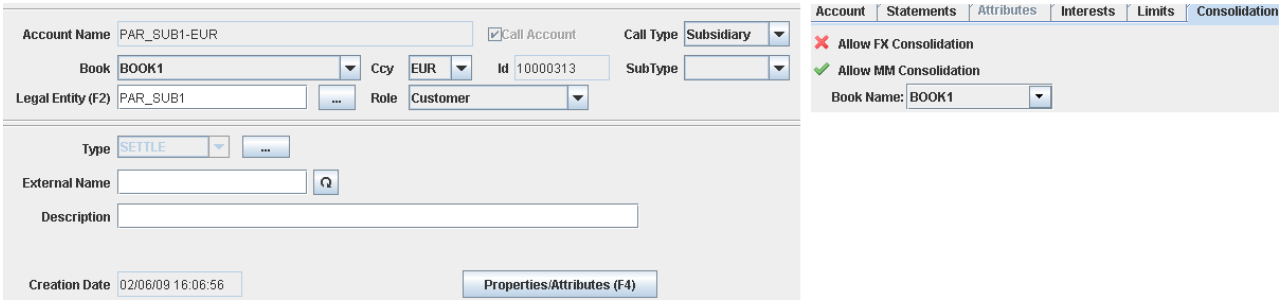


2.1.2 MM Consolidation

To flag a call account as eligible to the MM consolidation process, go to the **Consolidation** panel of the call account, activate the 'Allow MM Consolidation' flag and specify the consolidation book that will be used for the Cash trade creation.

In the example below, the internal subsidiary call account PAR_SUB1-EUR is consolidated in EUR using BOOK1.

This means that PAR_SUB1-EUR cannot carry a negative balance. Any negative balance on that call account will be automatically counterbalanced by a Cash trade in Calypso.

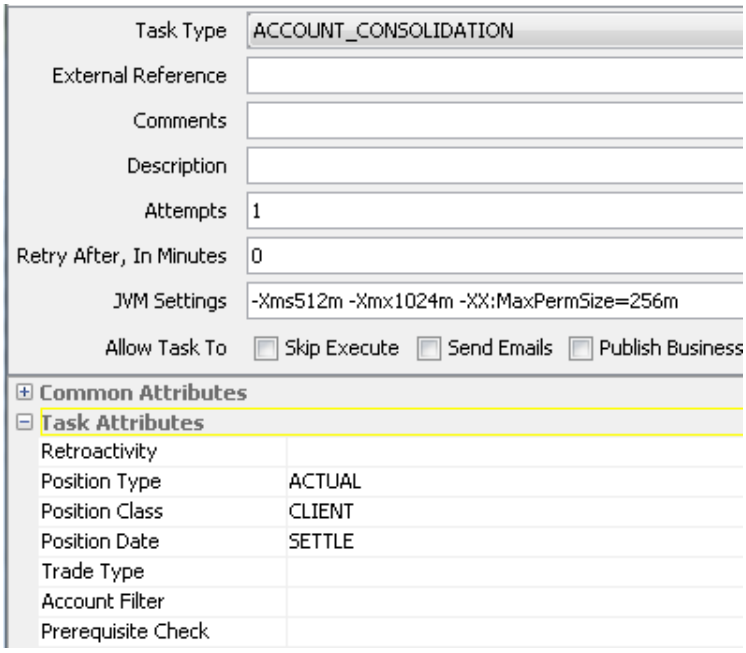


2.2 Scheduled Task ACCOUNT_CONSOLIDATION

Internal consolidation FX and MM deals are generated by the ACCOUNT_CONSOLIDATION scheduled task.

The ACCOUNT_CONSOLIDATION scheduled task must be linked (NextId) to the INC_CASH_STATEMENT scheduled task.

Once the INC_CASH_STATEMENT is reported as "successful", the system automatically runs that process to create the FX and Cash deals zero-balancing the "eligible" subsidiary call accounts.



Attributes

- Position Type: Specify the type of the inventory position to retrieve for eligible call accounts. The process must run on the ACTUAL position type.
- Position Class: Specify the class of the inventory position to retrieve for eligible call accounts. The process must run on the CLIENT position class.
- Position Date: Specify the date type of the inventory position to retrieve for eligible call accounts. The process must run on the SETTLE position date.
- Trade Type: Specify if the process must generate FX only, Cash only or both. This can be used when the user must rerun only part of the process on a limited scope of call accounts.
- Account Filter: Select the Static Data Filter among the list of available filters. This can be used, for instance, when the user wants to run or rerun the process on a limit scope of call accounts or for specific banks, etc.
- Prerequisite Check: Blank or SubsidiaryAccount. When SubsidiaryAccount value is set, the system checks if all the bank account statements for the call accounts concerned by that process have been received and processed: if yes the system creates the FX deals then the MM; else the system only creates the FX (and MM) deals for the call accounts for which we have received all the MT940. The check is performed based on the rules set in the Account Mapping table.

2.2.1 FX Consolidation deal generation process

When running the ACCOUNT_CONSOLIDATION scheduled task, the system first retrieves ALL the (subsidiary) call accounts that have been defined with an FX Consolidation Account.

For these call accounts, the system retrieves the CLIENT-ACTUAL-SETTLE inventory position as specified in the scheduled task attributes for the process Valuation Date (T-1).

For these call accounts, the system also retrieves the Account Mapping Rules to check if all statements have been received and processed for all processing organization accounts (one call account is linked to 1 or n P.O. bank account) associated to that call account in the Mapping table.

When all statements impacting a call account have been received and processed and when the CLIENT-ACTUAL-SETTLE inventory position for that call account is not null on that date, the system creates FX Spot deals according to the following rules:

- if that balance > 0, the system create two spot trades as follows:
  - FX Spot Deal # 1: Buy subsidiary call account ccy/Sell related FX consolidation account ccy
  - FX Spot Deal # 2: Sell subsidiary call account ccy/Buy related FX consolidaton account ccy
- If that balance < 0, the system will create two spot trades as follows:
  - FX Spot Deal #1 Sell subsidiary call account ccy/Buy related FX consolidation account ccy
  - FX Spot Deal #2 Buy subsidiary call account ccy/Sell related FX consolidation account ccy

For these FX deals:

- Trade and Settle Date = scheduled task Valuation Date
- Trade Book = Book set on the consolidation panel
- Trade role = Account Holder (subsidiary 1 & 2)
- FX rate is taken from the quotes for the scheduled task Valuation Date. The system uses the Quote Set and Pricing Params (QuoteUsage) associated with the pricing environment set on the scheduled task

Please note that **when the Subsidiary is also defined a as Processing Organization**, the system creates **four FX transactions** (instead of 2): 2 FX Spot (above mentionned) between the Parent Company as a Processing Organization and the Subsidiary as an AccountHolder plus 2 FX Spot between the Subsidiary as a Processing Organization and the Parent Company as a CounterParty. This may be needed for subsidiary reporting purposes.

If a Statement is missing or has not been processed and/or if an FX rate is missing, the system raises an exception in a dedicated report named “Process Status Report”.

2.2.2 MM Consolidation deal generation process

After generation of the FX consolidation deals, the ACCOUNT_CONSOLIDATION scheduled task creates the Cash deals for subsidiary call accounts that are not allowed to carry a negative balance in the consolidation currency.

When running the scheduled task ACC_CONSOLIDATION, after creation of FX consolidation deals, the system retrieves ALL the subsidiary call accounts that have been marked with ‘Allow MM Consolidation’.

For these call accounts, the system also retrieves the **Account Mapping Rules** to check if all statements have been received and processed for all processing organization accounts (one call account is linked to 1 or n P.O. bank account) associated to that call account in the Mapping table.

For these call accounts, the system simulates the CLIENT-ACTUAL-SETTLE inventory position (as specified in the scheduled task attributes) on the scheduled task Valuation Date, including transfers associated to the FX consolidation deals.

When this CLIENT-ACTUAL-SETTLE inventory position is negative on that date, the system creates Cash deals according to the following rules:

- Loan (this is displayed from the Processing Organization perspective => A Loan/PO is a Deposit ‘Client’)
- Trade role = Account Holder
- Trade book = Book set on the consolidation panel for MM
- Ccy = subsidiary call account ccy
- Nominal = (simulated) call account balance
- Trade Date = Start Date = Scheduled Task Valuation Date
- End Date = Start Date + 1 Business Day, considering the currency holidays
- Frequency = ZC
- Date Roll = MOD_FOLLOW
- Period = ADJUSTED
- No Amortization / No Stub / No Cmp
- Rate conditions (including Day Count) are taken from the call account interest configuration for the **Call Account Balance**, especially when tiered matrix applies. If floating rate applies, rate value is taken from the Quotes for the scheduled task Valuation Date. The system uses the Quote Set and Pricing Params (QuoteUsage) associated with the pricing environment set on the scheduled task

Please note that **when the Subsidiary is also defined a as Processing Organization**, the system creates **two MM transactions** (instead of 1): 1 Cash MM (above mentionned) between the Parent Company as a Processing

Organization and the Subsidiary as an AccountHolder plus 1 Cash MM between the Subsidiary as a Processing Organization and the Parent Company as a CounterParty. This may be needed for subsidiary reporting purposes.

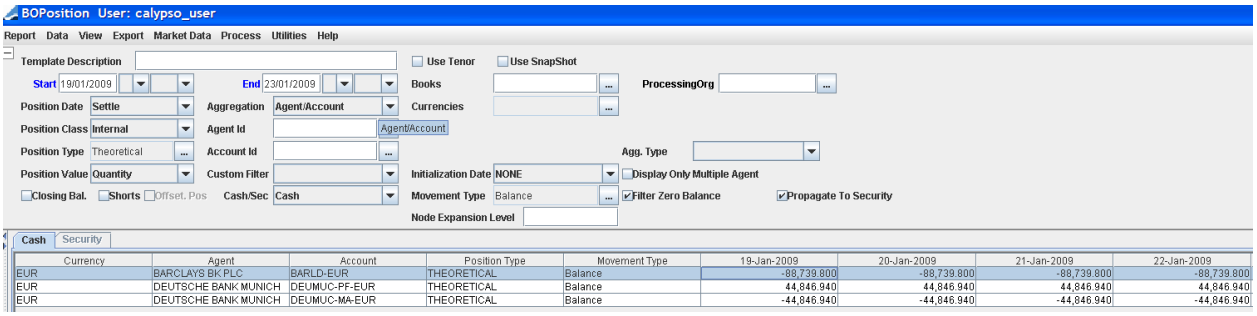
If a Statement is missing or has not been processed and/or if a MM rate is missing, the system raises an exception in a dedicated report named “Process Status Report”.

BO Positions and Process Status Reports

3.1 BO Position

The Calypso reporting to check the cash balance by bank/account/ccy, account/ccy or ccy is the Back-Office Position Report.

This reporting is available under **Report > Nostro/Custodian Position > Back Office Position**.



This report displays the global position based on the Aggregation level defined by the user (Agent/Account in the above example).

Double-click on a balance cell automatically displays the details of flows for that day.

Using the function Show - Default configuration available by a right click on a cell, the user can fix a predefined Transfer Report template that will always pop-up when double-clicking on a cell. This allows providing the adequate information (Flow type Forecast/CashPooling, etc.) that may not be available in the default transfer details report.

Load is based on the criteria set by the user. The BO Position uses the Calypso Reporting Framework and as such allows selecting, sorting and renaming the selected columns.

A detailed description of the report capabilities is available on-line using the Help menu.

3.2 Account Activity Report

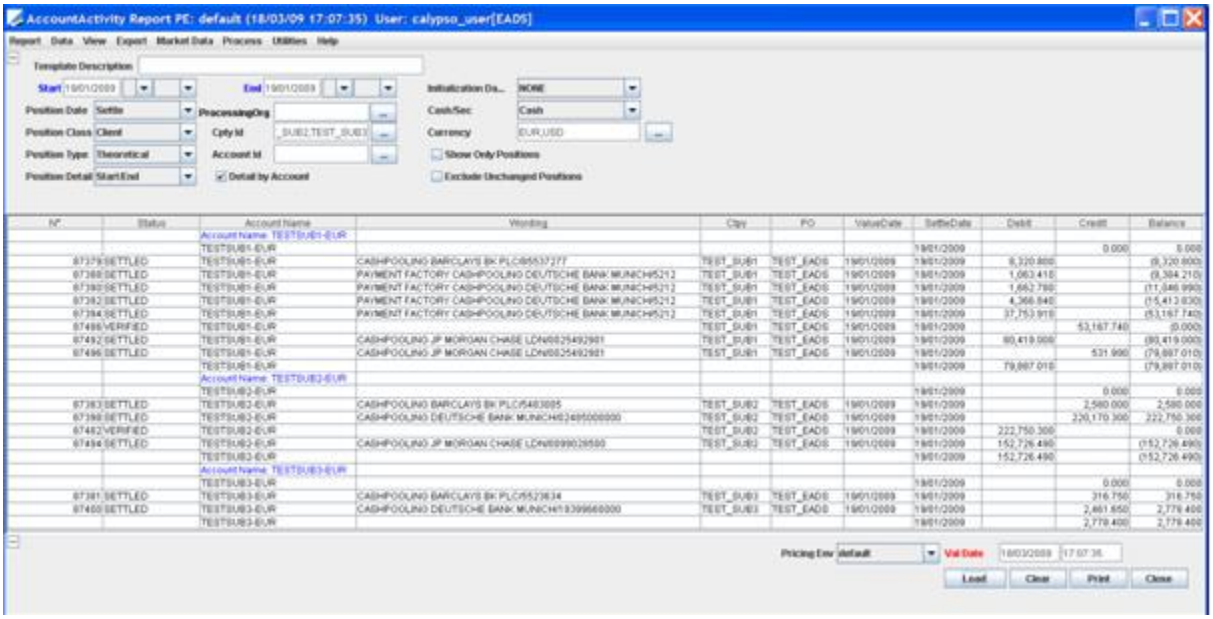
The Calypso reporting to check the opening/closing balance and movements is the Account Activity Report.

This reporting is available in **Report > Nostro/Custodian Position > Account Activity**.

We have added the possibility to display several accounts at the same time using the Detail by Account flag. This flag must be checked to display n subsidiary call accounts in the same report.

The Account Activity uses the Calypso Reporting Framework and as such allows selecting, sorting and renaming the selected columns.

The user must choose the Account Name as a column/sort and subheading criteria to display a correct view when Detail by Account is ticked, as shown below:



The Xfer Description is used to populate the type of movement (column Wording in the above example).

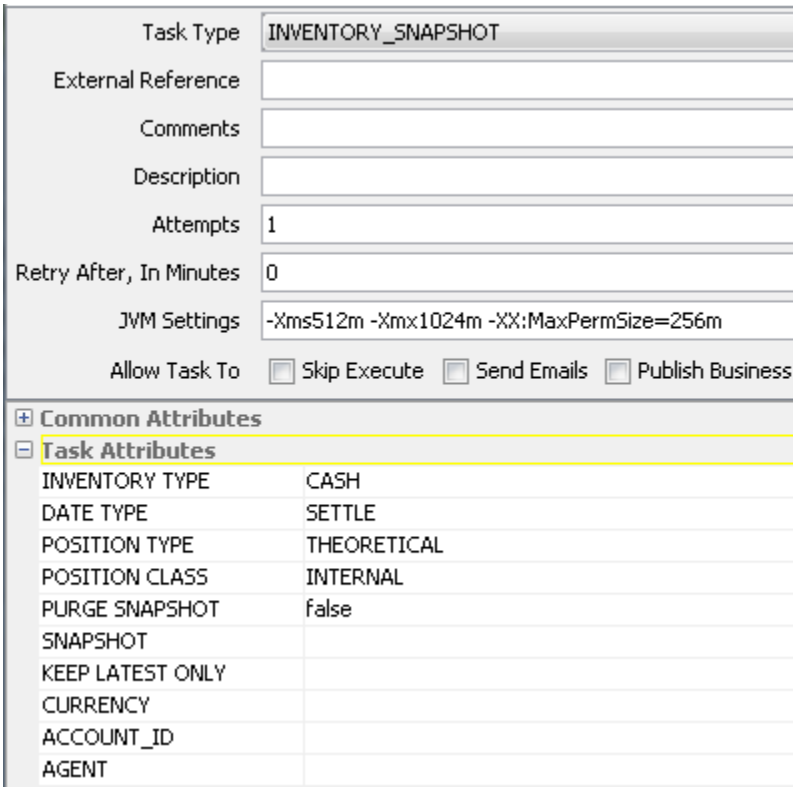
The report is based on a **Position Class** that can be set to **Client** (to display one or n subsidiary internal/call account) or **Internal** (to display one or n processing organization bank accounts) to display the opening, closing balances and movements. The Position Date allows displaying the account b alance and movements by Booking Date, Value Date or Settle Date.

3.3 BO Position Reconciliation Report

3.3.1 Scheduled Task INVENTORY_SNAPSHOT

Using the Inventory Snapshot Scheduled task, we provide the ability for the treasurer to freeze the position at any given time and identify the difference (transfers) in comparison with any previous snapshot or the current inventory position.

To access that scheduled task, go to **Configuration > Scheduled Tasks** and select the INVENTORY_SNAPSHOT. This Scheduled Task will load inventory position between the specified “From” and “To” days and will store it in the Inventory Tables. The snapshot date will be stored in the domain value Bo_position_snapshot with the types of position and the dates in the comment field.



This scheduled task loads inventory positions between the “From” and “To” days and stores them in the inventory tables.

The snapshot date is stored in the domain value Bo_position_snapshot with the types of position and the dates in the comment field.

Attributes

- INVENTORY TYPE: Allow the user to choose between ALL, CASH or SECURITY. Depending on the type of inventory selected you can store/freeze cash and security (ALL), only cash (CASH) or only security (SECURITY) inventory positions as “Snapshot Positions”
- DATE TYPE: Allow the user to freeze TRADE, SETTLE, AVAILABLE or ALL inventory positions date types
- POSITION TYPE: Allow the user to freeze the THEORETICAL, ACTUAL, FAILED or ALL inventory position types
- POSITION CLASS: Allow the user to freeze the INTERNAL, EXTERNAL, CLIENT or MARGINCALL inventory position classes
- PURGE SNAPSHOT: Set to false by default. If set to true, the scheduled task will delete all (if unspecified in the field SNAPSHOT) or only the specified “snapshot inventory positions” mentioned in the attribute SNAPSHOT
- SNAPSHOT: Only active when PURGE_SNAPSHOT is set to true. This allows the user to specify which snapshot inventory position he wants to purge/delete from the inventory
- KEEP LATEST ONLY: Active in conjunction with PURGE_SNAPSHOT set to true. The default value is false. When set to true, when performing the PURGE_SNAPSHOT the system will delete all snapshot inventory positions except the latest one

- CURRENCY: Default is blank (= ALL). Pick a Currency if you want to limit the snapshot process for the positions specified in that currency
- ACCOUNT_ID: Default is blank (= ALL). Enter an Account Id to limit the snapshot process to that specific account
- AGENT: Default is blank (= ALL). Enter the Agent Short Name to limit the snapshot process to accounts hold at that Agent only

From Days/To

The INVENTORY_SNAPSHOT scheduled task will store inventory positions as snapshots depending on the number of days specified in From and To field.

For instance, if you set From = -5 and To = 5, when running the scheduled task the system will store inventory positions from Valuation Date – 5 days till Valuation Date + 5days

In order to display the frozen position(s) in the BO Position, we have added a Snapshot flag on that position screen.

When activated, the system asks the user to choose which frozen position he wants to display. Once selected, the user has to hit the Load button to see that position. Double-clicking on a cell allows drilling-down to the snapshot movements for that position.

3.3.2 BO Position Reconciliation Report

To access the BO Position Reconciliation Report, go to **Processing > Reconciliation > BO Position Reconciliation**.

In the BO Position Reconciliation screen, the user has the ability to select a snapshot position class (in both the Base and/or Target) by flagging the “Use Snapshot” box available in these panels.

This allows the user to make reconciliations between any type of inventory position and a “snapshot” inventory position.

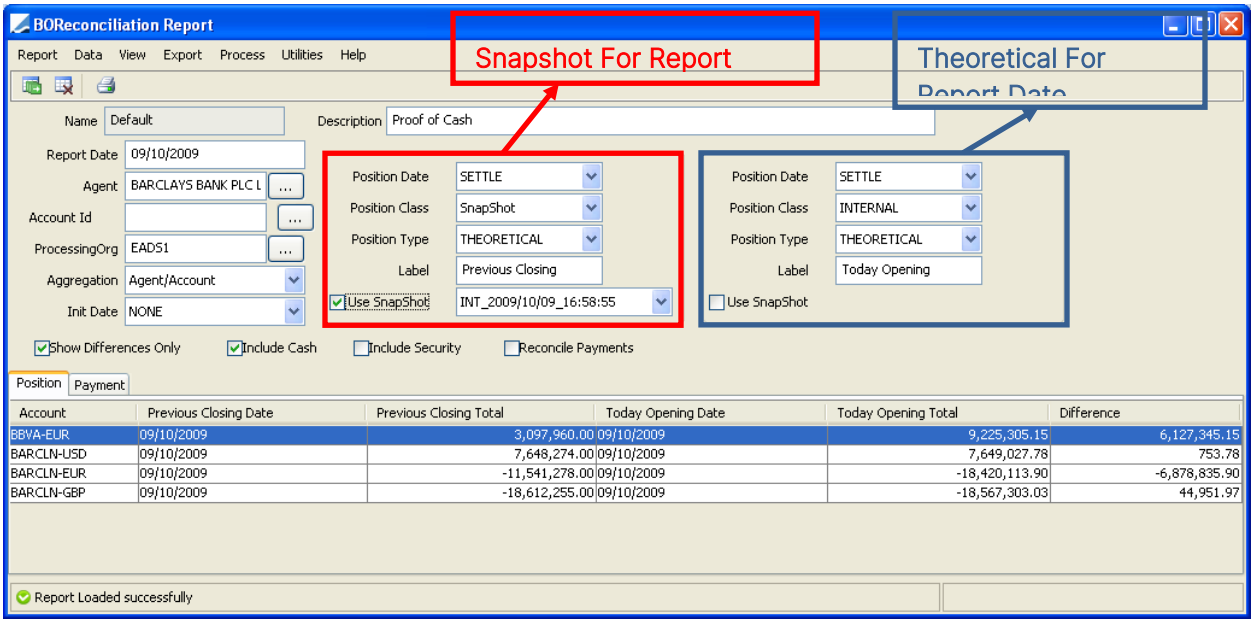
The user can limit the reconciliation process for:

- 1 or n Agent(s)/Bank(s)
- 1 or n Account(s)
- 1 or n Processing Organization(s)

The BO Position Reconciliation screen gives a summary view by agent/account of the reconciliation results between two position types. To do so, you must set a BO Position Reconciliation Template with the following criteria:

- Agent = ALL (blank)
- Account Id = ALL (blank)
- Processing Org. = Your P.O.
- Aggregation = Agent/Account
- Previous Closing = Snaphot Position of report date (Position Date = SETTLE, Position Class = Snapshot, Position Type) THEORETICAL and snapshot = latest snapshot position saved by the Scheduled Task)

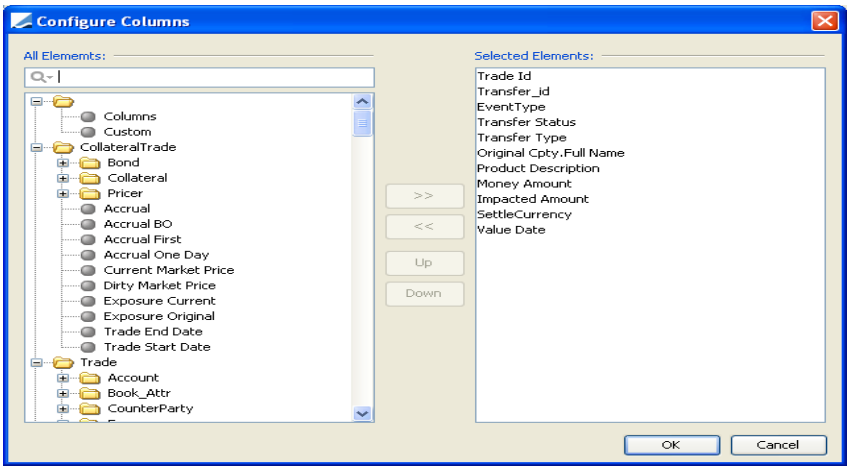
- Today Opening = Same day standard position (Position Date = SETTLE, Position Class = INTERNAL, Position Type = THEORETICAL)
- "Show Differences Only" must be picked
- "Include Cash" must be picked



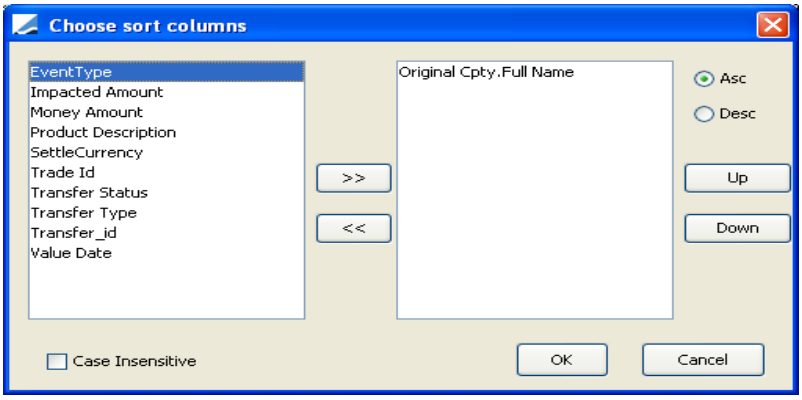
Then, to explain the **opening balance of the day** and display the **movements** explaining the difference between the previous day closing balance (snapshot of the THEORETICAL/SETTLE/INTERNAL) and the current day opening balance (standard THEORETICAL/SETTLE/INTERNAL position), the user must select one line/account, right-click and choose the function "Explain Difference" available in the Process menu of that report.

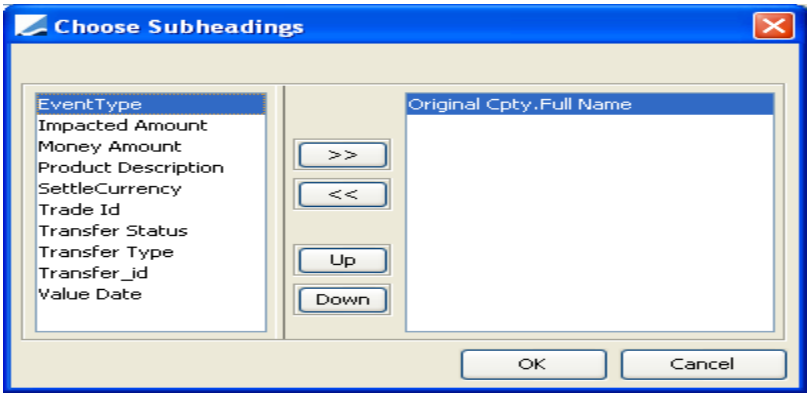
Before doing so, the user must define the transfer report template that will be called by this Explain Difference function.

To do so, go to the Transfer Report menu. Select the following columns in the Configure Columns menu (example):

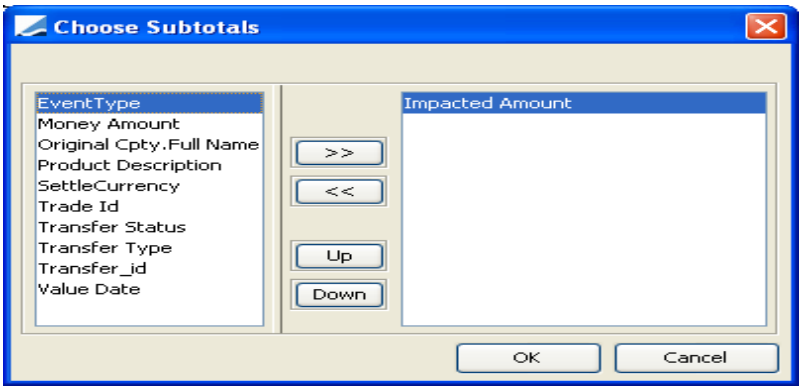


Select the Original Ctpy.Full Name as a Sort Criteria and SubHeadings criteria, as stated below:

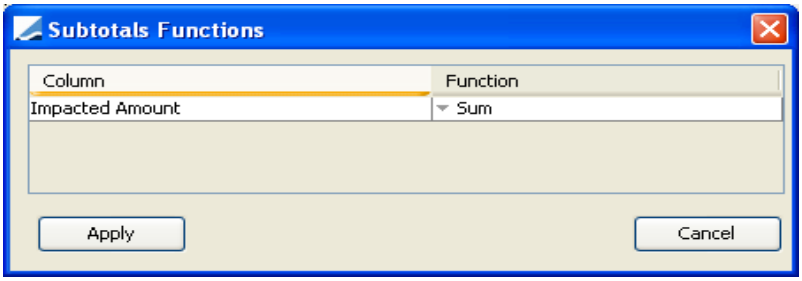




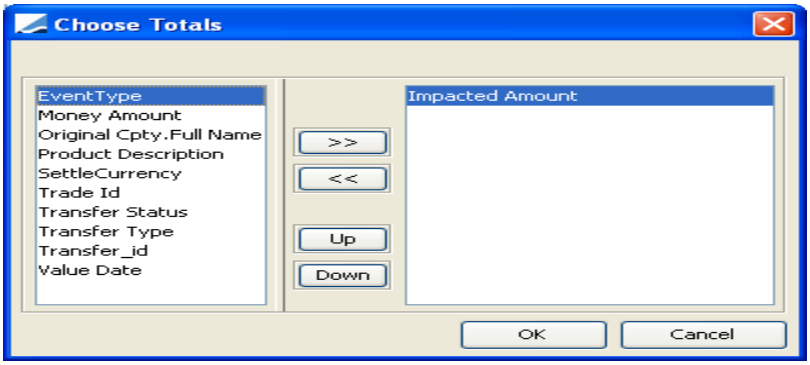
Select the Impacted Amount as Subtotals criteria:

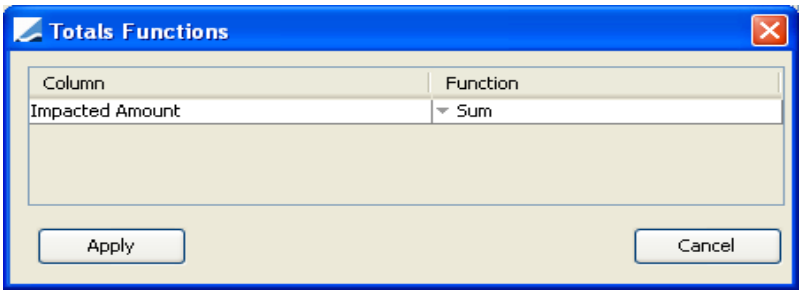


Select the Sum Function for that criteria as SubTotals Function, as stated below:



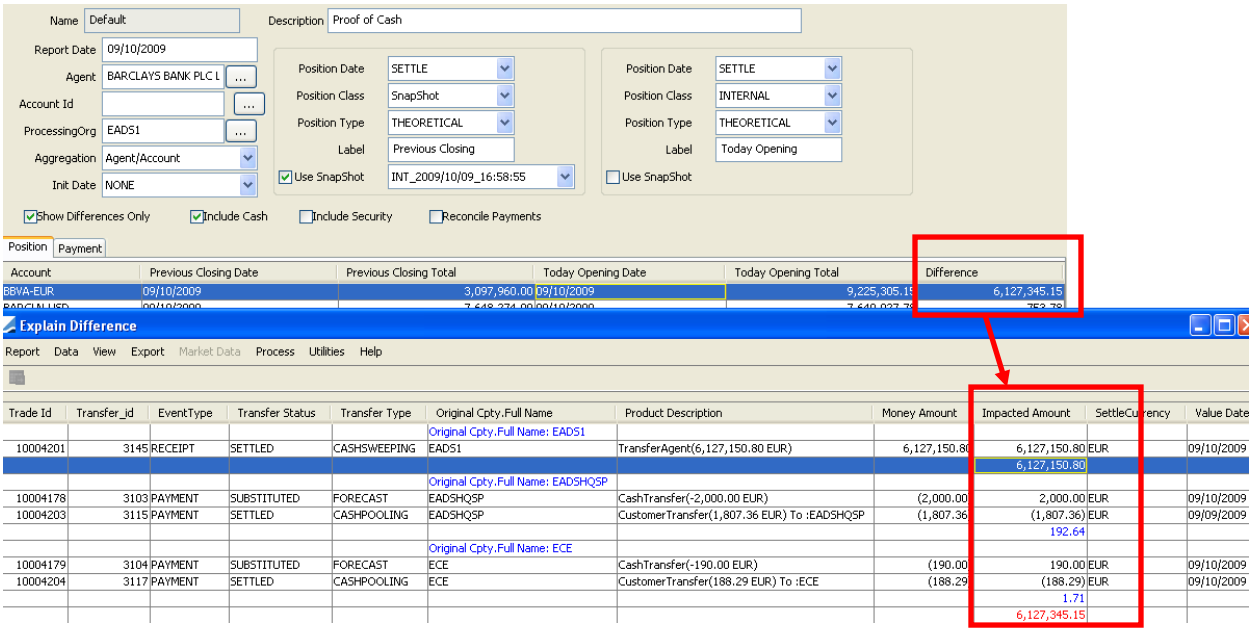
Do the same setup for the Totals, as stated below:





Then save that template with the **mandatory Template Name: Reconciliation** (to call it by default when choosing the BO Position Reconciliation – Explain Difference menu).

Once done, when the user chooses the "Explain Difference" function, the system pops-up that transfer template, as stated below:



The user can double-click on each line of that report to drill-down to the movement details. Additional information/columns can be displayed in that report. The only mandatory columns is the impacted amount and the subtotal/total using that specific column.

3.4 Process Status Report

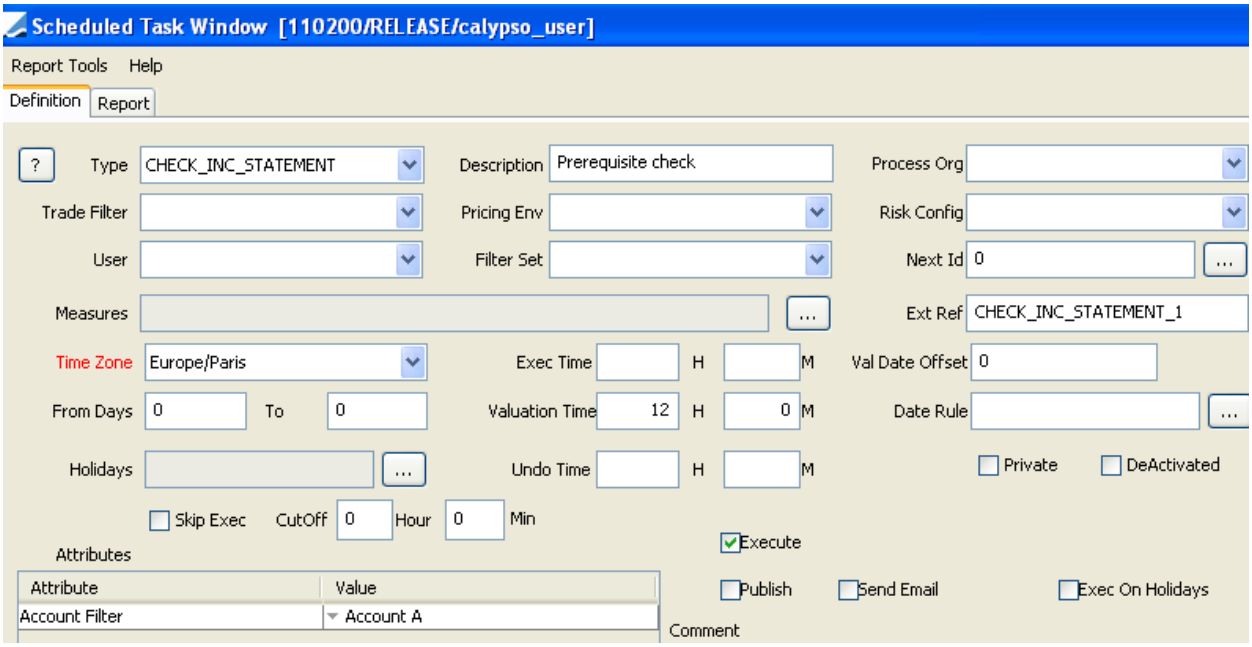
A new report named Process Status report (reporting.ReportWindow$ProcessStatus) has been added to report exceptions/errors blocking all or part of the chained process for all or a subset of subsidiaries accounts.

3.4.1 Scheduled Task CHECK_INC_STATEMENT

A new scheduled task CHECK_INC_STATEMENT has been created to check the **completeness** of the incoming statements (MT940) before running the ACCOUNT_CONSOLIDATION and ACCOUNT_STATEMENT scheduled tasks.

The scheduled tasks sequence should be the following:

- MESSAGE_MATCHING to integrate the incoming statements
- INC_CASH_STATEMENT to split and type the statement into sub-statements; for some types, the system automatically creates some deals based on the mapping information (and related static data) set in the Account Mapping table
- CHECK_INC_STATEMENT to send the 'status' information into the Process Status Report. An Account Filter (SD Filter) is available as an attribute to limit the process to a list of call accounts.



- ACCOUNT_CONSOLIDATION to generate the FX and MM deals to move the balance of specific internal call accounts to zero
- ACCOUNT_INTEREST to compute the daily interest entries for internal call accounts
- ACCOUNT_STATEMENT to generate the internal statements sent to subsidiairies (either MT940 or HTML statements)

The CHECK_INC_STATEMENT Scheduled Tasks creates the exception/errors reported in the Process Status Report.

3.4.2 Report

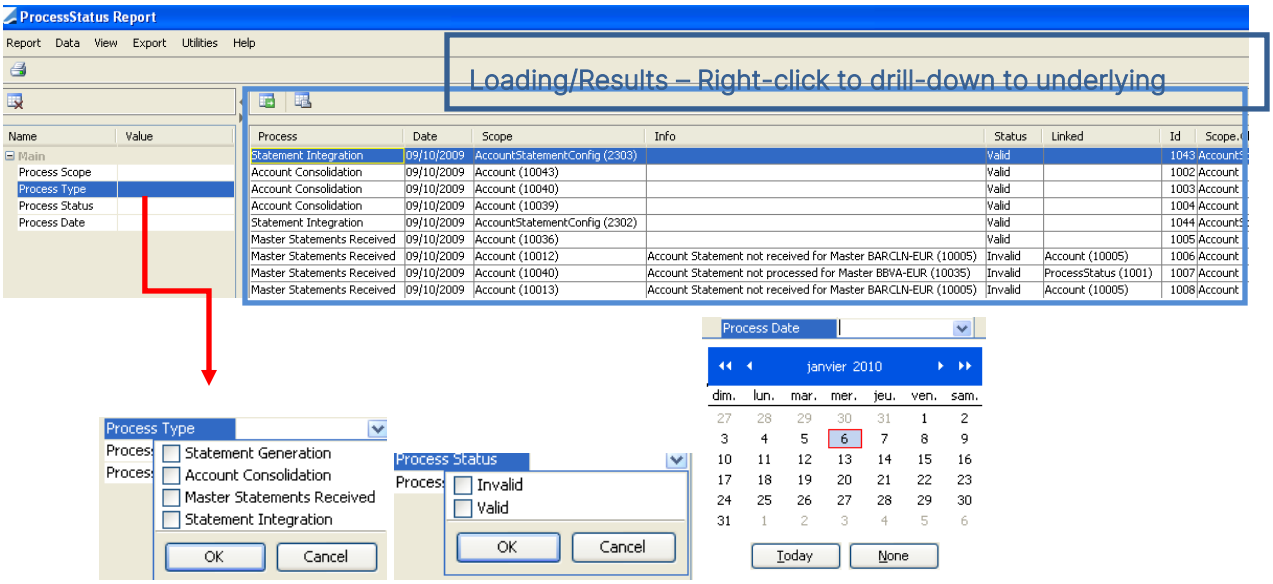
The report Process Status allows checking the status of automatic chained processes from the integration of incoming statement MT940 to the generation of internal subsidiary call account statement generation. Only incorrect or incomplete internal accounts are blocked. Others are fully processed.

This report lists the following processing scopes with their status (valid/invalid):

- Statement Integration: the report produces a line per P.O. Bank Account having an Incoming Statement Configuration. If the statement is expected and received (based on the incoming statement date rule) the status is valid, else the status is invalid and a comment (Info column) is provided to give the reason of failure. From that 'status' line the user can reach all the information regarding the account, statement config. etc. and take further action.
- Master Statement Received: Based on the Account Mapping table we store the relationships between the P.O. Bank Accounts (for which we receive MT940) and the Subsidiary Call Accounts impacted by these P.O. Accounts / Statements. Thus, for each call account, the Process Status reports a line type 'Master Statement Received' by Statement (Nostro) impacting the internal subsidiary account. If the Master Statement impacting that subsidiary internal account has been received, status is valid, else it is invalid.
- Account Consolidation: when all Master Statements impacting a call account for which we have to automatically create FX and MM deals have been received and processed, the Process Status reports an

Account Consolidation line with a valid status. When one or n Master Statements have not been received or processed, blocking the consolidation process, the Process Status reports an Account Consolidation error with an Invalid Status and a comment (Info column) to give the reason of failure. From that 'status' line the user can reach all the information regarding the account, etc. and take further action. The 'invalid' call accounts are blocked and FX/MM deals are not generated automatically for these internal accounts.

- **Statement Generation**: when all Master Statements impacting a call account have been received and processed and when the Account Consolidation has successfully run, the internal subsidiary call account statements can be generated. In such a case, the Process Status reports a Statement Generation line with a valid status. Else, it reports a Statement Generation line for that(these) call account(s) with an invalid status. The 'invalid' call accounts are blocked and the statements are not generated automatically for these internal accounts.
- **Layout Example**: the Process Status Report uses the calypso report framework with possibility to configure and rename columns, save templates and right-click to drill-down to the related object and investigate. An example is presented below:



Process	Date	Scope	Info	Status	Linked	Id
Statement Integration	09/10/2009	AccountStatementConfig (2303)		Valid		1043
Account Consolidation	09/10/2009	Account (10043)		Valid		1002
Account Consolidation	09/10/2009	Account (10040)		Valid		1003
Account Consolidation	09/10/2009	Account (10039)		Valid		1004
Statement Integration	09/10/2009	AccountStatementConfig (2302)		Valid		1044
Master Statements Received	09/10/2009	Account (10036)		Valid		1005
Master Statements Received	09/10/2009	Account (10012)	Account Statement not received for Master BARCLN-EUR (10005)	Invalid	Account (10005)	1006
Master Statements Received	09/10/2009	Account (10040)	Account Statement not processed for Master BBVA-EUR (10035)	Invalid	ProcessStatus (1001)	1007
Master Statements Received	09/10/2009	Account (10013)	Account Statement not received for Master BARCLN-EUR (10005)	Invalid	Account (10005)	1008

The Process Status allows blocking the Account Consolidation and Internal Subsidiary Call Account Statement Generation for incomplete Call Accounts. Thus, when one or n Master Statement impacting a subsidiary internal call account is/are missing or incorrect, the system blocks that internal account. An exception (Invalid Status) is reported in the Process Status and the process will have to be rerun manually for that internal account once the new statements are received.

Other internal subsidiary accounts that are complete are processed. Only incorrect or incomplete internal accounts are blocked.

Account Sweeping

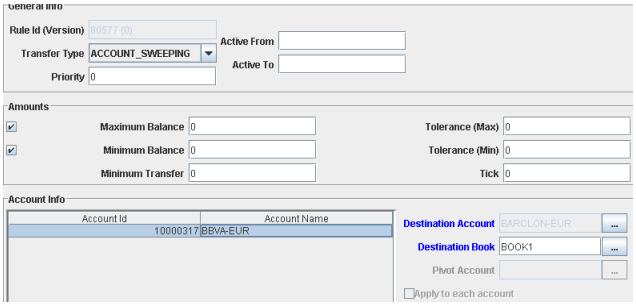
4.1 Active Account Sweeping Process from BO Position

We have added the possibility to trigger the account sweeping directly from the BO Position.

To use that feature, first go to **Configuration > Account > Account Sweeping** and define the account sweeping rules for accounts that are **actively** swept.

An example is presented below for BBVA-EUR account swept within BARCLON-EUR account.

Please refer to the general Cash Management User Guide for more information regarding account sweeping configuration.

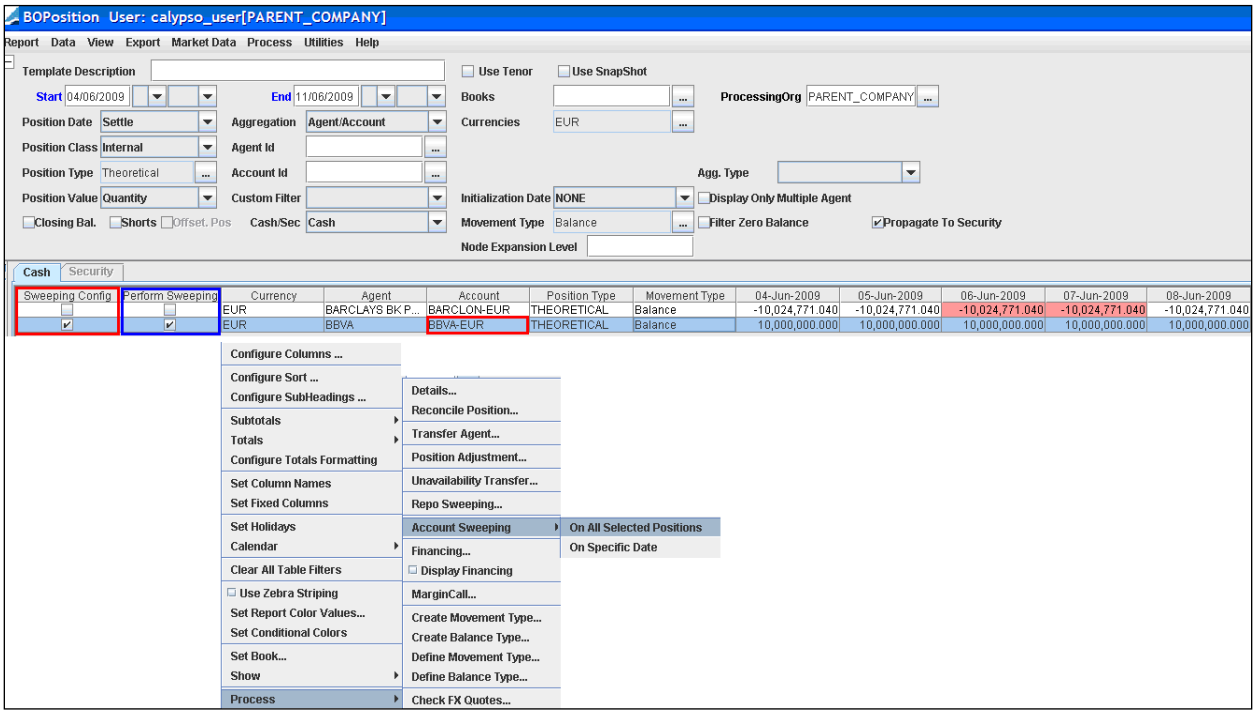


Then, go to the BO Position Report.

Two new columns have been added in that report:

- **Sweeping Config**: to identify accounts for which a sweeping rule is set in the account sweeping configuration. When ticked, it means the account can be swept based on the account sweeping rule criteria previously defined for that account;
- **Perform Sweeping**: to select an account to sweep. The user can select all accounts that have sweeping rules or only a subset of these accounts. Thus, when the sweeping is triggered from the BO Position, the user keeps control of the sweeping process and may decide to sweep only some accounts for which sweeping rules exist but not all.

An example is shown below:



To actively sweep an account from the BO Position, the user must right-click, select the Process function (or go to the menu Process) / **Account Sweeping** and choose:

- Account Sweeping / **On All Selected Positions**: this will create Transfer Agent deals for all accounts ticked in the 'Perform Sweeping' column. Transfer Agents deal will move the balance for the selected accounts based on the account sweeping rules set in the account sweeping configuration. When this option is chosen, the sweeping (Transfer Agent deals) is done on BO Position **Start Date;**
- Account Sweeping / **On Specific Date**: the system also allows performing account sweeping for a **Specific Date**, different from the BO Position Start Date. To use that function, the user must first select the accounts to sweep by checking the 'Perform Sweeping' flag for these accounts, select the a cell corresponding to the sweeping date, right-click to select the Process function / **Account Sweeping** and choose 'On Specific Date' option. The system will perform the sweeping (Transfer Agent deal generation) for all the selected balances on that date.

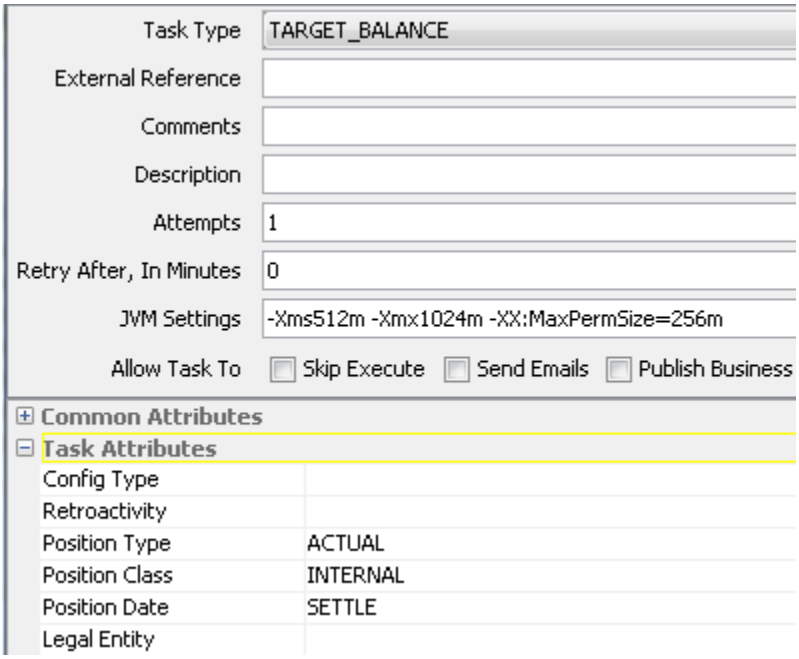
The BO Position Account Sweeping function must be used when the user wants to **keep control** of the sweeping process and determine manually (based on account characteristics/sweeping rules) which accounts, among the list of sweeping accounts, must finally be swept.

When no prerequisite control/order is required, the scheduled task **TARGET_BALANCE** must be used. It will always perform the sweeping for all accounts based on their sweeping rules.

4.2 TARGET_BALANCE Active Account Sweeping

When no prerequisite control/order is required, the scheduled task **TARGET_BALANCE** must be used. It will always perform the sweeping for all accounts based on their sweeping rules.

The Scheduled Tasks window is accessed from **Configuration > Scheduled Tasks**.



Attributes

- Retroactivity: Enter true or false. Setting this field to true means that in case of back valued transfers, the sweeping process will generate Trade transfers related to the back valued amounts and dates, instead of the process date.
- Position Type: Specify the type of the inventory position to retrieve for 'nostro' bank accounts. The process must run on the ACTUAL position type.
- Position Class: Specify the class of the inventory position to retrieve for 'nostro' bank accounts. The process must run on the position class INTERNAL.
- Position Date: Specify the date type of the inventory position to retrieve. The process must run on the SETTLE position date.
- Legal Entity: Specify an Agent or leave it blank to select ALL Agents.

This process is based on the Inventory Position.

The result is the generation of Transfer Agent trades (for 'nostro' bank account sweep).

To view the Transfer Agent trades, choose **Processing - Accounting Operations - Transfer Agent – Trade** - Open.

4.3 Passive Account Sweeping Process

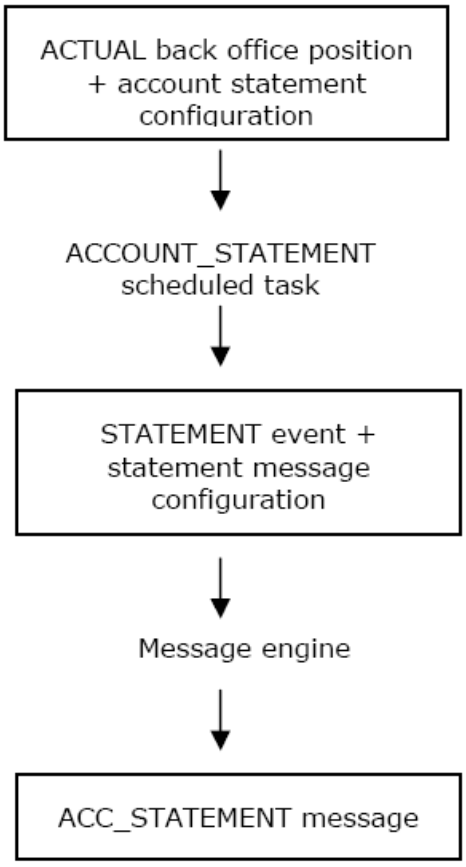
When account sweeping is passive (ie. generated by the bank on behalf of the processing organization), Calypso creates the Transfer Agent trade (or Simple Transfer trades when two processing organizations are involved) when processing the incoming MT940 in T+1. This is based on the Account Mapping Rules set in the Mapping table.

Please also refer to the Appendix for details in terms of mapping and creation rules.

Internal Messages

5.1 Scheduled Task ACCOUNT_STATEMENT

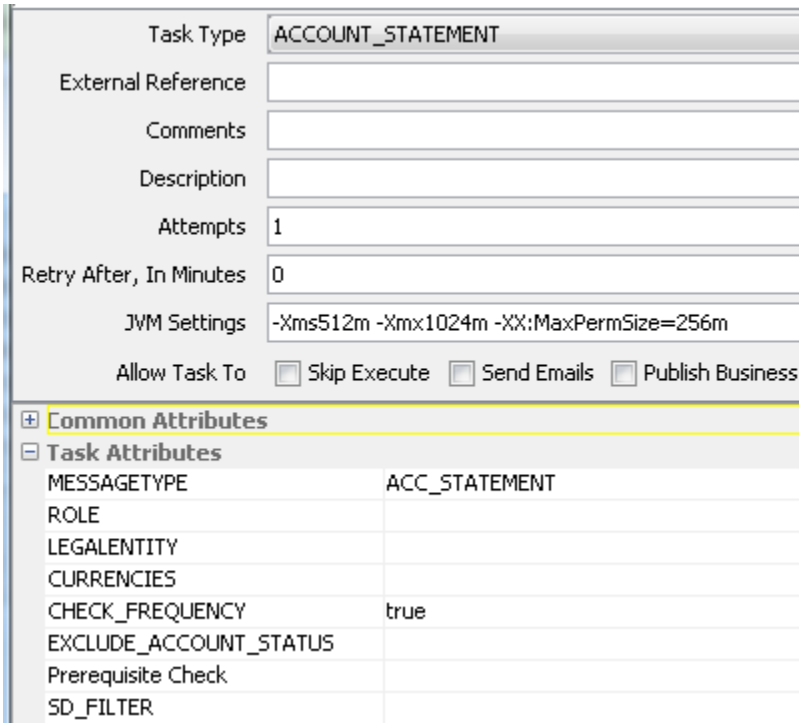
Account statement events are generated by the scheduled task ACCOUNT_STATEMENT based on **actual cash account positions**, and **account statement configurations**. The Message engine subscribes to account statements events and generates **account statement messages** based on **statement message configurations**



The scheduled task ACCOUNT_STATEMENT generates statement events based on account statement configurations and cash account positions computed by the Inventory engine. The statement events are then processed by the Message engine for the actual generation of the statement document.

Go to **Configuration > Scheduled Tasks** to open the Scheduled Tasks window, and select the ACCOUNT_STATEMENT scheduled task. Select a trade filter, a user, a pricing environment, and a processing organization.

In the scheduled task, you can specify as Attribute LEGALENTITY the AccountHolder (Customer) for which you want to generate a statement (to generate a statement only for that LE), and the type of message (it must be the same as the message configuration selected in the statement configuration = ACC_STATEMENT).



Attributes

- MESSAGETYPE – Select an account statement message type for this type of message to be generated (ACC_STATEMENT in our example to match the Advice Config)
- ROLE – Select the value set in the domain Name AccountHolderRole (Customer in our example) to be able to send account statements to customers
- LEGALENTITY – Enter the name of the L.E. or leave it blank for all Customers
- CHECK_FREQUENCY – Must be set to true to generate statements only for statement configuration date rules matching the Scheduled Task Valuation Date. If you want to use the Scheduled Task to generate a statement manually, this attribute must be set to false

5.2 Subsidiary Internal MT940

Each subsidiary call account must be defined with a statement configuration to produce daily internal Swift MT940.

These internal MT940 are sent by the processing organization to the subsidiary (Account Holder) and mirror what the bank account statement has confirmed for the related internal account.

A Date Rule, set on that configuration, determines the frequency of production of these internal MT940.

The Flag Zero Balance and No Movement leave the possibility to always send a MT940, including when there is no movement or when balance is zero.

The position to trigger the internal MT940 generation is the Client – Actual – **Booking Date** Position.

The Advice Configuration to use must refer to the Message Template MT940 (SWIFT). That Advice Configuration can be set for Product Type = N/A or Call Account.

An example of call account statement configuration is presented below.

Please note that you need to refer to an advice configuration previously saved in your message setup before configuring the Account Statement Configuration.

Generation of MT940 is triggered by the ACCOUNT_STATEMENT Scheduled Task based on the Statement Frequency.

We also allow a user to generate an ad-hoc MT940 or a duplicate of a previously generated MT940 using the Account Statement Configuration Report under **Reports > Accounting Reports > Account Statement Config**.

For more information regarding call account statements, please refer to the Client Custody Management User Guide.

5.3 Subsidiary Internal Bank Account Statement

Each subsidiary call account must have one (or n) dedicated statement configurations to produce HTML Bank Account Statements.

A Date Rule set on the Account Statement Configuration determines the frequency of production of these internal bank account statements.

The Flag Zero Balance and No Movement leave the possibility to always send a MT940, including when there is no movement or when balance is zero.

The position to trigger the internal MT940 generation is the Client – Actual – **Booking** or **Settle** Date Position.

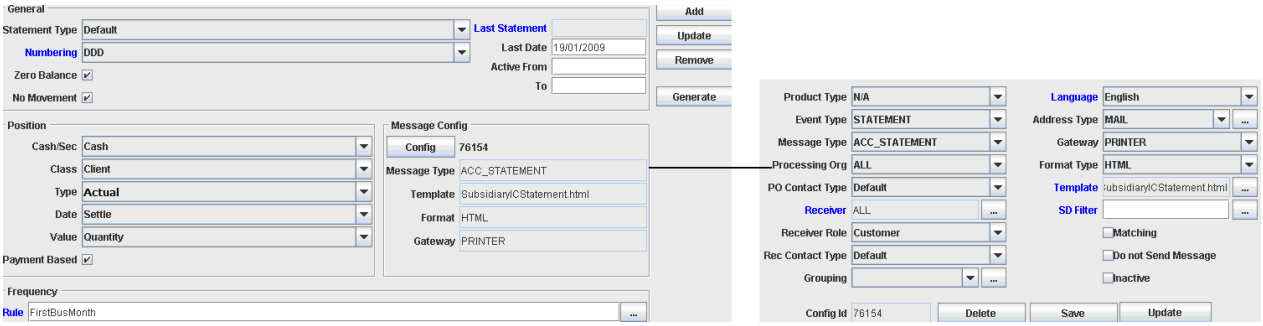
The Advice Configuration to use must refer to the HTML Templates:

- SubsidiaryICStatement.html for Value Date statements
- SubsidiaryICStatementBookingDate.html for Booking Date statements

These Advice Configurations can be set for Product Type = N/A or Call Account.

An example of call account statement configuration in **Value/Settle Date** is presented below.

Please note that you need to refer to an advice configuration already saved in your message setup before configuring the Account Statement Configuration:



Generation of internal call account statement is triggered by the ACCOUNT_STATEMENT Scheduled Task based on the Statement Frequency.

We also allow a user to generate an ad-hoc HTML Bank Account Statement or a duplicate of a previously generated Statement using the Account Statement Configuration Report under **Reports > Accounting Reports > Account Statement Config**.

All Statements are audited; we store for each one the Opening/Closing Balance and related movements for audit & investigation purposes.

An example of statement layout is presented below:

Calypso FINANCE

PARENT COMPANY
Bernard Dupont
Treasury
1 Straat
Ottobrun

GERMANY

Date: March 18, 2009
To: **PAR_SUB1**
Address: 106 Rue de la Paix
Frankfurt
GERMANY

This is Not a Formal Statement

Account Statement Number from 02-Jan-2009 to 19-Jan-2009

Account Number : **PAR_SUB1-EUR**

Account Owner : Alain Durand

Account Currency : EUR

Statement status : NEW

Original Statement Number :

MOVEMENTS DETAILS

VALUE DATE	SETTLE_DATE	TRAN REF	DETAILS	DEBIT	CREDIT	BALANCE
02/01/2009	02/01/2009		OPENING BALANCE			0.000
19/01/2009	19/01/2009	1000002533	CASHPOOLING BARCLAYS BK PLC/85537277	8,320.800		8,320.800DR
19/01/2009	19/01/2009	1000002538	PAYMENT FACTORY CASHPOOLING DEUTSCHE BANK MUNICH/5212	1,063.410		9,384.210DR
19/01/2009	19/01/2009	1000002539	PAYMENT FACTORY CASHPOOLING DEUTSCHE BANK MUNICH/5212	1,662.780		11,046.990DR
19/01/2009	19/01/2009	1000002540	PAYMENT FACTORY CASHPOOLING DEUTSCHE BANK MUNICH/5212	4,366.840		15,413.830DR
19/01/2009	19/01/2009	1000002541	PAYMENT FACTORY CASHPOOLING DEUTSCHE BANK MUNICH/5212	37,753.910		53,167.740DR
19/01/2009	19/01/2009		CLOSING BALANCE			53,167.740DR

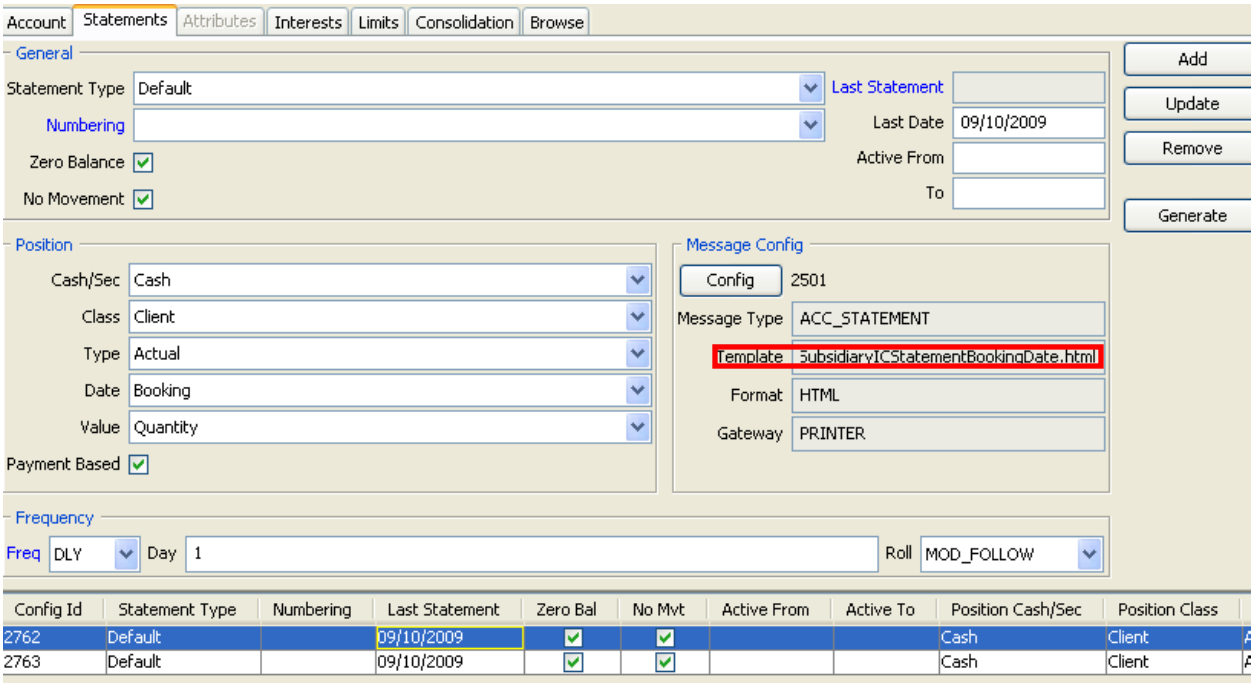
Please note that we also include the message keyword **INTEREST_DETAILS** to confirm the interest on the subsidiary internal call account using the layout below:

INTERESTS DETAILS

VALUE DATE	TYPE	INTEREST AMOUNT	INTEREST RATE	INTEREST TOTAL	INTEREST POSITION

Same logic applies for Booking Date Statements but in that case the template only displays the Booking and Settle Dates.

Call Account Statement Configuration for booking Date HTML Statement



Balance and movement details (only Booking and Settle Dates are displayed).

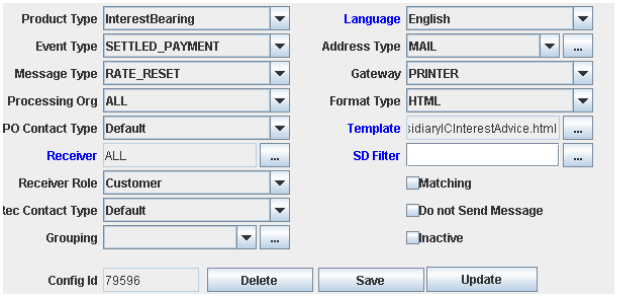
BOOKING DATE	SETTLE_DATE	TRAN REF	DETAILS	DEBIT	CREDIT	BALANCE
09/10/2009	09/10/2009		OPENING BALANCE			0.00
09/10/2009	09/10/2009	10004221	Loan 09/10/2009-09/11/2009-4.00000%		205.93	205.93CR
09/14/2009	09/14/2009	10004202	CASHPOOLING BBVA/3999-010-0468309	205.93		0.00
09/10/2009	09/10/2009		CLOSING BALANCE			0.00

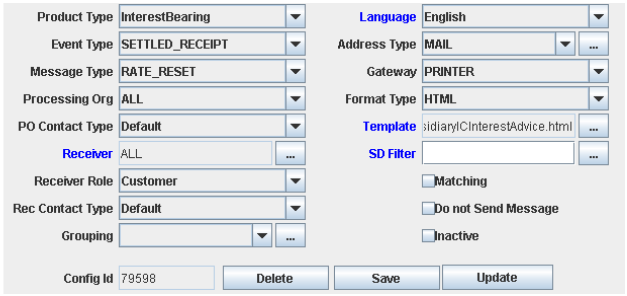
5.4 Subsidiary Internal Interest Advice Message

A dedicated HTML template 'SubsidiaryICInterestAdvice.html' allows confirming the amount of interests and the related interest details (daily position/rate used to compute daily interest/daily interest amount/total amount over the period).

This message must be triggered by the **Interest Bearing** product type, interests for internal call accounts being generated using the ACCOUNT_INTEREST Scheduled Task that creates or updates (daily) Interest Bearing trades for the interest period for each subsidiary call account.

Once interests transfer are **settled** we will trigger the Interest Advice html message generation as stated below (please note that the triggering events should be payment and receipt):





An example of the layout is presented below:

Date: April 24, 2018
To: Delete during implementation
Address: 99 Park Ave Rm 930
New York NY
UNITED STATES
Our Reference: 78430

InterestBearing Payment Advice

Dear Marion Fraser,

We are pleased to advice, for value date, , that you will .

INTERESTS DETAILS

VALUE DATE	TYPE	INTEREST AMOUNT	INTEREST RATE	SPREAD	FINAL RATE	INTEREST TOTAL	POSITION CHANGE	INTEREST POSITION
02/01/2018	INTEREST	602.71	1.31	0.005	1.315	602.71	0.00	16,500,000.00
02/02/2018	INTEREST	605.00	1.315	0.005	1.32	1,207.71	0.00	16,500,000.00
02/03/2018	INTEREST	605.00	1.315	0.005	1.32	1,812.71	0.00	16,500,000.00
02/04/2018	INTEREST	605.00	1.315	0.005	1.32	2,417.71	0.00	16,500,000.00
02/05/2018	INTEREST	570.62	1.24	0.005	1.245	2,988.33	0.00	16,500,000.00
02/06/2018	INTEREST	575.21	1.25	0.005	1.255	3,563.54	0.00	16,500,000.00
02/07/2018	INTEREST	570.62	1.24	0.005	1.245	4,134.17	0.00	16,500,000.00
02/08/2018	INTEREST	571.08	1.241	0.005	1.246	4,705.25	0.00	16,500,000.00
02/09/2018	INTEREST	571.54	1.242	0.005	1.247	5,276.79	0.00	16,500,000.00
02/10/2018	INTEREST	571.54	1.242	0.005	1.247	5,848.33	0.00	16,500,000.00
02/11/2018	INTEREST	571.54	1.242	0.005	1.247	6,419.88	0.00	16,500,000.00
02/12/2018	INTEREST	571.08	1.241	0.005	1.246	6,990.96	0.00	16,500,000.00
02/13/2018	INTEREST	571.08	1.241	0.005	1.246	7,562.04	0.00	16,500,000.00
02/14/2018	INTEREST	571.54	1.242	0.005	1.247	8,133.58	0.00	16,500,000.00
02/15/2018	INTEREST	572.00	1.243	0.005	1.248	8,705.58	0.00	16,500,000.00
02/16/2018	INTEREST	571.54	1.242	0.005	1.247	9,277.12	0.00	16,500,000.00
02/17/2018	INTEREST	571.54	1.242	0.005	1.247	9,848.67	0.00	16,500,000.00
02/18/2018	INTEREST	571.54	1.242	0.005	1.247	10,420.21	0.00	16,500,000.00
02/19/2018	INTEREST	572.00	1.243	0.005	1.248	10,992.21	0.00	16,500,000.00
02/20/2018	INTEREST	561.46	1.22	0.005	1.225	11,553.67	0.00	16,500,000.00
02/21/2018	INTEREST	572.00	1.243	0.005	1.248	12,125.67	0.00	16,500,000.00
02/22/2018	INTEREST	572.46	1.244	0.005	1.249	12,698.12	0.00	16,500,000.00
02/23/2018	INTEREST	561.12	1.22	0.005	1.225	13,259.24	-10,000.00	16,490,000.00
02/24/2018	INTEREST	561.12	1.22	0.005	1.225	13,820.36	0.00	16,490,000.00
02/25/2018	INTEREST	561.12	1.22	0.005	1.225	14,381.48	0.00	16,490,000.00
02/26/2018	INTEREST	2.29	0	0.005	0.005	14,383.77	0.00	16,490,000.00
02/27/2018	INTEREST	506.15	1.1	0.005	1.105	14,889.92	0.00	16,490,000.00
02/28/2018	INTEREST	505.69	1.099	0.005	1.104	15,395.62	0.00	16,490,000.00

RATE CHANGE DETAILS

No Rate Change

Please refer to our reference number in your correspondence concerning this payment.

5.5 Subsidiary Statement of Holdings Internal Message

We also provide the ability to send a statement of holdings for each subsidiary (e.g. All FX & MM deals + Balance of Internal Account, etc.).

For this functionality, please refer to the Client Custody Management User Guide.

Swift MT101 Payment Message

6.1 MT101 Setup

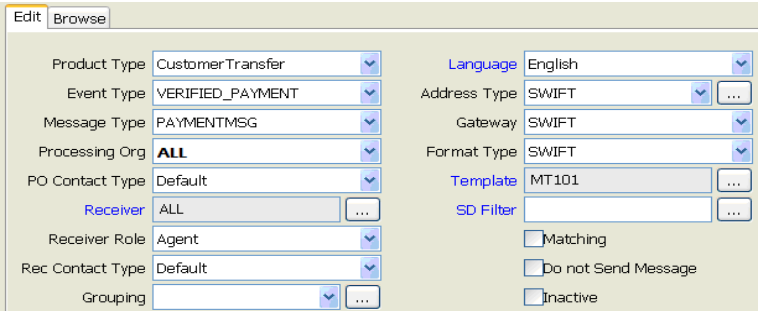
6.1.1 Individual MT101

Message Setup

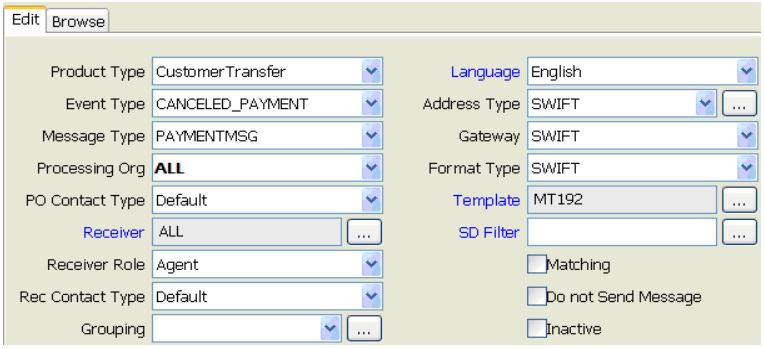
Go to **Configuration > Message & Matching > Message Set-up** and define the template MT101 for Payment Messages.

An example is presented below for the product Customer Transfer

VERIFIED_PAYMENT



CANCELED_PAYMENT to generate Swift MT192

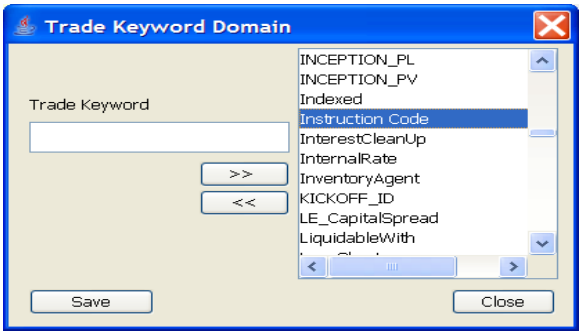


[NOTE: For MT101 “receipts”, you need to use the template MT101SAP]

Instruction Code

You need to define a new trade keyword “Instruction Code” to handle custom Tag 23E.

To do so, go to **Configuration > Domain Values**, choose the domain tradeKeyword and add the value Instruction Code.



Then add the trade keyword Name to the domain XferAttributes.

This trade keyword will be automatically copied as a Xfer Attribute.

Default value for that keyword is URGP.

The user can change the default by changing the keyword or using the UPDATE_MT101TAGS transfer workflow action.

6.1.2 Grouped MT101

The setup presented below is only needed if you want to use the grouping functionality.

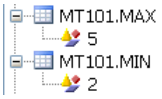
If only individual MT101 are used, please refer to the MT101 individual section.

Domain Values

Select domainName and define the values TemplateName.MIN and TemplateName.MAX (MT101.MIN and MT101.MAX).

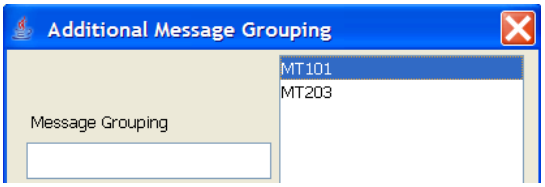
Then, go to the domain MT101.MIN and add (in its value field) the minimum number of underlying MT101 to group (value must be set to 2) and in the domain MT101.MAX the maximum number of underlying MT101 to group.

These TemplateName.MIN and TemplateName.MAX domains are used by the system to determine the number of underlying MT101 to include into the multiple MT101Grouped.



Message Grouping

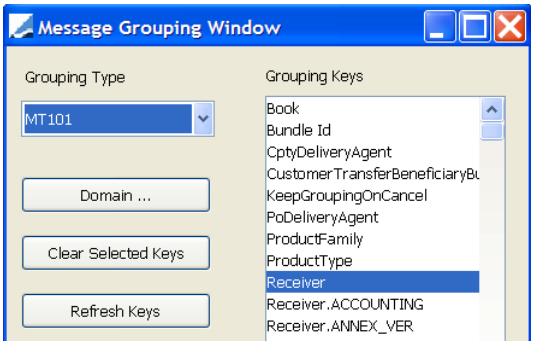
Go to **Configuration > Message & Matching > Message Grouping**, select the Domain button which opens the additional message grouping window and create the new message grouping MT101 as shown below.



Grouping Keys

Then define the MT101 **grouping keys**. These keys will be used by the Message Engine to group the underlying MT101 into the GroupedMT101.

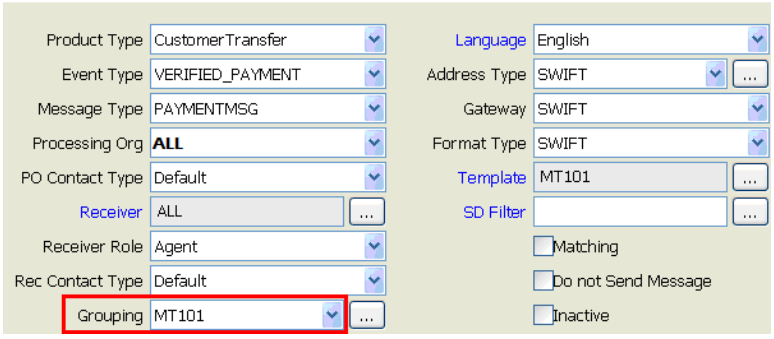
Select the Receiver + Settlement Ccy + Value Date as grouping keys.

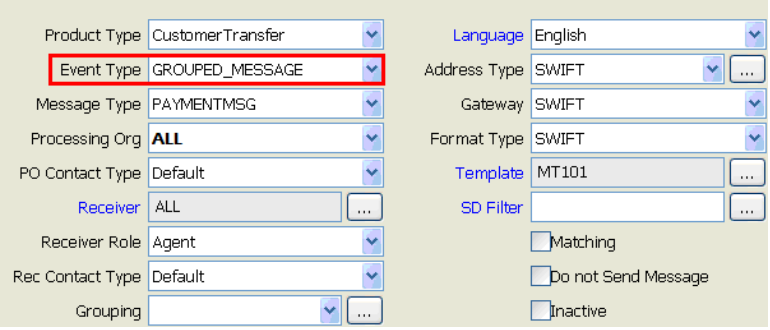


Grouped MT101 Message Setup

Go to **Configuration > Message & Matching > Message Setup** and define the message configuration to handle the grouping of MT101.

An example is presented below for a Customer Transfer product type:





Message Workflow

Specific actions and rules must be added on your message workflow to handle the message grouping function.

First, the action GROUP must be added to move all underlying into a GROUPED status when the grouped MT101 is created. The GROUPED_MESSAGE event triggering the creation of Grouped MT101 is generated by the MESSAGE_GROUPING scheduled task. When that grouped message is created, the system applies the action specified in the scheduled task (GROUP in our configuration). See the MESSAGE_GROUPING scheduled task section below for more details.

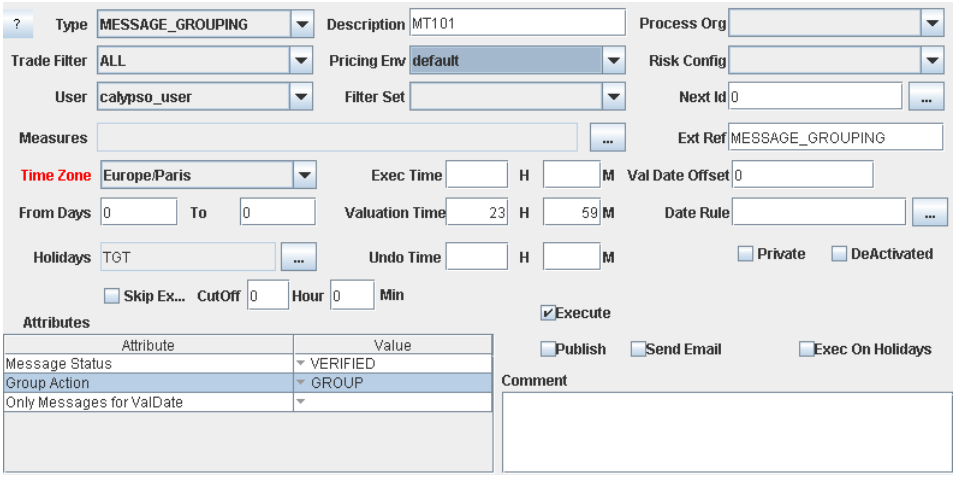
In order to manage AMEND and/or CANCEL actions that can be done on underlying trades/transfers for which MT101 are underlying of a Grouped MT101 - before or after the SEND of the Grouped Message - we need to add the AmendGroup and RemoveGroup message rules, in conjunction with the “KeepGroupingOnCancel” message grouping key (when used).

An example of message workflow for message grouping is presented below:

Subtype	Orig Status	Action	Resulting Status	Rules	Filter
PAYMENTMSG	GROUPED	AMEND	VERIFIED	AmendGroup	
PAYMENTMSG	GROUPED	CANCEL	CANCELED	RemoveGroup	
PAYMENTMSG	GROUPED	SEND	SENT		Grouped MT101
PAYMENTMSG	VERIFIED	GROUP	GROUPED		
PAYMENTMSG	VERIFIED	UNGROUP	VERIFIED		

The GroupedMT101 Static Data Filter is defined as MSG_ATTRIBUTE.Group IN MT101 to return true only for Grouped MT101.

MESSAGE_GROUPING scheduled task



Attribute	Description
Message Status	Specify the Underlying Message Status to be taken into account when running the MESSAGE_GROUPING Scheduled Task. In our example, Message Status = VERIFIED. This means the Scheduled Task will only select VERIFIED Messages marked with the Message Attribute Grouping = MT101.
Group Action	Specify the Message Workflow Action to be applied to messages to be grouped under MT101.
Only Messages for ValDate	By default, the value of that attribute is set to false. This means when running the Scheduled Task MESSAGE_GROUPING we try to group all underlying message marked as Grouping = MT101 (with the same keys) till Scheduled Task Valuation Date. The criteria is the Original Date of the underlying message compare to Scheduled Task Valuation Date. When set to true, we only try to GROUP underlying messages marked as Grouping = MT101 with the same keys when underlying message original date = Scheduled Task Valuation Date.

UPDATE_TAGS and UPDATE_MT101_TAGS Function

When using the customer transfer product, Tag 70 (Remittance Information) and 71A (Details of Charges) are driven by the trade fields Remittance Information and Details of Charges.

When using the transfer action ASSIGN for 'nostro' transfer associated to a customer transfer trade, the system pops-up the trade window allowing the user to change the 'nostro' SDI and Tag70/71A to impact the MT101.

For other product types (eg. FX, Cash), when the user wants to change the default Tag70 and 71A driven by the CounterParty SDI attributes, he will have to use the UPDATE_TAGS transfer workflow action that will pop-up a window to change these values.

In addition, a new transfer workflow action UPDATE_MT101TAGS has been added to overwrite the default value of the trade keyword 'Instruction Code' (Tag 23E) also propagated as a XferAttribute.

Forecasts and In-House Transactions

7.1 Subsidiary Forecasts

Forecast transactions will be entered or interfaced into the system as **Simple Transfer** with a specific **Transfer Type = FORECAST**.

The usage of the **Transfer Type = FORECAST is critical** as the scheduled task INC_CASH_STATEMENT retrieves the list of transfers to exclude from the inventory position based on this type.

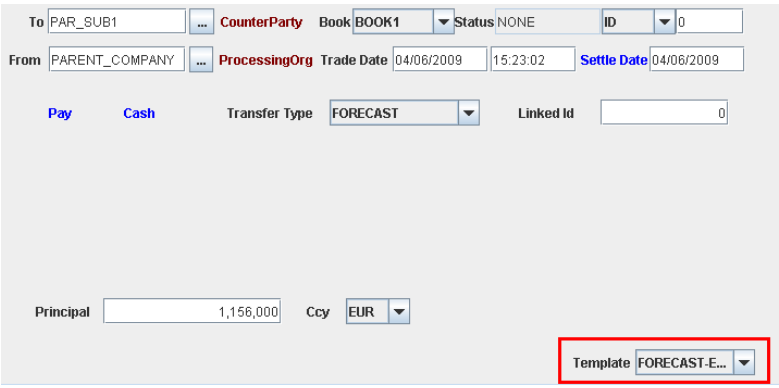
FORECAST transfers will impact the Processing Organization Bank Account.

To manually enter a Forecast within Calypso, the user can save a Simple Transfer following the steps described below:

- Let's assume PAR_SUB1 wants to report a Forecast movement of – 1,156,000.00 EUR value 04/06/2009 on its Barclays London Sub-Account;
- To enter that deal in the system, go to **Processing > Accounting Operations > Simple Transfer** and open the trade screen;
- The user has to enter the Book (defaulted from the user default - the Processing Organization field is automatically populated with the Processing Organization associated to the Book), Counterparty (= Subsidiary), Settle Date, Direction (Pay/Receive), Transfer Type = **FORECAST**, select the right Currency and enter the Amount.

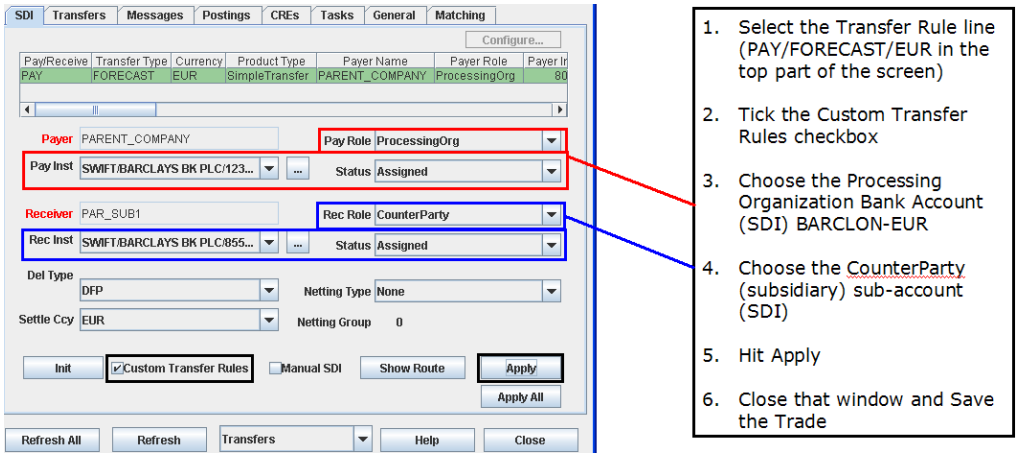
Please note the possibility to define, for instance, 'Forecast-EUR', 'Forecast-USD', 'Forecast-GBP' **trade templates** with Transfer Type defaulted to FORECAST.

An example is presented below:



- Before saving the trade, open the **BO Window** (Trade menu **Back Office > BO Window**) and **Assign SDI** to be applied by the system to impact the appropriate processing organization bank account.

An example is presented below:



The transfer engine creates automatically the related FORECAST transfer hitting the selected PO Nostro Account, as shown below:

Settlements

Transfer_id	Transfer Type	Product Type	Xfer Pay/Rec	Transfer Amount	SettleCurrency	Settle Date	GL Account	Payer Inst_Status	Receiver Inst Status	PO Settle Method	PO Agent
88166	FORECAST	SimpleTransfer	PAY	(1,156,000.000)	EUR	04/06/2009	BARCLON-EUR	Assigned	Assigned	SWIFT	BARCLAYS BK PLC

7.2 In-House Transactions

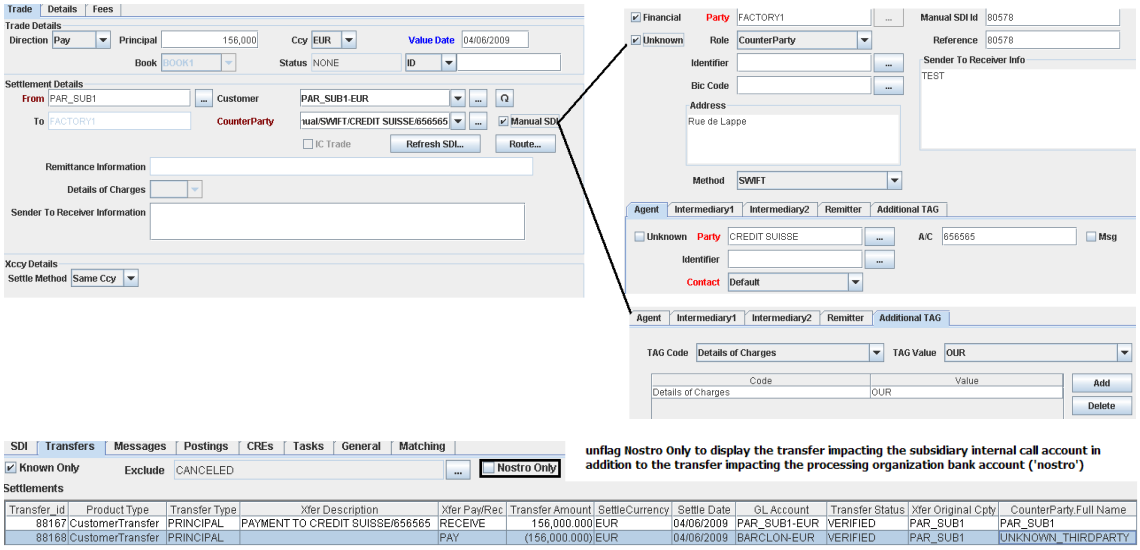
7.2.1 Payment on behalf of

This category covers the thirdparty payments. This type of transaction must be entered as a Customer Transfer, possibly using ManualSDI.

For more information around Customer Transfer setup and Manual SDI, please refer the Client Custody Management User Guide.

To manually enter a thirdparty payment within Calypso, the user must save a Customer Transfer. An example is presented below:

- Let's assume PAR_SUB1 wants to pay an invoice to Factory1 (which is not a Legal Entity known in the system) for 156,000.00 EUR value 04/06/2009
- To enter that deal in the system, the user will have to go to **Processing > Accounting Operations > Customer Transfer** and enter a customer transfer on the subsidiary call account (PAR_SUB1-EUR) using the ManualSDI function where you can add an 'unknown' beneficiary and its settlement instructions used only for that payment. Please note that when the beneficiary and its instructions are known in the system, there is no need to define ManualSDI, the user only picks the payment instructions from the list of valid SDI for that beneficiary.



Please note that a Customer Transfer Trade generates two transfers of different category:

- One transfer impacting the subsidiary call account which is used to:
  - Impact the subsidiary inventory position (subsidiary view)
  - Trigger the subsidiary bank account statements
  - Generate interest bearing trades for that subsidiary call account
- One transfer impacting the processing organization bank account (nostro) which is used to impact/trigger:
  - Payment Messages (MT101/210)
  - Internal BO Position
  - Cash Settle Postings

For more information around Customer Transfer setup and Manual SDI, please refer the Client Custody Management User Guide.

7.2.2 Intercompany Transactions

This category includes:

- Inter-company transaction: transfer of cash from one internal company (call) account to another internal company (call) account in the same currency
- Inter-company FX transaction: transfer of cash between two internal company (call) accounts in a different currency
- Intra-company FX transaction: transfer of cash between two internal company (call) accounts in a different currency, both belonging to the same subsidiary

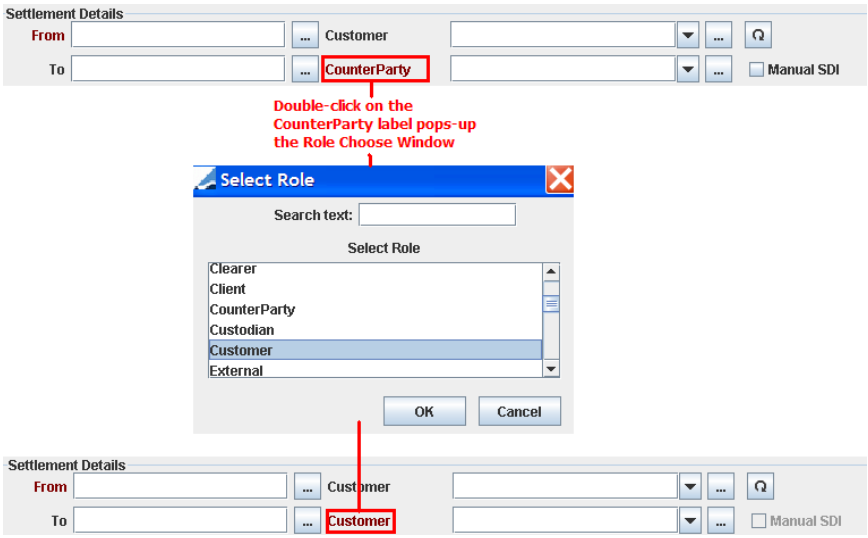
Inter-company transaction in the same currency

This type of transaction must be entered as a **Customer Transfer**.

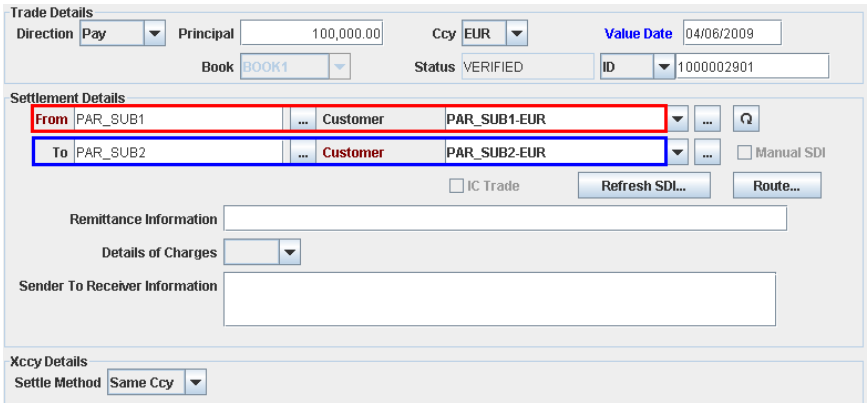
To do so, go to **Processing > Accounting Operations > Customer Transfer** and enter a **unique** customer transfer deal between two Account Holders (AccountHolderRole = Customer in the example below).

This can be achieved by changing the default role 'CounterParty' to Customer (= AccountHolderRole) for the second party involved in the customer transfer trade screen.

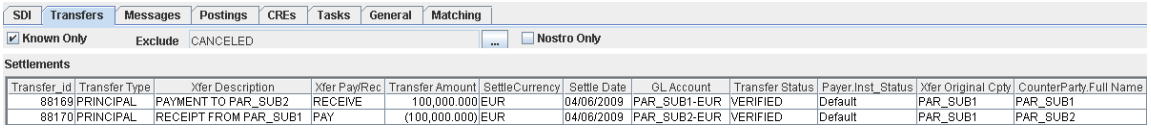
To do so, open the customer transfer trade window, double-click on the CounterParty role and select the AccountHolder role (Customer in the example below). You can save that configuration as a Customer Trade Template 'InterCompany' for instance.



Once done, you can enter the trade details. An example is presented below, where we assume that the subsidiary PAR_SUB1 transfers EUR 100,000 from its internal EUR account to the subsidiary PAR_SUB2 internal EUR account value 04/06/2009. This requires a unique customer transfer deal as stated below:



To display the transfers generated for that internal deal, go to ***Back Office > BO Window***, open the Transfer panel and **untick the Nostro Only** flag. Transfers only impact the subsidiary call accounts, no impact on the processing organization bank account as this is an internal deal.



Transfer_id	Transfer Type	Xfer Description	Xfer Pay/Rec	Transfer Amount	SettleCurrency	Settle Date	GL Account	Transfer Status	Payer.Inst_Status	Xfer Original Cpty	CounterParty.Full Name
88169	PRINCIPAL	PAYMENT TO PAR_SUB2	RECEIVE	100,000.000	EUR	04/06/2009	PAR_SUB1-EUR	VERIFIED	Default	PAR_SUB1	PAR_SUB1
88170	PRINCIPAL	RECEIPT FROM PAR_SUB1	PAY	(100,000.000)	EUR	04/06/2009	PAR_SUB2-EUR	VERIFIED	Default	PAR_SUB1	PAR_SUB2

Please note that the transfer engine requires a “PIVOT-EUR” processing organization SDI (as defined in Section 2 to match the Internal call account SDI for both subsidiaries.

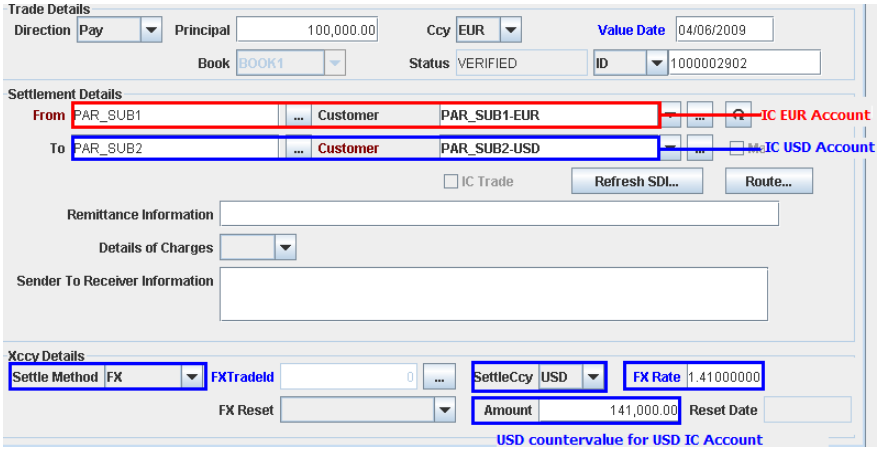
Cross-Currency Intercompany transaction

This type of transaction must be entered as a **XCcy Customer Transfer**.

To do so, go to **Processing > Accounting Operations > Customer Transfer** and enter a **unique** Cross XCcy customer transfer deal between two Account Holders (AccountHolderRole = Customer in the example below).

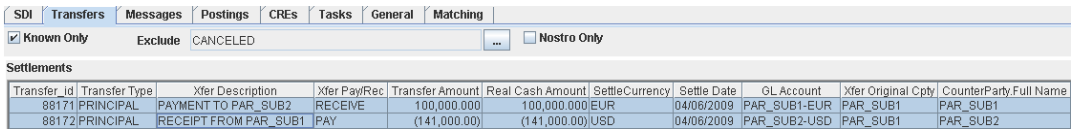
To do so, you can define a ‘XCcyInterCompany’ trade template to default the role ‘CounterParty’ to Customer (= AccountHolderRole) for the second party involved in the customer transfer and fix the Settle Method = FX in the XCcy Details part of the window.

Once done, you can enter the trade details. An example is presented below, where we assume that the subsidiary PAR_SUB1 transfers the USD countervalue of EUR 100,000 from its internal EUR account to the subsidiary PAR_SUB2 internal USD account value 04/06/2009. PAR_SUB1 EUR internal (call) account is debited for EUR 100,000 and PAR_SUB2 USD internal (call) account is credited for USD 141,000. This requires a unique customer transfer deal as stated below:



To display the transfers generated for that internal deal, go to **Back Office > BO Window**, open the Transfer panel and **untick the Nostro Only** flag.

Transfers only impact the subsidiary call accounts, no impact on the processing organization bank account as this is an internal deal.



Transfer_id	Transfer Type	Xfer Description	Xfer Pay/Rec	Transfer Amount	Real Cash Amount	SettleCurrency	Settle Date	GL Account	Xfer Original Cpty	CounterParty.Full Name
88171	PRINCIPAL	PAYMENT TO PAR_SUB2	RECEIVE	100,000.000	100,000.000	EUR	04/06/2009	PAR_SUB1-EUR	PAR_SUB1	PAR_SUB1
88172	PRINCIPAL	RECEIPT FROM PAR_SUB1	PAY	(141,000.00)	(141,000.00)	USD	04/06/2009	PAR_SUB2-USD	PAR_SUB1	PAR_SUB2

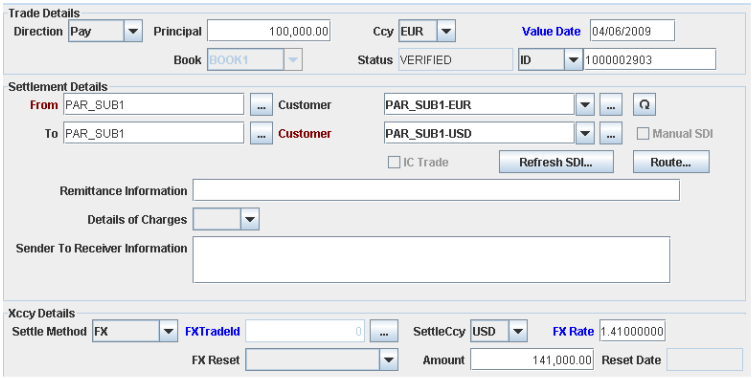
Please note that the transfer engine requires a “PIVOT-EUR” and “PIVOT-USD” processing organization SDIs (as defined in Section 2) to match the Internal call account SDI for both subsidiaries.

Cross-Currency Intra-company transaction

This type of transaction must also be entered as a XCcy Customer Transfer.

No difference with the preceding example, except that the transfer of funds is done between two internal (call) accounts of the same subsidiary.

An example is presented below, where we assume that the subsidiary PAR_SUB1 transfers the USD countervalue of EUR 100,000 from its internal EUR account to its internal USD account, value 04/06/2009. PAR_SUB1 EUR internal (call) account is debited for EUR 100,000 and PAR_SUB1 USD internal (call) account is credited for USD 141,000. This requires a unique customer transfer deal as stated below:

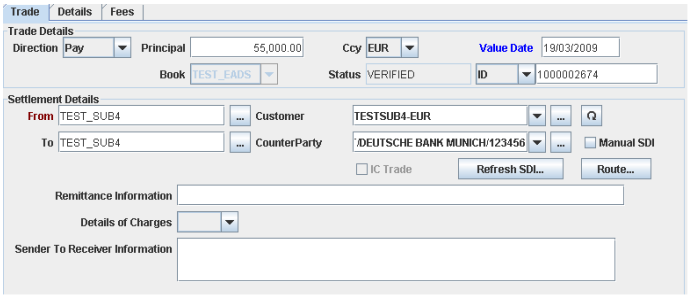


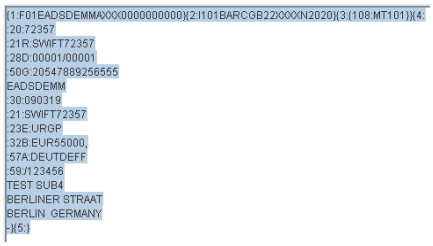
Manual Cash Pooling

Manual Cash Pooling entered by the Back Office on behalf of manually cash-pooled subsidiaries will be saved as Customer transfers, exactly as an automatic cash pooling deal generated from the incoming MT940.

To be able to enter a Manual Cash Pooling transaction, you must initially define for the related subsidiary an external ‘SWIFT’ SDI materializing the not managed sub-account at the external agent and a call account for that currency (the system will create the related AccountHolder SDI automatically). Please refer to section 2 of this document for details.

Once done, you can save the manual cash pooling entry as a standard customer transfer deal. An example is presented below between TEST_SUB4 as a Customer (to pick the Internal subsidiary call account) and TEST_SUB4 as a CounterParty (to select the external subsidiary not managed account – DEUTSCHE BANK MUNICH/Acct# 123456 in our example):





Settlements

	Xfer Description	Transfer Type	Product Type	Transfer Amount	Transfer Status	PO Settle Method	Value Date	SettleCurrency	Xfer Pay/Rec	PO Agent	Xfer Their Agent	GL Account
0	PAYMENT TO DEUTSCHE BANK MUNICH/123456	PRINCIPAL	CustomerTransfer	55,000.000	VERIFIED	Internal	19/03/2009	EUR	RECEIVE	BARCLAYS BK PLC	TEST_EADS	TESTSUB4-EUR
0		PRINCIPAL	CustomerTransfer	(55,000.000)	VERIFIED	SWIFT	19/03/2009	EUR	PAY	BARCLAYS BK PLC	DEUTSCHE BANK MUNICH	BARLD-EUR

This Customer Transfer Trade generates two transfers:

- One transfer impacting the subsidiary call account which is used to:
  - Impact the subsidiary inventory position (subsidiary view)
  - Trigger the subsidiary bank account statements
  - Generate interest bearing trades for that subsidiary call account
- One transfer impacting the processing organization bank account (nostro) which is used to impact/trigger:
  - Payment Messages (MT101/210) => See MT101 above
  - Internal BO Position
  - Cash Settle Postings

For more information around Customer Transfer setup and Manual SDI, please refer the Client Custody Management User Guide.

There is no forecast for this type of subsidiaries, and consequently no substitution process for the Back Office Position.

The reconciliation of the Manual Cash Pooling ‘nostro’ transfer falls into the ‘nostro’ category (sub-MT940 versus nostro transfer).

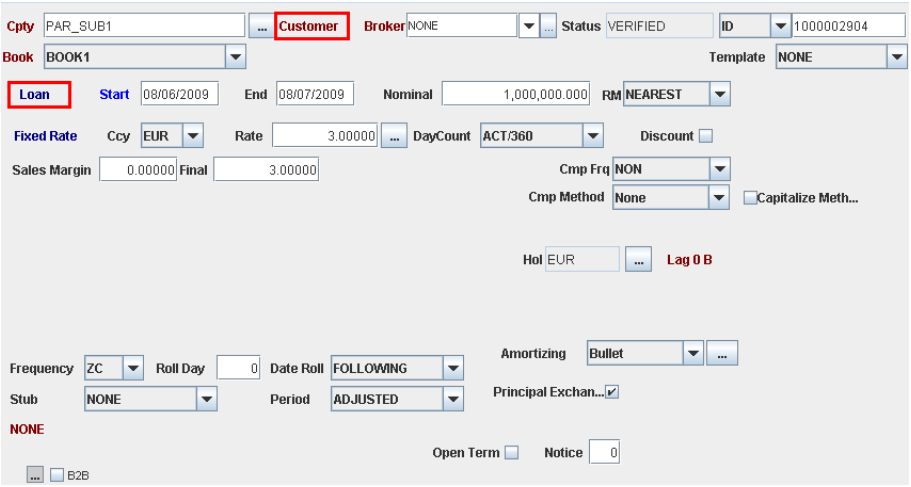
The transfer workflow rule “SettleLinkedDDA” must be put on the transfer SETTLE action to automatically move the subsidiary call account transfer of that deal in a SETTLED status when the related processing organization ‘nostro’ transfer is matched (settled) with the MT940 entry.

7.2.3 “In-House” Market Transactions with a Subsidiary

All market transaction types can be done with a subsidiary.

The internal (or in-house) transaction ordered by the subsidiary must be entered in calypso with the trade role = Account Holder in order to select the subsidiary call account SDI and generate only internal transfers.

An example is presented below for a Loan done with the subsidiary PAR_SUB1. Please note that trade details and related transfers are always shown from the processing organization perspective.



Related transfers impacts the subsidiary internal call account only

Settlements

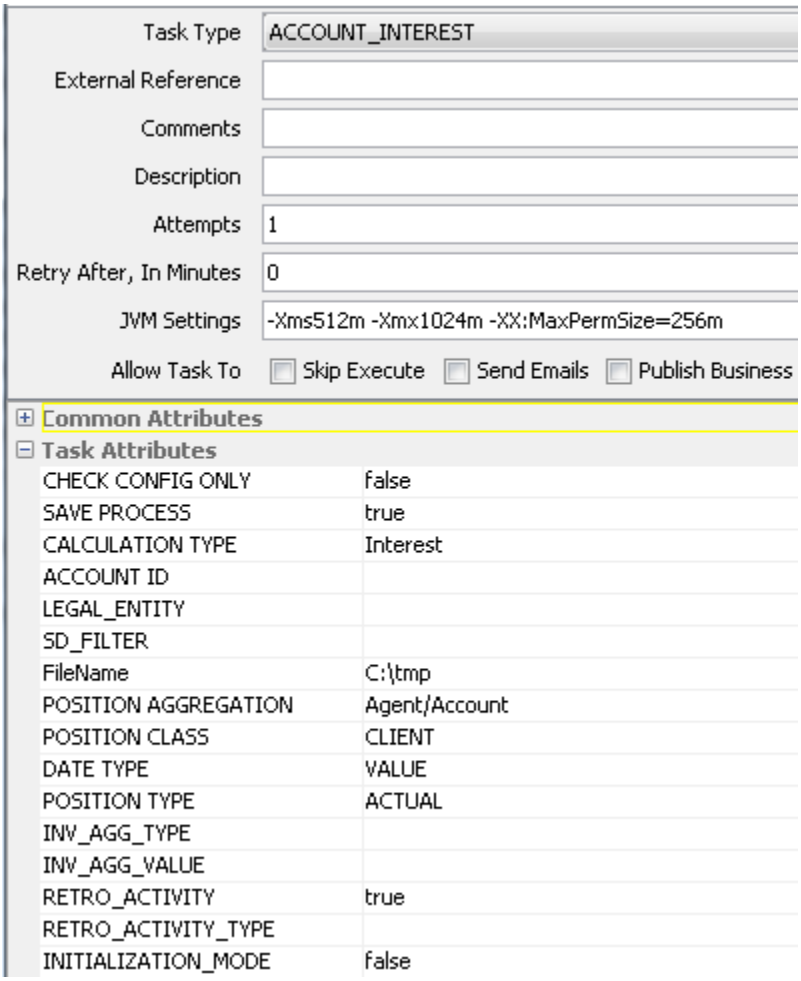
Transfer_id	Transfer Type	Product Type	Xfer Description	Xfer Pay/Rec	Transfer Amount	SettleCurrency	Settle Date	GL Account	Xfer Original Cpty	Transfer Status
88175	NONE	Cash	RECEIPT EUR CASH	PAY	(1,000,000.000)	EUR	08/06/2009	PAR_SUB1-EUR	PAR_SUB1	VERIFIED
88176	NONE	Cash	PAYMENT EUR CASH	RECEIVE	1,002,500.000	EUR	08/07/2009	PAR_SUB1-EUR	PAR_SUB1	VERIFIED

7.2.4 Interest Bearing Transactions

Interests for Call Accounts are calculated based on the Interest Bearing Configuration.

Interest bearing trades are based on the back office position computed by the Inventory engine on accounts associated with an interest bearing rule.

You can process interest computation from the Interests panel in the Accounts window by clicking the Process Interest button, or you can run the ACCOUNT_INTEREST scheduled task in **Configuration > Scheduled Tasks**.



Select a trade filter and a pricing environment in the Common Attributes.

Then, set the Task Attributes to be able to generate Interest Bearing Trades for Customers.

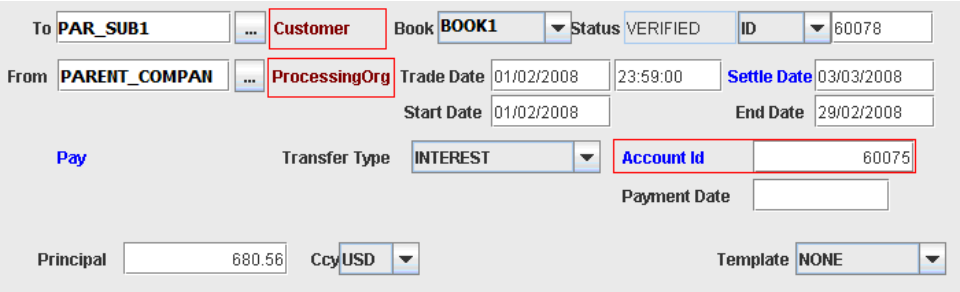
- CHECK CONFIG ONLY - Set to true to check that all the selected accounts have a valid config from the valDate to the valDate+ to day of the Scheduled Task (default is 1 month). The expired config is displayed in the file given as attribute. This produces a report which lists all the interest matrices that are attached to live accounts and that have expired or are due to expire in the coming month (or number of days defined in the "To Days" of the Scheduled Task). If the config is not valid, an ACCOUNT_INTEREST exception is raised and an email is sent
- SAVE PROCESS - If False, the Scheduled Task will only generate a report to be checked by the users. If True, it will generate an Interest Bearing Trade
- CALCULATION TYPE - Refer to the type defined on the matrix; choose Interest, Claim, etc. Interest in this example
- ACCOUNT ID - You can enter one particular account if you want to launch the process on one specific account only
- LEGAL_ENTITY - You can enter one particular Client name if you want to launch the process for one particular Client only

- SD FILTER - Define an SD Filter if you want to restrict the process to one given Processing Organization or Book
- FileName - Set the name and the directory of the report displaying the interests
- POSITION AGGREGATION - Define the type of position aggregation that will be used as the basis of interest calculation: “Agent/Account” or “Book/Agent/Account”. For a Position Class = CLIENT, the Position Aggregation must be set to Agent/Account, at least without the book which is not a criteria for Client Position
- POSITION CLASS - Client: to calculate interest on client accounts (Internal: to calculate interest on NOSTRO Cash accounts)
- DATE TYPE - Trade: to calculate interest based on trade date positions, Settle: to calculate interest based on settle date positions (real settle date), or Value: to calculate interest based on value date (theoretical settle date) => Interest Bearing will have to be computed on a CLIENT / VALUE / ACTUAL Position by Agent/Account
- POSITION TYPE - Actual: for interest calculations, Theoretical: should not be used. Here for flexibility purposes in case one client wants to use it as well, or Failed: for claim calculation
- RETRO_ACTIVITY - True: the process will work in retroactivity mode, or False: the process will not apply any retroactivity rules. Retroactivity process will generate ADJUSTMENT entries and back dated RATE_CHANGE for backdated entries
- INITIALIZATION MODE - Set to True: Normally you set this the first time you calculate interest in the system. This will allow the system to create trades of interest without generating adjustments.

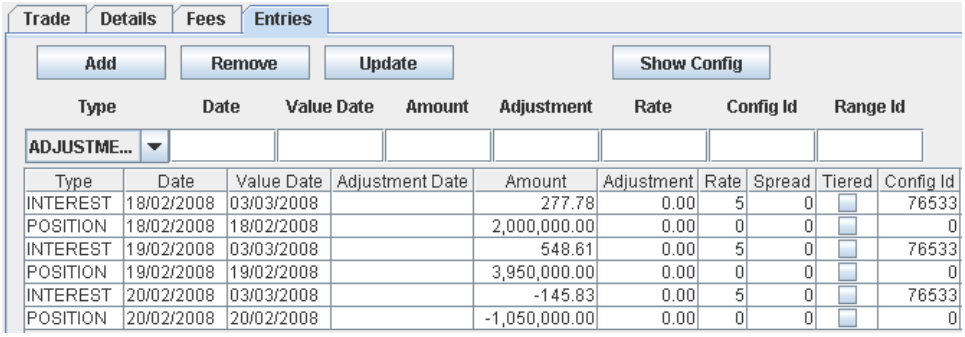
The scheduled task retrieves all SETTLE accounts with the “Interest Bearing” checkbox checked, and the positions corresponding to the selected attributes.

The position is stored each day and retrieves the corresponding interest bearing rule to compute the interest amount. The scheduled task creates (or amends) interest bearing trades.

To view the interest bearing trades, choose **Processing > Accounting Operations > Interest Bearing > Trade** - Open. It opens the trade selector. Click ‘Show Trades’ to view all interest bearing trades.



Go to the Entries panel to display the details of daily interest.



Type	Date	Value Date	Adjustment Date	Amount	Adjustment	Rate	Spread	Tiered	Config Id
INTEREST	18/02/2008	03/03/2008		277.78	0.00	5	0		76533
POSITION	18/02/2008	18/02/2008		2,000,000.00	0.00	0	0		0
INTEREST	19/02/2008	03/03/2008		548.61	0.00	5	0		76533
POSITION	19/02/2008	19/02/2008		3,950,000.00	0.00	0	0		0
INTEREST	20/02/2008	03/03/2008		-145.83	0.00	5	0		76533
POSITION	20/02/2008	20/02/2008		-1,050,000.00	0.00	0	0		0

Transfers (type INTEREST) associated to the interest bearing trade only impact the subsidiary internal (call) account balance. These transfers are re-created as soon as interest bearing trade is amended.

ACCOUNT_INTEREST for Nostro Bank Account

You have the possibility (this is optional) to generate Interest Bearing transactions for Nostro Accounts, using the daily balance as confirmed by the Bank. This allows reconciling the interests paid by the bank when we receive them from the MT940.

To do so, define another ACCOUNT_INTEREST Scheduled Task which is triggered by the **EXTERNAL – ACTUAL – SETTLE DATE** position (the position created from the MT940, based on the BOCashSettlement trade/transfers, to save into Calypso the position as confirmed by the bank, regardless of its reconciliation status)

Attribute	Value
POSITION AGGREGATION	Agent/Account
POSITION CLASS	EXTERNAL
DATE TYPE	SETTLE
POSITION TYPE	ACTUAL

This process will select the bank confirmed position and will use that position to calculate the expected interest bearing on the Nostro Bank Accounts.

Appendix

Integration & Processing of MT940: Mapping algorithm for Automatic Cash Pooling

The default implementation for processing incoming MT940 is the following:

The Scheduled Task INC_CASH_STATEMENT, for incoming MT940 identified as complete and consistent, saves a global statement and splits that statement into sub-statements.

Each sub-statement is typed with one of the following category:

- CashPooling (for cash pooling entries related to not managed sub-accounts)
- CashPoolingPF (for cash pooling entries related to Payment Factory)
- GuaranteeFees (for guarantee fees accounts)
- BankFee (for automatic bank fee creation within a Payment Factory)
- CashSweeping (for automatic transfer agent generation when cash sweeping is automated by the bank itself)
- Nostro (for entries that only need to be reconciled with existing movements/transfers)

The Calypso default algorithm to find a sub-statement type is the following:

- We first retrieve the GL Account using the Statement Tag 25 for a Sender (Agent) and Receiver (Processing Organization) BIC. That information is mapped to the GL Account Attribute “XferAgentAccount” for the PO/Agent;
- We then check if the Nostro identified from Tag 25 is a Payment Factory Nostro (account attribute PaymentFactory = true), a Guarantee Fees nostro (account attribute GuaranteeFees = true) or a ‘Standard’ Nostro (no attribute);
- We then parse the Statement Tag 61 or 86 in order to find an Identifier that is referenced in the Account Mapping Table for that Nostro;
- If there is no matching entry in the Account Mapping Table, then we create a sub-statement marked with CashStatementProcess = Substatement.Nostro;
- If there is one matching entry in the Account Mapping Table:

  The system lookups for a CounterParty SDI such as:

  - Role = Counterparty
  - Agent = Swift Sender
  - Beneficiary = Call Account Holder from Mapping
  - Agent A/C = Account Mapping Identifier
  - Currency = Swift Currency
  - Products is Empty or CustomerTransfer
  - Valid for the Swift Settle Date

Or, if such SDI does not exists or is not marked as Preferred, for the Call Account Holder identified from the Mapping Table, the system lookups for a CounterParty SDI such as:

  - Role = Counterparty
  - Agent = Swift Sender
  - Beneficiary = Processing Organization associated to the master account of the mapping table for that identifier
  - Currency = Swift Currency
  - Products is Empty or CustomerTransfer
  - Valid for the Swift Settle Date

  (This generic SDI defined for the PO with Role = CounterParty is also used for Payment Factory Statements where the subsidiary account number is not confirmed in the statement)

- If such a CounterParty SDI is found and if the Nostro Account from Tag 25 is a Payment Factory Account, then we create a sub-statement marked with CashStatementProcess = SubStatement.CashPoolingPF
- If such a CounterParty SDI is found and if the Nostro Account from Tag 25 is a Guarantee Fees Account, then we create a sub-statement marked with CashStatementProcess = SubStatement.GuaranteeFees
- If such a CounterParty SDI is found and if the Nostro Account from Tag 25 is a standard Nostro, then we create a sub-statement marked with CashStatementProcess = Substatement.CashPooling
- If no CounterParty SDI is found for that identifier, the system lookups for a Processing Organization SDI such as:
  - Role = ProcessingOrg
  - Beneficiary = Swift Receiver
  - Agent = Swift Sender
  - Agent A/C = Account Mapping Identifier
  - Product is Empty or TransferAgent
  - Currency = Swift Currency
  - Valid for the Swift Settle Date
- If a Processing Organization SDI is found, we check if a transfer agent or simple transfers (when the P.O. Destination Account Holder set in the mapping table is different from P.O. Master Account Holder) already exist for that amount/date/ccy/account. If not, we create a sub-statement marked with CashStatementProcess = SubStatement.CashSweeping
- If no Processing Organization SDI is found and if the Nostro Account from Tag 25 is a standard Nostro Account, then we create a sub-statement marked with CashStatementProcess = SubStatement.Nostro

A new message workflow rule named “HandleCashPooling” should be used in order to block the not successfully processed sub-statement; this rule will raise exception in the TaskStation for issues in the mapping and SDI retrieving. The same rule will be used to reprocess these sub-statements once the issues have been manually fixed and will allow creating automatic trades based on the Account Mapping rules.

Calypso default implementation creates the trade depending on the sub-statement type (CashStatementProcess):

- For SubStatement.Nostro, we do not create any trade; this type of entry is automatically reconciled with existing transfers (matching)

- For SubStatement.CashPooling and SubStatement.CashPoolingPF, we create a CustomerTransfer trade such as:
  - CashStatementProcess Trade Keyword = 'Automatic CashPooling' when coming from a standard Nostro (Tag 25 Statement)
  - CashStatementProcess Trade Keyword = 'Automatic CashPooling PF' when coming from a Payment Factory Nostro (Tag 25 Statement)
  - CashStatementProcess Trade Keyword = 'Automatic Guarantee Fees' when coming from a Guarantee Fees Nostro (Tag 25 Statement)
  - CounterParty SDI = CounterParty SDI identified from Tag 61 or 86 matching the A/C field (or generic SDI for PO with Role CounterParty for Payment Factory Cash Pooling)
  - CounterParty = Beneficiary of the found CounterParty SDI
  - Book = Call Account Book (identified from the CounterParty SDI Currency = Swift Currency and Legal Entity/Account Holder = CounterParty)
  - Processing Organization SDI is retrieved based on GL Account identified from the statement Tag 25, Agent (= Swift Sender), Currency (= Swift Currency) and same Settle Method as the CounterParty SDI
  - Customer = CounterParty
  - Customer Account SDI = DDA Account SDI associated to the Call Account defined for that CounterParty and Swift Currency
- For SubStatement.BankFee, we create a Simple Transfer trade such as:
  - CashStatementProcess Trade Keyword = Automatic BankFee
  - Transfer Type = BANKFEE
  - Processing Organization = Swift Receiver
  - Book = derived from the Processing Organization LEAttribute DEFAULT_BOOK
  - CounterParty = derived from the Processing Organization LEAttribute DEFAULT_CPTY
  - Trade Keyword TradeSource = CashStatement
  - Trade Keyword TargetAccountId= Id of the GL Account (SimpleTransfer SDI Selector) to select the Nostro Account as set from the Statement Tag 25
- For SubStatement.CashSweeping, we lookup Transfer Agent (within the same PO) trades such as:
  - ProductType = TransferAgent
  - Status <> CANCELED
  - Processing Organization = Swift Receiver
  - Settle Date = Swift Settle Date
  - Internal Ref = "CASHSWEEPING-yyyyMMDD" (Settle Date)
  - CrossChecked Trade Keyword Empty
  - Currency = Swift Currency
  - Amount = Swift Amount
  - From/To Accounts are the same as the one from Tag 25 and the one linked to the Processing Organization SDI found in the previously described algorithm

- If no Trade found , then we create a new TransferAgent same characteristics as the one searched and marked it with the Trade Keyword CashStatementProcess = Automatic CashSweeping and TransferType = CASHSWEEPING

- If a Trade is found, then we just apply the action AMEND and set the CrossChecked Trade Keyword to true

When Mapping Rules are set with Account Holder of the Master Account ≠ Account Holder of the Destination Account and both Account Holders are Processing Organizations (= Sweeping of Nostro Accounts belonging to two different Processing Organization of the same Group), then we lookup for a couple of Simple Transfer trades such as:

- ProductType = SimpleTransfer
- Status <> CANCELED
- Processing Organization = Swift Receiver
- Settle Date = Swift Settle Date
- Internal Ref = "CASHSWEEPING-yyyyMMDD" (Settle Date)
- CrossChecked Trade Keyword Empty
- Currency = Swift Currency
- Amount = Swift Amount
- From/To Accounts associated to these deals are the same as the one from Tag 25 and the one linked to the Processing Organization SDIs found in the previously described algorithm
- If no Trade found , then we create two Simple Transfer trades (identical but opposite) saved between the DEFAULT_BOOK set on each PO (as LE Attribute) and the other PO with the Role CounterParty. The deals have the same characteristics as the one searched, and are marked with the Trade Keyword CashStatementProcess = Automatic CashSweeping and TransferType = CASHSWEEPING

- If a Trade is found, then we just apply the action AMEND and set the CrossChecked Trade Keyword to true on both Simple Transfers.

- The Transfer Engine then marks the Transfers generated with the XferAttribute ExpectedStatus = SETTLED for:
  - TransferAgent marked with CashStatementProcess Keyword = Automatic CashSweeping and CrossChecked keyword = true;
  - SimpleTransfer marked with CashStatementProcess Keyword = Automatic BankFee
  - CustomerTransfer marked with CashStatementProcess Keyword = 'Automatic CashPooling' or 'Automatic CashPooling PF'
- Using the out-of-the-box CheckToBeSettled transfer workflow rule, the transfers associated to these trades automatically go to a SETTLED Status, impacting the Settled BO Position
- Would the default implementation not be satisfying, the CashStatementHandlerFactory tries to retrieve a Handler respecting the following order:
  - Gateway - ExternalMessage Type - Sender Address Code

- Gateway - ExternalMessage Type - Sender Address Code Without branch
- Gateway - ExternalMessage Type
- Gateway - Sender Address Code
- Gateway - Sender Address Code Without branch
- ExternalMessage Type - Sender Address Code
- ExternalMessage Type - Sender Address Code Without branch
- Sender Address Code
- Sender Address Code Without branch
- If nothing is found then the Default implementation is used.

Creation of one Interface to customize the AccountMappingDialog

```
    public interface AccountMappingDialogCustomizer {
        /**
         * @return the list of Mapping Type supported by the Customizer
         */
        public List<String> getMappingTypeDomain();
        /**
         * Allow to customize the Mapping Type.
         * Individual attributes can be updated/retrieve through the
controller.getAttribute(attributeName)
         * and controller.setAttribute(attributeName, value) methods
         * @return The populated Panel for the given mapping type
         */
        public JComponent getMappingTypePanel(String mappingType,
AccountMappingController controller);
    }
```

This interface needs to be implemented in the apps.refdata.AccountMappingDialogCustomizerImpl.

The standard screen has two toggle Buttons: "Other Account" and "Same As Master”. When the customizer handles new mapping, the "Same As Master" becomes a ToggleSplit Button (Combination of a Split and a MenuButton).

The Entries of the Menu will be the one provided by the Customizer + the standard "Simple Transfer" which existed already.

When selecting the split menu items (except for "Simple Transfer"), a custom Panel will be retrieved from the Customizer.

If the extra parameters are necessary for the mapping type, they will have to be stored/retrieved as AccountMapping attribute.

The communication with the dialog controller will be insured thanks to the setAttribute/getAttribute methods handled in the controller

Example provided:

calypsox.apps.refdata.AccountMappingDialogCustomizerImpl

Creation of API entries in the MT940CashStatementHandler in order to allow customization of the Trade creation

```
protected boolean customCashStatementProcessAttribute(AccountMapping mapping,
                                                      BOMessage miniMessage,
                                                      List errors) throws
MessageParseException {
        return false;
}

protected void customMiniMessageAttribute(SwiftMessage swift,
                                          BOMessage miniMessage,
                                          List errors) throws MessageParseException {
}

protected Trade customTradeCreation(BOMessage miniMessage, List errors)
```

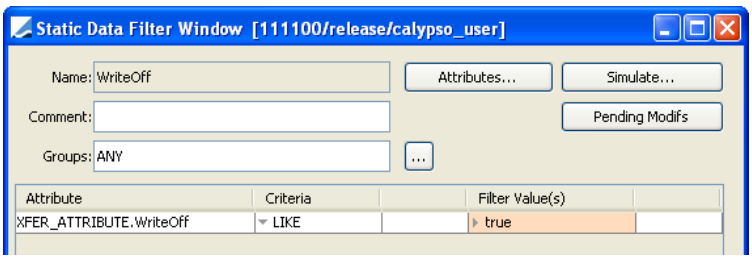
These three methods allow doing the following:

- customCashStatementProcessAttribute: set a Special CashStatementProcess message attribute on the mini message (substatement message)
- customMiniMessageAttribute: allow to create temporarly attributes for special processing. these should then be reused in the customTradeCreation
- customTradeCreation: depending on the CashStatementProcess attribute, we should try to create the Trade, using the custom attributes created in customMiniMessageAttribute

Example provided:

calypsox.tk.util.swiftparser.cashstatement.MT940CashStatementHandler

Partial Match Option - Transfer workflow - example

The transfer workflow presented below is only an example to be used as a starting point. Clients must then adjust based on their business process, especially in terms of manual and automatic write-off.

Orig Status	Action	Resulting Status	Different User	Use STP	Product Type	Rules	Processing Org	Filter
CANCELED	UNMATCH	CANCELED	false	false	ALL		ALL	
CANCELED	UPDATE	CANCELED	false	false	ALL		ALL	
FAILED	AUTOMATCH	SETTLED	false	false	ALL		ALL	
FAILED	CANCEL	CANCELED	false	false	ALL		ALL	
FAILED	MANUALMATCH	SETTLED	false	false	ALL		ALL	
FAILED	PARTIAL_SETTLE	SPLIT	false	false	ALL		ALL	
FAILED	SETTLE	SETTLED	false	false	ALL		ALL	
FAILED	SPLIT	SPLIT	false	false	ALL		ALL	
FAILED	UPDATE	FAILED	false	false	ALL		ALL	
FAILED	WRITEOFF	WRITEOFF	false	true	ALL		ALL	WriteOff
HELD	AUTHORIZE	VERIFIED	false	false	ALL		ALL	
HELD	CANCEL	CANCELED	false	false	ALL		ALL	
HELD	UPDATE	HELD	false	false	ALL		ALL	
INVALID	ASSIGN	CANCELED	false	true	ALL	ApplyDefaultSDI	ALL	
INVALID	CANCEL	CANCELED	false	false	ALL		ALL	
INVALID	UPDATE	INVALID	false	false	ALL		ALL	
MATCHED	UNMATCH	UNMATCHED	false	false	ALL		ALL	
NONE	NEW	INVALID	false	false	ALL	CheckSDI	ALL	
NONE	NEW	PENDING	false	false	ALL	CheckCorrectSDI,PropagateTradeKeyword	ALL	
PENDING	AMEND	PENDING	false	false	ALL		ALL	
PENDING	ASSIGN	CANCELED	false	false	ALL		ALL	
PENDING	AUTHORIZE	VERIFIED	false	true	ALL	CheckNetting	ALL	
PENDING	CANCEL	CANCELED	false	false	ALL		ALL	
PENDING	EXECUTE	SPLIT	false	false	ALL	SecurityNetting	ALL	
PENDING	EXECUTE	VERIFIED	false	false	ALL	SetKnownFlag	ALL	
PENDING	SPLIT	SPLIT	false	false	ALL		ALL	
PENDING	UNSPLIT	CANCELED	false	false	ALL		ALL	
PENDING	UPDATE	PENDING	false	false	ALL		ALL	
SETTLED	CANCEL	CANCELED	false	false	ALL		ALL	
SETTLED	UNMATCH	VERIFIED	false	false	ALL		ALL	
SETTLED	UPDATE	SETTLED	false	false	ALL		ALL	
SPLIT	UPDATE	SPLIT	false	false	ALL		ALL	
SUBSTITUTED	CANCEL	CANCELED	false	false	ALL		ALL	
SUBSTITUTED	UNSUBSTITUTE	VERIFIED	false	false	ALL		ALL	
UNMATCHED	UNMATCH	UNMATCHED	false	false	ALL		ALL	
VERIFIED	AMEND	PENDING	false	false	ALL	CheckKnownFlag	ALL	
VERIFIED	ASSIGN	CANCELED	false	false	ALL		ALL	
VERIFIED	AUTOMATCH	SETTLED	false	false	ALL		ALL	
VERIFIED	CANCEL	CANCELED	false	false	ALL		ALL	
VERIFIED	FAIL	FAILED	false	true	ALL	CheckToBeFailed	ALL	
VERIFIED	HOLD	HELD	false	false	ALL		ALL	
VERIFIED	MANUALMATCH	SETTLED	false	false	ALL		ALL	
VERIFIED	PARTIAL_SETTLE	SPLIT	false	false	ALL		ALL	
VERIFIED	SETTLE	SETTLED	false	true	ALL	CheckToBeSettled	ALL	
VERIFIED	SPLIT	SPLIT	false	false	ALL		ALL	
VERIFIED	SUBSTITUTE	SUBSTITUTED	false	false	ALL		ALL	
VERIFIED	UNMATCH	VERIFIED	false	false	ALL		ALL	
VERIFIED	UNSPLIT	CANCELED	false	false	ALL		ALL	
VERIFIED	UPDATE	VERIFIED	false	false	ALL		ALL