DTCC MarkitServ DSMatch Integration Guide

June 2015 – Fifth Edition

This document guides you through the setup, generation and reconciliation of DTCC payments and DTCC confirmations.

Online help is provided from the individual windows referred to in this document.

Revision date	Edition	Comments
December 2012	First edition	First edition for external module Version 1.2.1.
June 2013	Second edition	Updated for Version 1.5.
May 2014	Third edition	Updated for Version 2.0.
August 2014	Fourth edition	Updated for Version 2.2: Integration of 2014 Credit ISDA Definition.
June 2015	Fifth edition	Updated for Version 2.11.

Contents

Section 1. Overview

The DTCC MarkitServ DSMatch integration with Calypso is three-fold:

- Payment Reconciliation and DTCC Central settlement: Generation of payments and reconciliation with imported payments – as defined in DTCC V4.1 specifications, using CSV files format.
- Confirmation matching:
  - Generation of confirmations and reconciliation with imported confirmations – as defined in DTCC V11.0 specifications for Credit Derivatives, *DTCC V6.0 for Equity Linked Swaps -Equity Perfomance only (service will be deprecated in August 2015 as announced by MarkitServ)*, using FPML files and MQSeries middleware V7.5
  - Possibility to use GTR link for Regulatory Reporting
- Novation Consent: Generation of confirmations and reconciliation with imported confirmations – as defined in DTCC V4.0 specifications for Credit Derivatives

Payment Reconciliation Process

The DTCC payment reconciliation process works in the following manner: you send a file to DTCC containing payments to be made using the scheduled task DTCC_PAYMENT_EXPORT, then you import a file from DTCC that contains the status of the payments, using the scheduled task DTCC_PROCESS_PAYMENTS. The Calypso payments are updated accordingly, and you can then net matched payments.

Confirmation Reconciliation Process

The DSMatch confirmation reconciliation process works in the following manner: you generate DTCC confirmations using the message engine, then you send the confirmations to DTCC using the Sender engine. The DTCC gateway used to send the confirmation is real-time, it uses dedicated lines and the message transfer protocol is MQSeries.

Finally you import confirmations from DTCC in real-time using the Import Message engine (and the MQSeries protocol as for the export). The incoming confirmations are reconciled with the Calypso outgoing confirmations, and based on the status sent by DTCC will be confirmed or not.

Calypso currently supports DTCC confirmations for the products: Credit Default Swap, CDS Index, CDS ABS Index, CDS Index Tranche, CreditDefaultSwaption and CDIndexOption and Equity Linked Swap (Equity Performance only) and for the activities: Trade, Full Termination, Partial Termination, Increase, Assignment, Amendment, and Exit from DTCC.

Note that Confirmation reconciliation for ELS will be deactivated in DTCC in August 31^{s} , 2015.

Credit Confirmation Reconciliation incorporates support for establishing connectivity to the Reporting Repository (DTCC Global Trade Repository) and Triparty ticket model. If clients have subscribed to this service, they will be able to send their regulatory obligation under DFA.

Novation Consent Process

The DTCC novation consent process is based on the same technical architecture as DTCC confirmations.

Section 2. Installation

The DTCC libraries are not part of the core Calypso product. You need to download the DTCC integration module from the download website, calypso-dtcc-2.2.0-c….-rel.zip, that includes files for DTCC Dsmatch:

Calypso Version	jars version 1.5.0.2
V11.1.02	calypso-dtcc-derivserv-protocol-2.X.X-c11102 calypso-dtcc-derivserv-core-2.X.X-c11102
V11.1.04.SP4	calypso-dtcc-derivserv-protocol-2.X.X-c11104sp4 calypso-dtcc-derivserv-core-2.X.X-c11104sp4
V12	calypso-dtcc-derivserv-protocol-2.X.X-c12 calypso-dtcc-derivserv-core-2.X.X-c12
V13	calypso-dtcc-derivserv-protocol-2.X.X-c13 calypso-dtcc-derivserv-core-2.2.X-c13
V14	calypso-dtcc-derivserv-protocol-2.X.X-c14 calypso-dtcc-derivserv-core-2.X.X-c14

1- Connectivity to DTCC (MQ series setup, account opening at DTCC for eligible asset classes) should have been performed.

2- Install DTCC Common jar: calypso-dtcc-common-2.X.X-c…jar

DTCC module uses **DTCC Common jar** as basement: refer to documentation "DTCC Common Guide"

- ⇨ ***You have to download DTCC Common from the website***

  calypso-dtcc-common-2.X.X-c…..jar (included in DTCC integration module)

- ⇨ ***You have to run DTCCCommonSchemaData.xml to create domain values (select GTR Common and GT Domains categories)***
- ⇨ ***You have to execute ReportingAttributes.xml that describes the reporting attributes used for reporting depending on your needs***

The "rel" jars contains the main following folders:

- resources: DTCCXferWorflow.wf, DTCCNovationsTradeWorkflow.wf, DTCCNovationsMsgWorkflows.wf, DTCCMsgWorkflows.wf
- jars
- bin : DTCCSchemaData.xml

The following engines to be run:

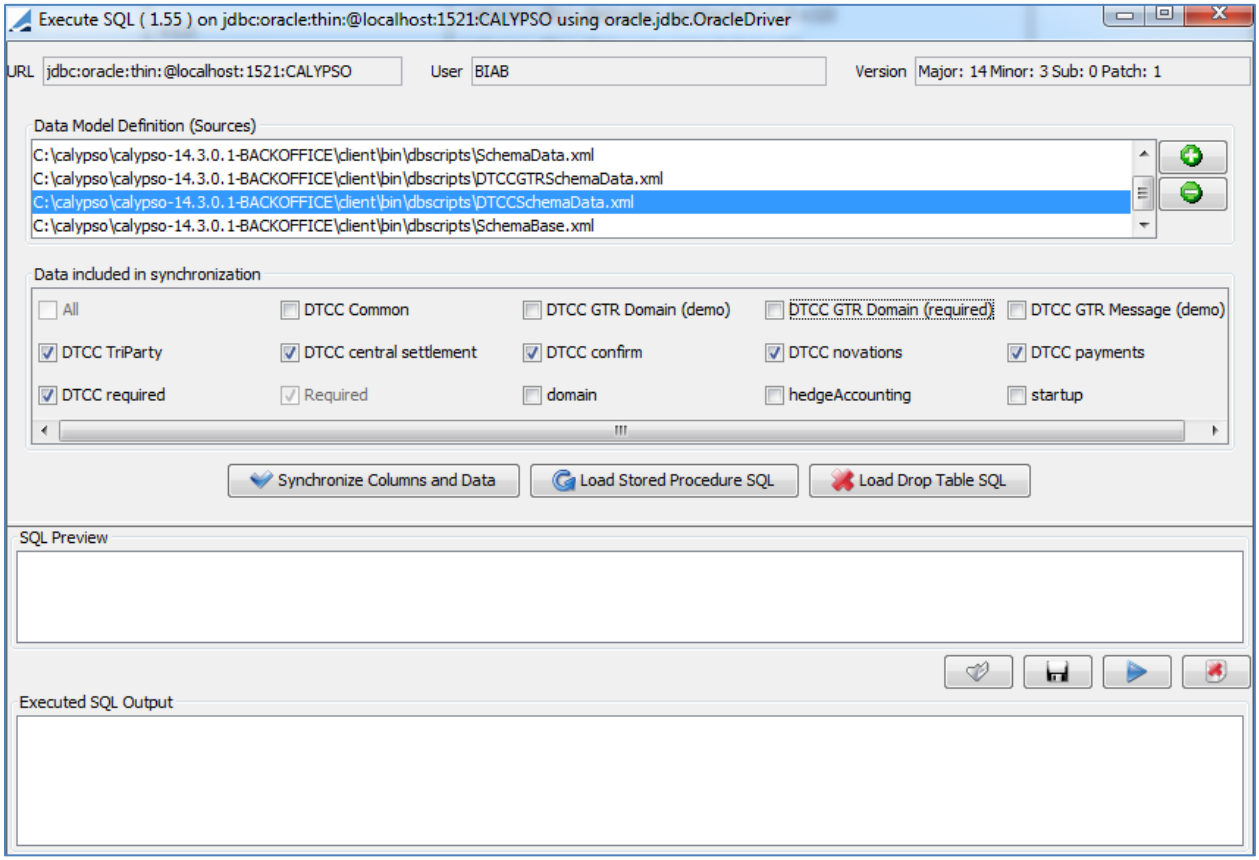
- Message engine, for outgoing message generation
- Sender engine, to send message to MQ
- Import message engines, to import message from MQ

2.1 DTCC Data

Unzip the file, add the required jars to your CLASSPATH. Apply the XML script `DTCCSchemaData.xml` to initialize DTCC static data.

Note that, depending on whether you want to install, you will have to synchronize the corresponding data category:

	DTCC payments	DTCC Confirm	DTCC Novation Consent	Triparty	DTCC Matching
DTCC central settlement	X				
DTCC Confirm		X			
DTCC novations			X		
DTCC payments	X				
DTCC required	X	X	X		X
DTCC Triparty		X			



Static data created by the above XML script:

Domain addressMethod – value DTCC

Domain masterConfirmAdditionalField – values

- isDTCC
- Holidays
- Calculation Agent
- Calculation Agent City

Domain calcAgentCityCode – values: all valid cities and corresponding codes

Domain masterConfirmationType – values: all valid FPML/DTCC master confirmation types, including the new 2014 Credit ISDA Definitions

Domain masterConfirmationTypeMatrix – values: only valid FPML/DTCC master confirmation types which correspond to physical settlement matrix or standard terms. , including the new 2014 Credit ISDA Definitions. This is a subset of the masterConfirmationType domain.

Domain masterConfirmationTypeAPS – values: only valid FPML/DTCC master confirmation types which correspond to Master confirmation types to be considered as Asia/Pacific and Sovereign, including the new 2014 Credit ISDA Definitions. This is a subset of the masterConfirmationType domain.

Domain masterConfirmationRelevantDates - value:

- DTCC_BackloadEffectiveDate
- TerminationTradeDate
- TransferTradeDate

For Master confirmation selection: the relevant date used to compare a trade to the "from" and "to" dates of a MC, will be Max(trade date, "MasterConfirmationRelevantDates")

In order not to have discrepancies in the novation DTCC templates, where the old transaction part of the message needs to match the original confirmation, we will stop using the current templates, and use only short form templates.

Domain MsgAttributes – values:

- Error Code/s
- Incoming Msg Status
- Contra TradeId-PartyReference
- Contra Trade Ref. Supplement Id
- Your Trade Id
- Your Trade Ref. Supplement Id
- PO Super Id
- New PO Super Id
- Incoming Msg Activity
- Incoming Msg Transaction Type
- DK Reason
- Consent Type
- Consent Type Comment
- Contra Super Id
- New Contra Super Id
- PO Desk Id
- Contra Desk Id
- New PO Desk Id
- New Contra Desk Id

Domain workflowRuleTrade – values:

- CheckMasterConfirmation
- CheckDTCCNovation

- CheckNotDTCCNovation
- DTCCCleanupGTRUSI

Domain workflowRuleMessage – values:

For Confirm:

- IncomingReply
- CancelAlleged
- CleanupAlleged
- DKAlleged
- IndexAlleged
- UnIndexDTCC

For DTCC Novation:

- DTCCNovationReply
- IncomingNovationRequest
- CleanupOutgoingNovationRequest
- CleanupIncomingNovationRequest
- DTCCApplyNovation
- SetAttributes

For DTCC Central Settlement:

- DTCCSplitGross
- DTCCUpdateTransfer

2.2 Environment Properties and domains

2.2.1 Calypso trade id management

The environment properties AUTO_FEED_INTERNAL_REF must be set to true.

In the DTCC module, Calypso internal reference is used as the DTCC TradeReferenceNumber and as a result it has to be filled for all trades to be sent to DTCC.

Using internal_reference instead of Calypso trade_id has various advantages. Among them are:

- When implementing Calypso, if you have a collection of trades already confirmed in DTCC in your old trading system, you can set Calypso internal reference to be the trade reference used to confirm these trades in DTCC originally. As a result the switch from the old system to DTCC will be transparent from a DTCC and counterparty point of view.
- Various trade lifecycle events in Calypso such as partial termination, notional increase or novation , terminate the existing trade and create a new one. But because this is still the same deal, you need to keep the same TradeReferenceNumber in DTCC. When performing these actions, the Calypso internal reference is propagated from the old trade to the new one when AUTO_FEED_INTERNAL_REF is true.

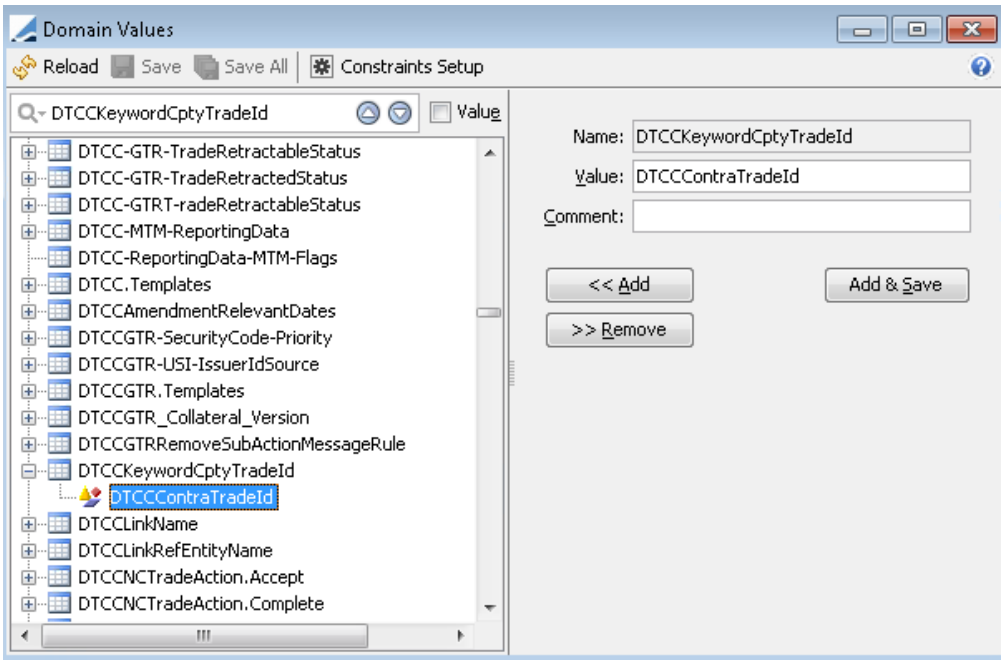
2.2.2 Counterparty trade id management

When an incoming message received from DTCC contains counterparty trade id information, it is stored on the trade itself for future use. Therefore Calypso users are able to query trades knowing only the counterparty trade id or can do trade reporting including this information.

By default, the counterparty trade id (DTCC's "Contra Trade Id") is store in trade external_reference.
If your setup doesn't allow to use this field for this purpose (typically this would be the case when this field is

already used to store FO system trade id when Calypso is a BO only system) then you can ask Calypso to store this information in a trade keyword instead of external_reference.

Domain DTCCKeywordCptyTradeId has been added. If this domain contains a keyword, the counterparty trade id is stored on that keyword. Otherwise, it is stored on the external trade reference. In the example below, it is stored in the "DTCCContraTradeId" keyword.



Section 3. DTCC Payment Matching

The Payment reconciliation process uses a dedicated transfer workflow in order to match transfers with the counterparty. At the end of the process, transfers are either matched or unmatched, matched ones are netted, and both netted and remaining unmatched are finally settled in the "normal" (non-DTCC) workflow.

3.1 Additional Settings

Run DTCCSchemaData.xml. See section 2.1 DTCC Data

In order to create the DTCC transfer workflow, choose Navigator > Configuration > Workflow > Workflow Configuration. Then import the workflow contained in the `DTCCXferWorkflow.wf` file.

3.1.1 Domain Values

The following domain values are created by executing the DTCCSchemaData.xml

Choose Navigator > Configuration > System > Domain Values if you want to modify these domain values.

Domain XferProfileType - value: DTCC.

Domain XferWorkflowType - value: DTCC.

Domain XferAttributes – values:

- DTCCParticipantId
- DTCCUserId
- IndexStartDate
- MatchingSystem
- RefEntity
- RefTradeId
- ScheduledMatDate
- SettleDate
- TradeCcy
- TradeDate
- TradeNotional
- DTCCCashflowId
- DTCCNetPaymentId
- DTCCPmtMsgStatus
- Incoming Msg Status
- Incoming Msg Product Type

MsgAttributes - values:

- DTCCCashflowId
- DTCCNetPaymentId
- PAYRECEIVE_TYPE
- Payment Amount
- Payment Currency
- RecordQualifier
- RecordType
- SettleDate

Domain DTCCProcessPaymentsTransferStatus - The values here define the list of transfer status loaded during the scheduled task DTCC_PROCESS_PAYMENT. For example, MATCHED, UNMATCHED, WAITINGMATCH, WAITINGCANCEL.

Domains DTCCTransferAction.<action name> - This set of domains allows to map DTCC matching statuses with the corresponding action in Calypso (MATCH/UNMATCH). For example:

- DTCCTransferAction.Linked/MisMatched - value: UNMATCH
- DTCCTransferAction.Linked/NoCpty/Input - value: UNMATCH
- DTCCTransferAction.MatchedPayments - value: MATCH

Domain scheduledTask - values:

- DTCC_PAYMENT_EXPORT
- DTCC_PROCESS_PAYMENTS
- EOD_TRANSFER_NETTING

Domain exceptionType - values:

- EX_DTCC_PAYMENT_EXPORT
- EX_DTCC_PROCESS_PAYMENTS

Domain workflowRuleTransfer – value ResetWorkflowType

Domain workflowRuleMessage – values:

- DTCCSplitGrossMessageRule: this rule splits a DTCC Gross Settlement message into separate messages.
- DTCCUpdateTransferDescriptionMessageRule: This rule updates a message's transfer

These are used for Central Settlement.

messageType – values:

- IncomingTIWSettlement
- IncomingTIWPayment

These are used for Central Settlement.

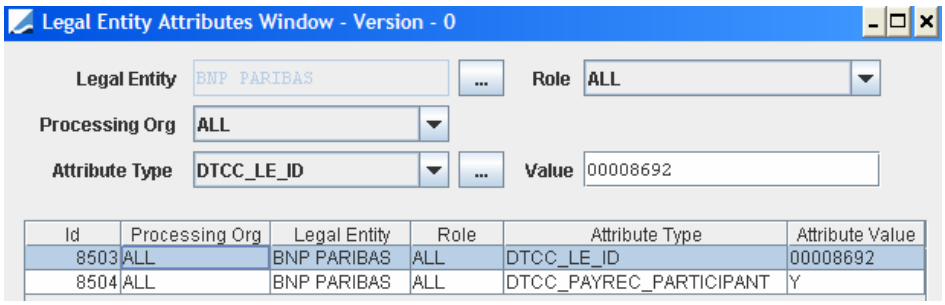
3.1.2 Processing Org and Counterparty Attributes

Choose Navigator > Configuration > Legal Data > Legal Entities and load a legal entity (processing org or counterparty). Then click **Attributes** to specify the attributes.

The **DTCC_LE_ID** has been created by the SQL scripts. It should be set to the DTCC member id for the processing organization, AND for the trade counterparty.

The DTCC_PAYREC_PARTICIPANT attribute should be added. For each Legal Entity, this attributes will be used to define if it uses DTCC form payments matching. Therefore you should set this attribute to "Y" for every Processing Organization and Counterparty for which transfers will be sent to DTCC.

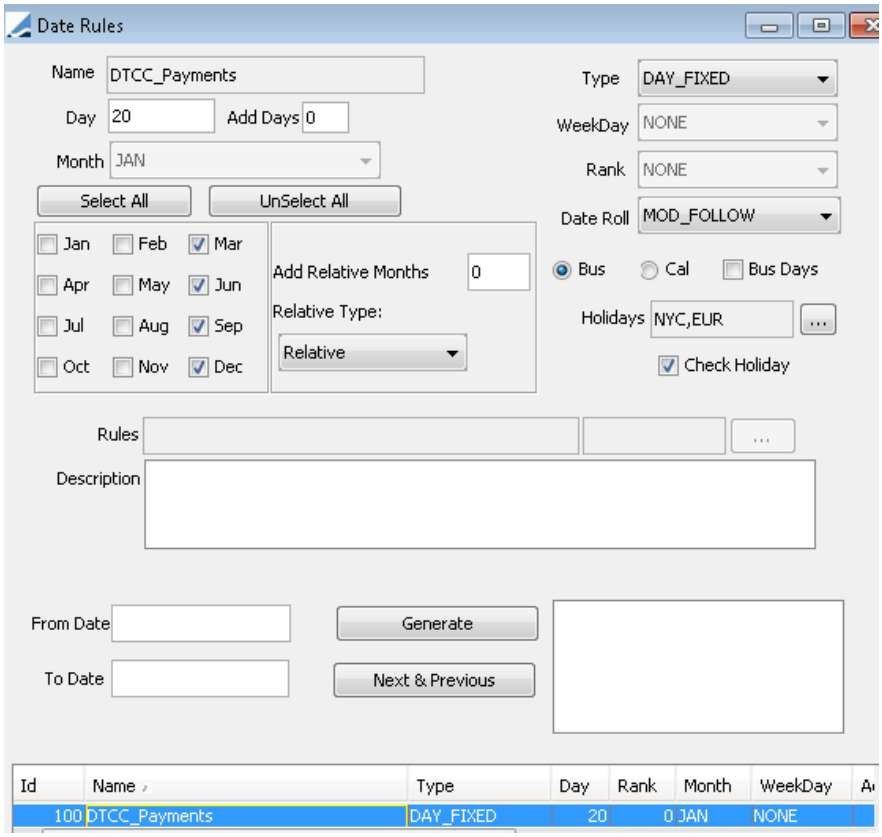
Sample Processing Org Attributes



3.1.3 Date Rules

The following date rule should be created. It will filter all payments with settlement date as of the 20th of each quarter. It will be used in the static data filter definitions.

Choose Navigator > Configuration > Definitions > Date Schedule Definitions > Date Rule to define a date rule.



3.1.4 Static Data Filters

Before being netted and settled, transfers sent to DTCC will follow a dedicated Workflow, apart from the standard one. This particular workflow will be chosen by the Transfer Engine based on the static data filter defined below

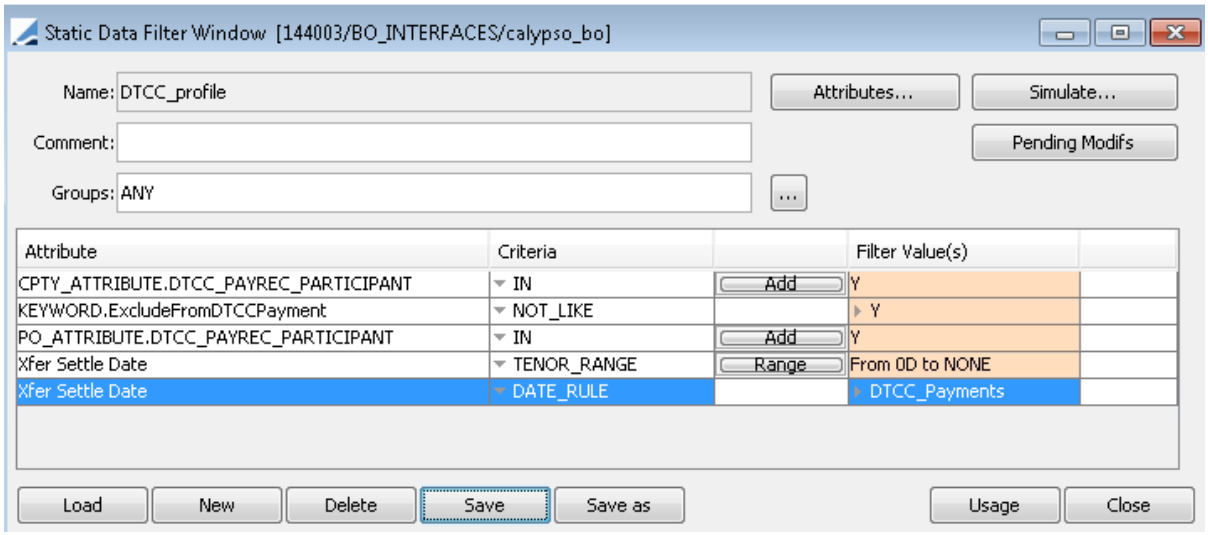
Choose Navigator > Configuration > Filters > Static Data Filter to define static data filters.

DTCC Profile

[NOTE: This static data filter does not need to be attached to anything. It must just exist. It must be named DTCC_profile. The name is case sensitive]

This static data filter will be used to identify transfers that have the XferProfileType = DTCC.

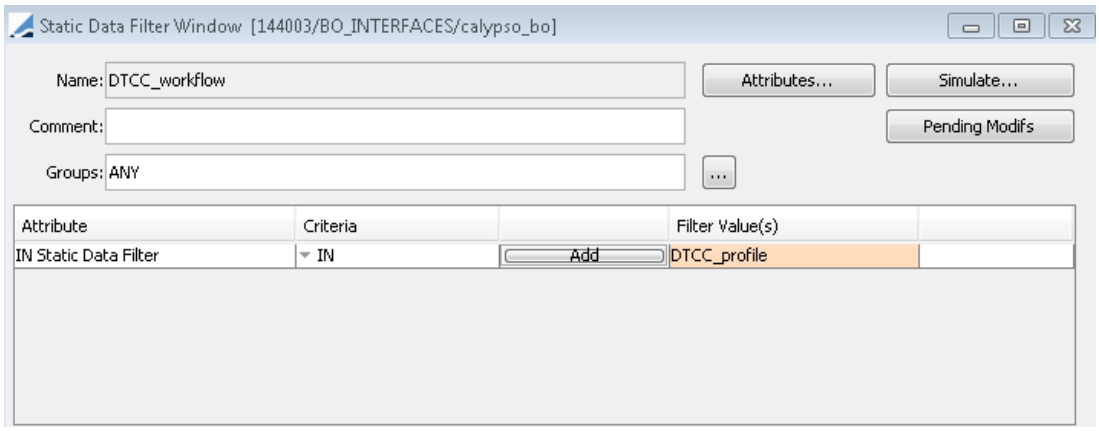
Sample DTCC_profile



DTCC Transfer Workflow

[NOTE: This static data filter does not need to be attached to anything. It must just exist. It must be named DTCC_workflow. The name is case sensitive]

This static data filter will upload all transfers related to DTCC so that they will follow the DTCC transfer workflow (XferWorkflowType = DTCC).



You may customize these filters to adjust to the exact list of payments you want to send to DTCC

3.1.5 Transfer Workflow

You need to setup a workflow for the subtype DTCC, and modify the standard workflow.

You can import the default Workflow DTCCXferWorkflow.wf provided by Calypso.

Choose Navigator > Configuration > Workflow > Workflow Config, or Workflow Graph Config to define workflows.

DTCC Workflow

This workflow will be used by transfers, provided the static data filter DTCC_workflow is specified in the system.

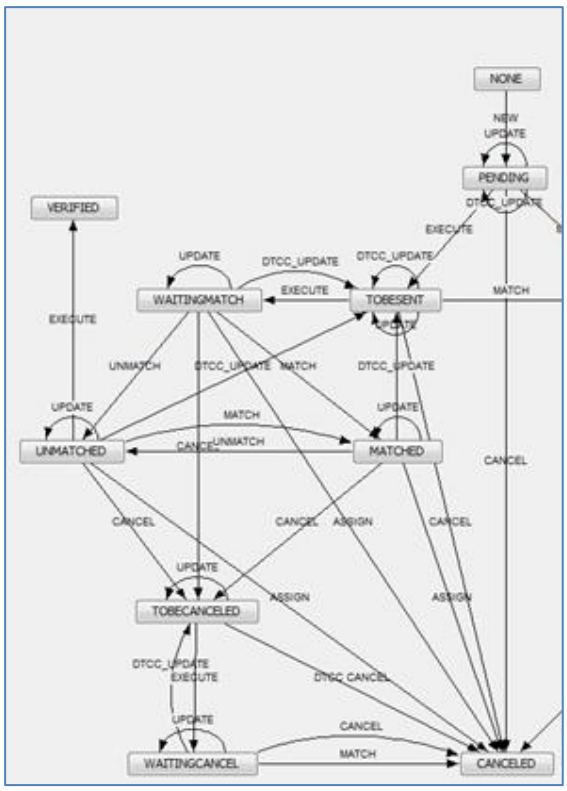
Orig Status	Action	Resulting Status	Use STP	Subtype	Rules	Kick Off/ Cut Off	Filter
MATCHED	ASSIGN	CANCELED	false	DTCC		false	
MATCHED	CANCEL	TOBECANCELED	false	DTCC		false	
MATCHED	DTCC_UPDATE	TOBESENT	false	DTCC		false	
MATCHED	NEW	PENDING	false	DTCC		false	
MATCHED	UPMATCH	UNMATCHED	false	DTCC		false	
MATCHED	UPDATE	MATCHED	false	DTCC		false	
NONE	NEW	PENDING	true	DTCC		false	
PENDING	CANCEL	CANCELED	false	DTCC		false	
PENDING	DTCC_UPDATE	PENDING	false	DTCC		false	
PENDING	EXECUTE	TOBESENT	true	DTCC	CheckKickOff	true	
PENDING	MATCH	DTCCMATCHED	false	DTCC		false	
PENDING	UPDATE	PENDING	false	DTCC		false	
TOBECANCELED	CANCEL	TOBECANCELED	false	DTCC		false	
TOBECANCELED	DTCCCANCEL	CANCELED	false	DTCC		false	
TOBECANCELED	EXECUTE	WAITINGCANCEL	false	DTCC		false	
TOBECANCELED	UPDATE	TOBECANCELED	false	DTCC		false	
TOBESENT	CANCEL	CANCELED	false	DTCC		false	
TOBESENT	DTCC_UPDATE	TOBESENT	false	DTCC		false	
TOBESENT	EXECUTE	WAITINGMATCH	false	DTCC		false	
TOBESENT	UPDATE	TOBESENT	false	DTCC		false	
UNMATCHED	ASSIGN	CANCELED	false	DTCC		false	
UNMATCHED	CANCEL	TOBECANCELED	false	DTCC		false	
UNMATCHED	DTCC_UPDATE	TOBESENT	false	DTCC		false	
UNMATCHED	EXECUTE	VERIFIED	false	DTCC	ResetWorkflowType	false	
UNMATCHED	MATCH	MATCHED	false	DTCC		false	
UNMATCHED	UPDATE	UNMATCHED	false	DTCC		false	

Orig Status	Action	Resulting Status	Use STP	Subtype	Rules	Kick Off/ Cut Off	Filter
WAITINGCANCEL	CANCEL	WAITINGCANCEL	false	DTCC		false	
WAITINGCANCEL	DTCC_UPDATE	TOBECANCELED	false	DTCC		false	
WAITINGCANCEL	MATCH	CANCELED	false	DTCC		false	
WAITINGCANCEL	UPDATE	WAITINGCANCEL	false	DTCC		false	
WAITINGMATCH	ASSIGN	CANCELED	false	DTCC		false	
WAITINGMATCH	CANCEL	TOBECANCELED	false	DTCC		false	
WAITINGMATCH	DTCC_UPDATE	TOBESENT	false	DTCC		false	
WAITINGMATCH	MATCH	MATCHED	false	DTCC		false	
WAITINGMATCH	UNMATCH	UNMATCHED	false	DTCC		false	
WAITINGMATCH	UPDATE	WAITINGMATCH	false	DTCC		false	

The STP transition PENDING – EXECUTE – TOBESENT is applied **with a kick-off**, see Kick-Off Configuration.

The **DTCC_UPDATE** action is hard-coded. It impacts the transfer matching. DTCC does not accept cancel/correct in case of changes in the settlement amount or in the currency. However, the standard logic of the Transfer engine is to cancel and create a new transfer in case of such changes. Therefore the Transfer engine logic has been modified in that case. This is why we have added the notion of transfer Profile.
See §3.2 for more details about how the transfer engine knows, in case of DTCC transfers, when a modification should lead to an update of existing transfers, or a cancellation+creation of a new transfer.

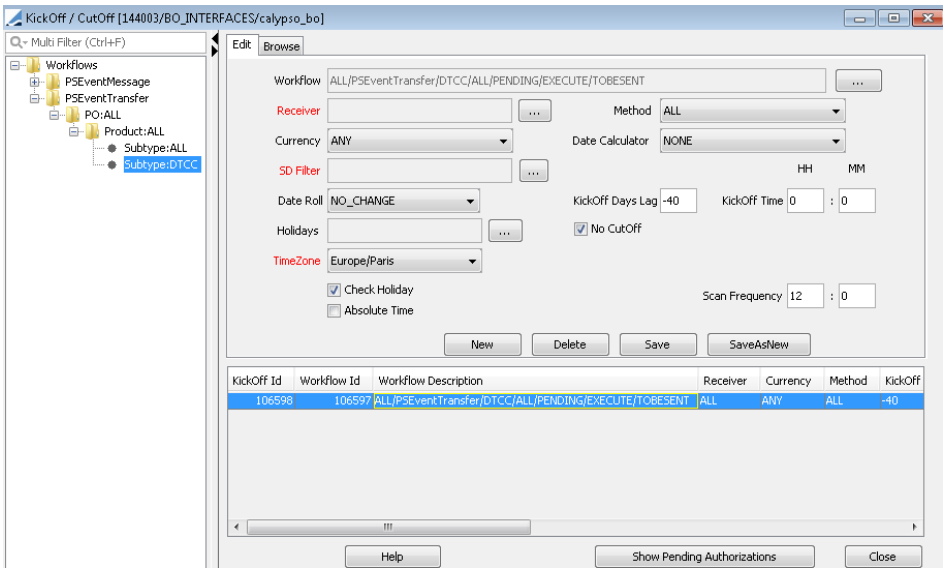
The ***ResetWorkflowType*** rule set in the UNMATCHED – EXECUTE – VERIFIED transition allows to settle transfers that, at the end of the DTCC payment matching process, are still unmatched.
This rule resets the workflow type of the transfer to "NONE" so that the transfer will follow the "standard" workflow for transfers. Note: this rule can only be set on an action where the resulting status exists in the standard workflow, otherwise transition will fail.



3.1.6 Kick-Off Configuration

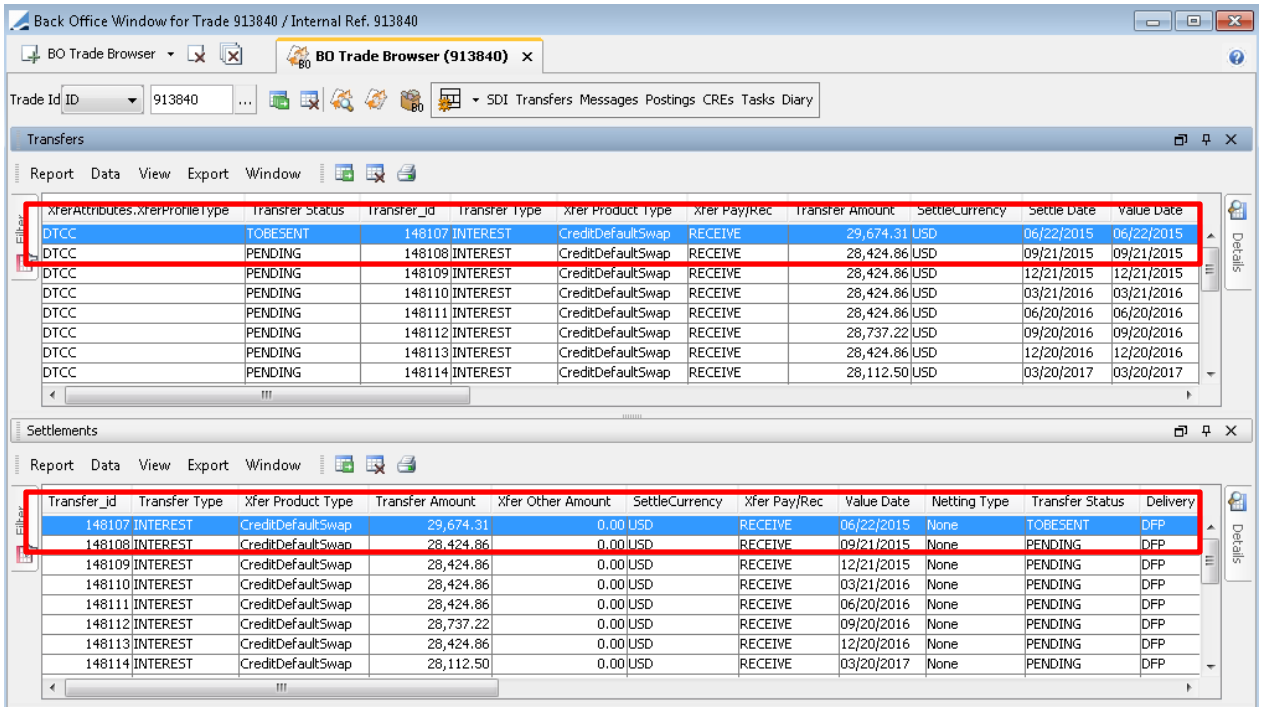
This kick-off configuration indicates how many days, before the next settlement date payment, the system can start sending payment declarations to DTCC.

Choose Navigator > Configuration > Workflow > Kick-off Cut-off Config to define a kick-off configuration.



Usually, you will set the day lag to a little more than one month in order to start sending messages to DTCC a reasonable number of days before the settlement date, but less than a quarter in order to send only messages related to the next payment date and not the following ones.

For example, on the 1st of August, you would expect the 20/09 transfer being "TOBESENT" while the followings remaining "PENDING":



3.2 Processing DTCC Payments

The DTCC payment reconciliation process works in the following manner: you send a file to DTCC containing payments to be made using the scheduled task DTCC_PAYMENT_EXPORT, then you import a file from DTCC that contains the status of the payments, using the scheduled task DTCC_PROCESS_PAYMENTS. The Calypso payments are updated accordingly, and you can then net matched payments.

In DTCC, you cancel very rarely a payment. You update it. In order to change the Transfer engine logic of matching, we have added the notion of transfer profile. This allows to define a specific matching and also to set the DTCC attributes on the transfers. Those will then be used in the files sent to DTCC.

The **action CANCEL** is applied only if:

- The DTCC participant Id is changing
- The Profile Type is changing
- The Value Date is changing

Depending on the situation, the filters might decide if a payment needs status TOBECANCELED, this means that a file will be generated containing a row with the keyword CAN.

The **action DTCC_UPDATE** is applied if:

- The settlement amount is changing
- The settle currency is changing
- The trade currency is changing
- The trade direction is changing (from PAY to RECEIVE and vice versa)
- The maturity date
- The trade notional

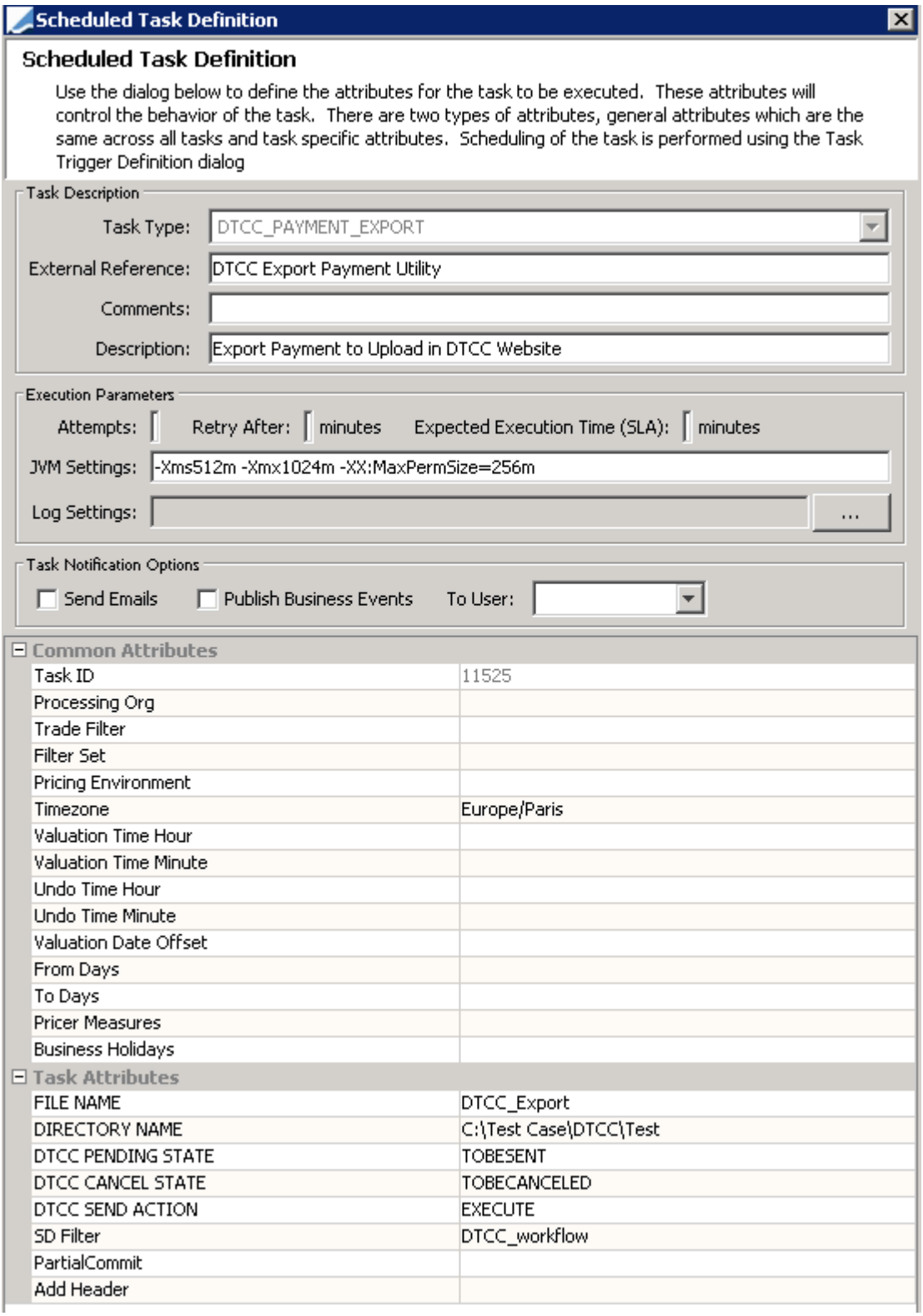
- The reset rate

In these cases the original trade must be resent to the DTCC containing the changes but the keyword will be NEW. This means that upon reception of the file containing the update, the DTCC will replace the former payment with the new one.

3.2.1 Exporting Payments to DTCC

In order to export payment to DTCC, run the scheduled task DTCC_PAYMENT_EXPORT. It will generate a file that contains the payments for the next payment iteration.

Choose Navigator > Configuration > Scheduled Tasks > Scheduled Tasks to configure and run the scheduled task DTCC_PAYMENT_EXPORT.



Specify the following attributes:

FILE NAME — Name of the `.csv` file.

DIRECTORY NAME — Name of the directory to store the `.csv` file. The file will then be sent manually to DTCC.

DTCC PENDING STATE — Enter the transfer status from which you want to send the payment to DTCC. It corresponds to the NEW (and therefore also the updated) types of payments.

DTCC CANCEL STATE — Enter the transfer status from which you want to send the CANCEL payments to DTCC.

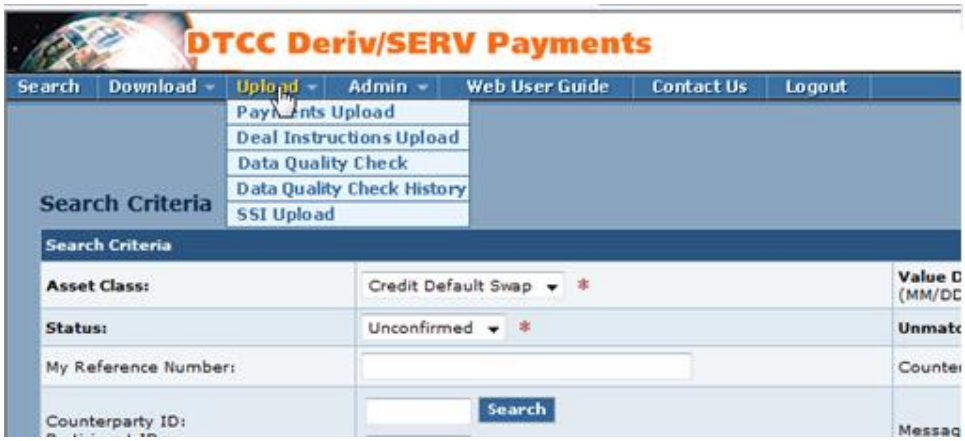
DTCC SEND ACTION — Enter the action that should be applied to the transfers in status DTCC PENDING STATE and DTCC CANCEL STATE.

The scheduled task will log exceptions of type EX_DTCC_PAYMENT_EXPORT if the execution is not successful.

This csv file generated has to be uploaded on DTCC site:

- Go to DTCC payment reconciliation application.

- Then upload > Payment Upload

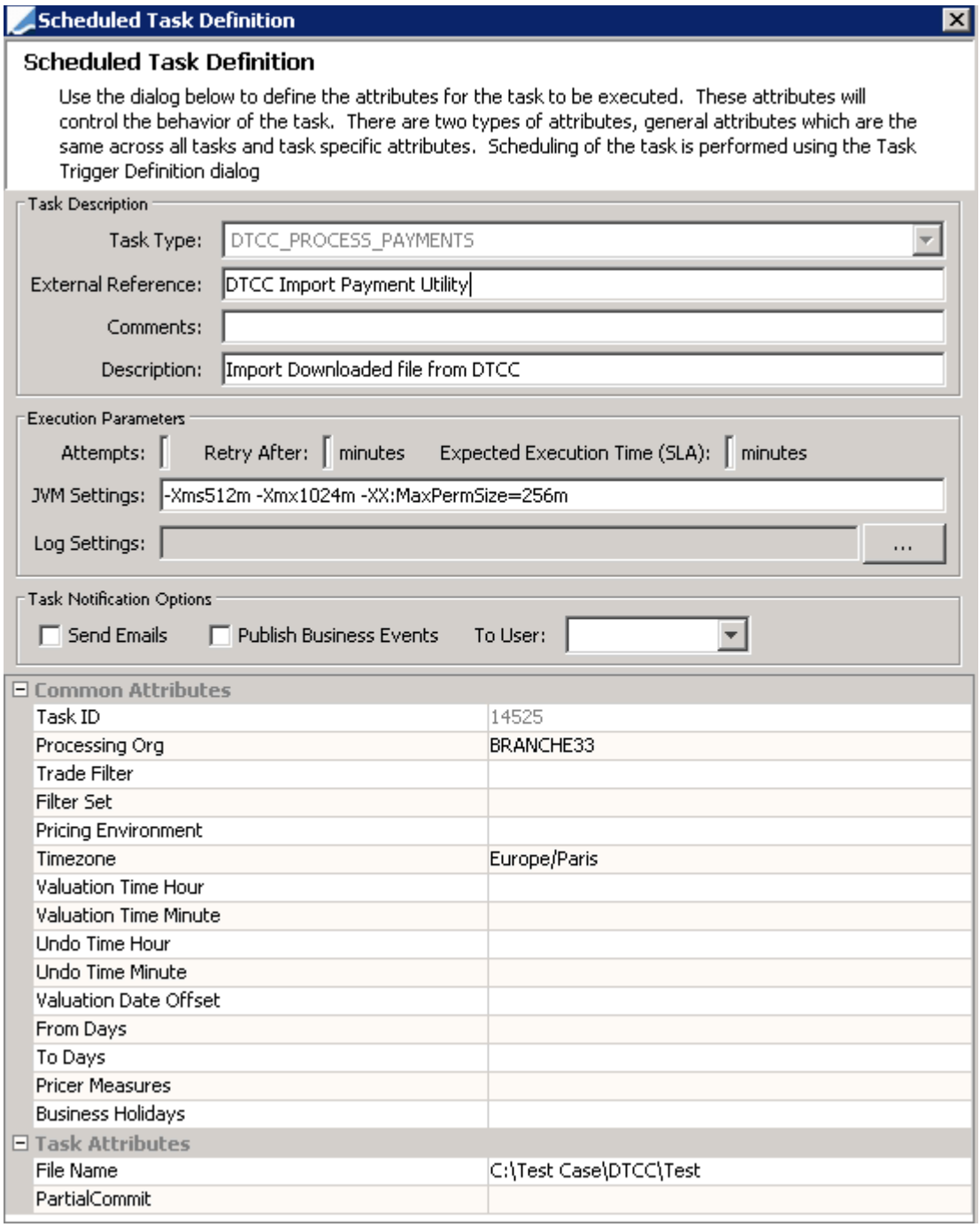


3.2.2 Importing Payments from DTCC

You can import a file from DTCC that will contain the status of each payment previously sent (MATCHED, UNMATCHED, etc.) using the scheduled task DTCC_PROCESS_PAYMENTS.

On DTCC, in the Download menu, select Full Data Extract Report.

Choose Navigator > Configuration > Scheduled Tasks> Scheduled Tasks to configure and run the scheduled task DTCC_PROCESS_PAYMENTS.



Specify the following attributes:

FILE NAME — Name of the `.csv` file with the source directory.

Partial Commit — If you are importing a large amount of payments, you can choose to import the payments in bulk (in order to avoid performance problems). For example, if you set the value to 5000, the scheduled task will import 5000 payments at a time.

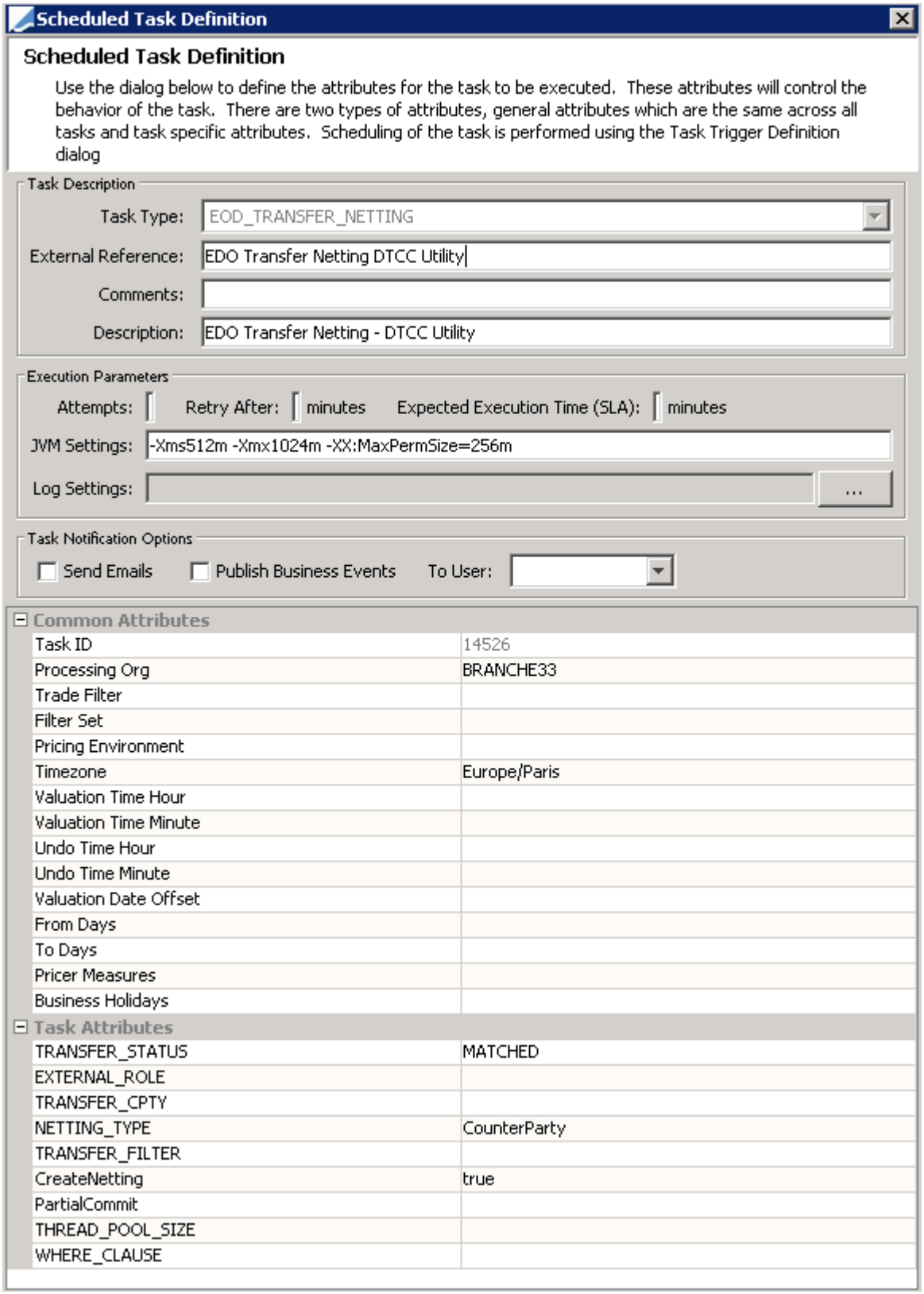
The scheduled task will log exceptions of type EX_DTCC_PROCESS_PAYMENTS if the execution is not successful.

3.2.3 Netting Matched Transfers

Once payments have been matched by DTCC, they can be netted using the scheduled task EOD_TRANSFER_NETTING.

Choose Navigator > Configuration > Scheduled Tasks > Scheduled Tasks to configure and run the scheduled task EOD_TRANSFER_NETTING.

Please refer to "Automatic Netting in Batch" paragraph in the TransferGeneration user guide for a full documentation of the EOD_TRANSFER_NETTING scheduled task.



Recommended attributes are:

TRANSFER_STATUS: MATCHED (this is the status of transfers eligible to netting).

NETTING_TYPE — Counterparty.

CreateNetting — "true" (this means that during the netting process, existing underlying transfers will be updated, whereas "false" means that existing transfers would be cancelled and replaced by new ones).

Then save the configuration. You can start the scheduled task on the fly if you have checked the Execute checkbox. Otherwise it can be executed by the Scheduling engine.

3.2.4 Settlement of Unmatched Transfers

§3.2.3 shows how to net DTCC MATCHED transfer. But all DTCC transfers that are not matched must also be paid, outside of a netting process.

At this step, all DTCC transfers that have not been canceled or netted should be UNMATCHED. If some transfers are still in some other states (PENDING, TOBESENT, WAITINGMATCH), *process them manually to UNMATCHED*.

Then, apply the EXECUTE action to them: they are now VERIFIED and, due to the ***ResetWorkflowType*** rule, are now processed in the "standard" workflow and can be settled one by one.

Section 4. DTCC Central Settlement

In partnership with CLS Bank International, DTCC provides central, automated settlement of payments for contracts processed through the Warehouse's Central Settlement service.

DTCC Central settlement enhances Payment reconciliation process by taking in account that:

- By default all transfers can be matched against a TIW cashflow and settled in CLS, whereas only quarterly roll coupons used to be matched during the payment reconciliation process
- Cashflows correctly calculated by DTCC in the TIW don't require their transfer to go through the payment reconciliation process
- Netted transfers settled by DTCC in CLS on behalf of the Processing Org don't need a "real" settlement message in Calypso.

Assumption: SDI have to be configured in order to select by default CLS SDI if transfers follow this behavior. SDI will not be updated automatically by transfer update.

To support DTCC Central settlement services, you need to:

1- Ensure that you have created the schema data for data category DTCC Central settlement

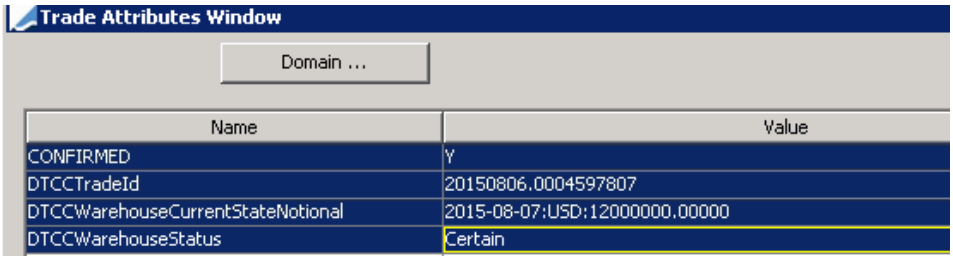
2- Import these following message workflows:

- IncomingTIWPayment
- IncomingTIWSettlement

Process overview

- Payment messages are real-time messages generated and transmitted when a trade has been confirmed and is certain in the Warehouse.

When trade is confirmed, trade attributes are updated with following information:



In Calypso, Payment messages are imported as BOMessage, compared against corresponding BOTransfer, and update Transfer status accordingly.

- Settlement messages are point in time (batch-driven) generated starting at settlement minus five days for trades that are certain in the Warehouse

In Calypso, Settlement message are imported as BOMessage and used to reflect the current settlement status in the CLS system.

4.1 IncomingTIWPayment Message Workflow

IncomingTIWPayment Messages are recognized based on the <TradeMsg> and "Payment" <TransType>

```
<TradeMsg>
    <Activity>New</Activity>
    <Status>Auto</Status>
    <TransType>Payment</TransType>
    <ProductType>CreditDefaultSwapShort</ProductType>
    <YourTradeId>
```

Orig Status	Action	Resulting Status	Use STP	Subtype	Rules	Filter
NONE	NEW	PENDING	false	IncomingTIWPayment		
PENDING	INDEX	INDEXED	false	IncomingTIWPayment	DTCCUpdateTransfer	
PENDING	ManualIndex	TOBESENT	false	IncomingTIWPayment	DTCCUpdateTransfer	

Incoming Messages are:

- Indexed with the corresponding transfer if all criteria matched (trade ref number, value date, ccy, amount, payer, receiver)
- Indexation updates transfer status and update transfer attributes: DTCCNetPaymentId, DTCCPmtMsgStatus, DTCCCashflowId
- Manually matched otherwise

Attributes

Name	
Asset Class	Credit
DTCCParticipantId	00004T35
DTCCUserId	00004T33
DayCount	ACT/360
EndDate	09/21/2015
LegId	1
MatchingSystem	DTCC
Notional	12,000,000
RefEntity	Ford Motor Company
RefTradeId	921303
ResetRate	1.00000000
ScheduledMatDate	09/20/2020
SettleDate	08/07/2015
StartDate	08/07/2015
TradeCcy	USD
TradeDate	08/06/2015
TradeNotional	12,000,000
XferProfileType	DTCC
XferWorkflowType	DTCC
DTCCNetPaymentId	04T3304T35U15264
DTCCPmtMsgStatus	Auto
DTCCCashflowId	000000000032383401

4.2 IncomingTIWSettlement Message Workflow

IncomingTIWPayment Messages are recognized based on the <TradeMsg> and "Settlement" <TransType>
This message is imported automatically

```
<TradeMsg>
  <Activity>New</Activity>
  <Status>Locked</Status>
  <TransType>Settlement</TransType>
  <ProductType>CentralSettlement</ProductType>
  <YourTradeId>
    <partyTradeIdentifier>
      <fpml:partyReference href="DTCC00004T33" />
      <fpml:tradeId tradeIdScheme="DTCCNetPaymentId">04T3304T34U13079</fpml:tradeId>
    </partyTradeIdentifier>
  </YourTradeId>
</TradeMsg>
```

Orig Status	Action	Resulting Status	Use STP	Subtype	Rules	Filter
NONE	NEW	PENDING	false	IncomingTIWSettlement		
PENDING	CANCEL	CANCELED	false	IncomingTIWSettlement		
PENDING	IGNORE	IGNORED	false	IncomingTIWSettlement		
PENDING	INDEX	INDEXED	false	IncomingTIWSettlement		
PENDING	SPLIT	SPLIT	false	IncomingTIWSettlement	DTCCSplitGross	

The DTCCSplitGross rule splits a DTCC Gross Settlement message into separate messages (create "fake" IncomingTIWSettlement – recognized by SplitGross RecortType

Individual messages are then indexed with the corresponding transfer (indexed with DTCCCashflowId)

Transfer is updated to reflect CLS settlement status.

Index action process only "Net" messages. Keys for indexation are TradeRefNbr, DTCCNetPaymentId and currency.

4.3 Transfer Workflow

DTCC transfer workflow need to be enhanced with these following status/ action/ Filter:

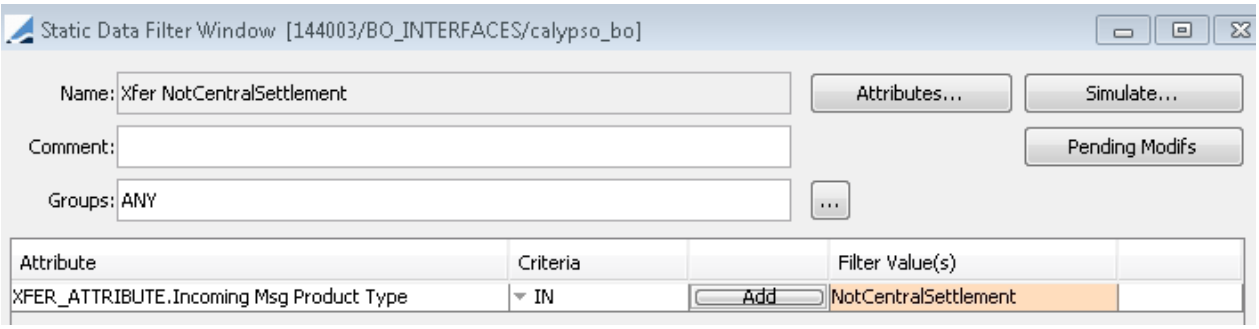
Orig Status	Action	Resulting Status	Use STP	Subtype	Rules	Kick Off/ Cut Off	Filter
DTCC CANCELED	CANCEL	CANCELED	false	DTCC			
DTCC MATCHED	CANCEL	TOBECANCELED	false	DTCC			
DTCC MATCHED	DTCC CANCEL	DTCC CANCELED	false	DTCC			
DTCC MATCHED	Locked Final	Without CLS info	false	DTCC			
DTCC MATCHED	Locked IPIS	IPIS	false	DTCC			
DTCC MATCHED	MATCH	DTCC MATCHED	false	DTCC			
DTCC MATCHED	NotCentralSettlt	ExcludedFromCLS	false	DTCC			Xfer NotCentralSettlement

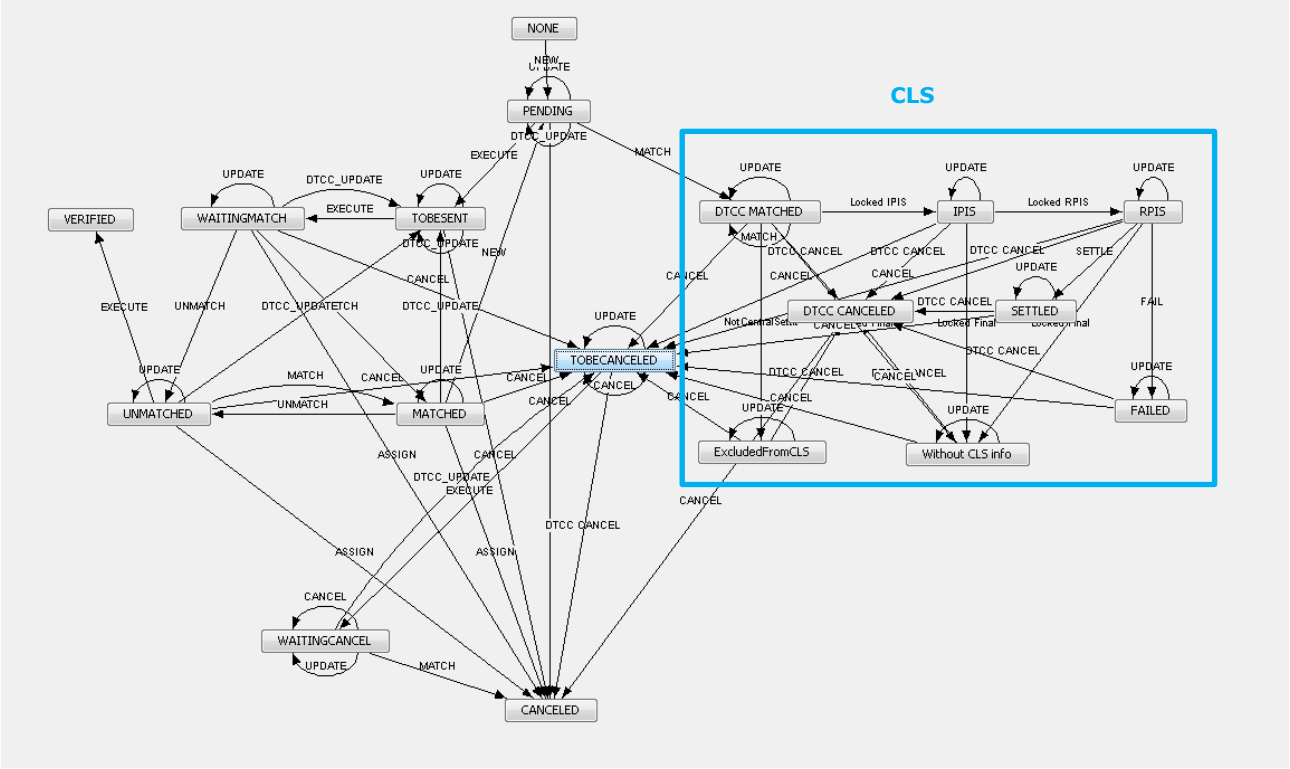
Orig Status	Action	Resulting Status	Use STP	Subtype	Rules	Kick Off/ Cut Off	Filter
DTCC MATCHED	UPDATE	DTCC MATCHED	false	DTCC			
ExcludedFromCLS	CANCEL	TOBECANCELED	false	DTCC			
ExcludedFromCLS	DTCC CANCEL	DTCC CANCELED	false	DTCC			
ExcludedFromCLS	UPDATE	ExcludedFromCLS	false	DTCC			
FAILED	CANCEL	TOBECANCELED	false	DTCC			
FAILED	DTCC CANCEL	DTCC CANCELED	false	DTCC			
FAILED	UPDATE	FAILED	false	DTCC			
IPIS	CANCEL	TOBECANCELED	false	DTCC			
IPIS	DTCC CANCEL	DTCC CANCELED	false	DTCC			
IPIS	Locked Final	Without CLS info	false	DTCC			
IPIS	Locked RPIS	RPIS	false	DTCC			
IPIS	UPDATE	IPIS	false	DTCC			
RPIS	CANCEL	TOBECANCELED	false	DTCC			
RPIS	DTCC CANCEL	DTCC CANCELED	false	DTCC			
RPIS	FAIL	FAILED	false	DTCC			
RPIS	Locked Final	Without CLS info	false	DTCC			
RPIS	SETTLE	SETTLED	false	DTCC			
RPIS	UPDATE	RPIS	false	DTCC			
SETTLED	CANCEL	TOBECANCELED	false	DTCC			
SETTLED	DTCC CANCEL	DTCC CANCELED	false	DTCC			
SETTLED	UPDATE	SETTLED	false	DTCC			
Without CLS info	CANCEL	TOBECANCELED	false	DTCC			
Without CLS info	DTCC CANCEL	DTCC CANCELED	false	DTCC			
Without CLS info	UPDATE	Without CLS info	false	DTCC			

RPIS : Revised Pay In Scheduled

IPIS: Initial Pay In Scheduled

Xfer NotCentralSettlement SD filter:





Section 5. DTCC Confirmation Matching

5.1 Additional Settings

Run DTCCSchemaData.xml. See section 2.1 DTCC Data

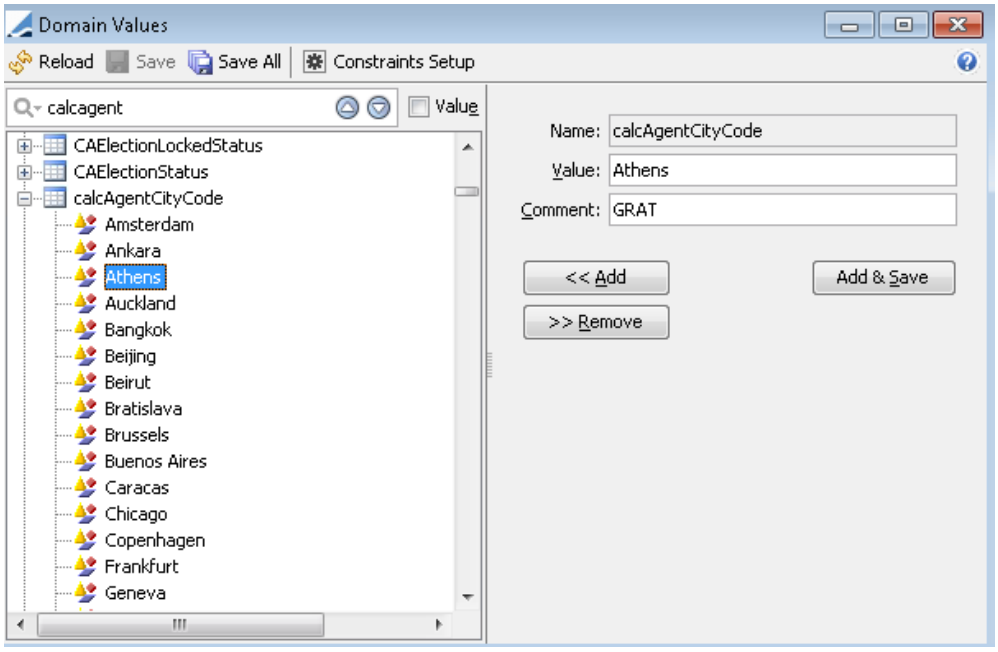
Then import the workflow contained in the `DTCCMsgWorkflows.wf` file under the folder "resources".

5.1.1 Domain Values

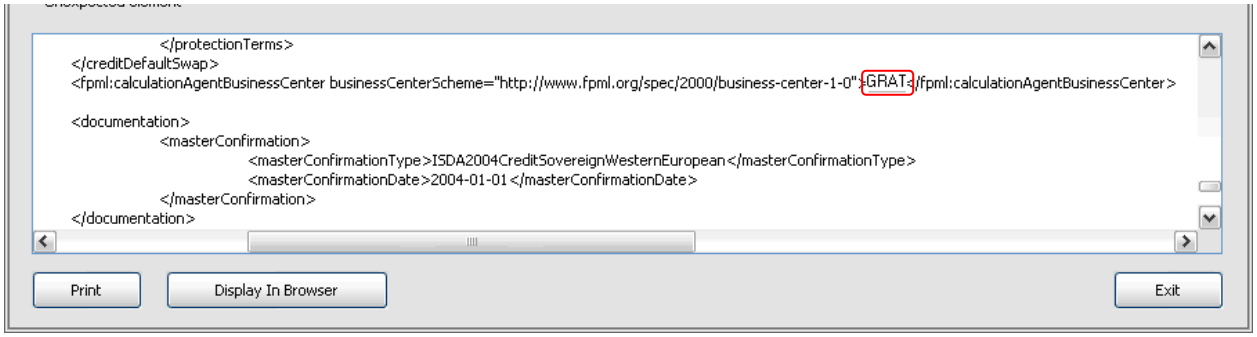
Choose Navigator > Configuration > System > Domain Values to define domain values.

Domain calcAgentCityCode – This domain is used at trade level to choose the calculation agent city of a CreditDefaultSwap, which correspond in DTCC messages to the filed calculationAgentBusinessCenter.

For example: the city of Athens will use the code "GRAT" in DTCC messages



Sample DTCC message:



Domain messageType – values:

- DTCC_CONFIRM
- DKCONF
- INCOMINGCONF

Domain DTCC.Templates – values:

- DTCC.selector – template selector for DTCC
- All CreditSwap_* templates apply to product type CreditDefaultSwap when using master confirmation.
- All CreditSwap_Matrix_* templates apply to product type CreditDefaultSwap when using physical settlement matrix
- All CreditIndex_* templates apply to product type CDSIndex/CDSABSIndex when using master confirmation.
- All CreditIndex_StandardTerms_* templates apply to product type CDSIndex/CDSABSIndex when using standard terms
- All CreditIndexTranche_* templates apply to product type CDSIndexTranche
- All InterestSwap_* templates apply to product type Swap
- All CreditSwaption_* templates apply to product type CreditDefaultSwaption
- All CreditIndexSwaption_* templates apply to product type CDSIndexOption
- All EquityShareSwap_* templates apply to product type EquityLinkedSwap
- WorkflowUpdate_New_A_To_DTCC.xml template apply to the Workflow Update transaction type.

Message keywords for DTCC templates are described in details under DTCC Message Keywords.

Domain DTCCWorkflowUpdateFields – used to list the trade audit values related to a WorkflowUpdate transaction in DTCC. Values:

- ADDKEY#DTCC_PO_SuperId
- MODKEY#DTCC_PO_SuperId
- DELKEY#DTCC_PO_SuperId
- ADDKEY#DTCC_PO_DeskId
- MODKEY#DTCC_PO_DeskId
- DELKEY#DTCC_PO_DeskId
- ADDKEY#DTCC_CPTY_SuperId
- MODKEY#DTCC_CPTY_SuperId
- DELKEY#DTCC_CPTY_SuperId
- ADDKEY#DTCC_CPTY_DeskId
- MODKEY#DTCC_CPTY_DeskId
- DELKEY#DTCC_CPTY_DeskId

Domain allegedIndexingMsgStatus: To indicate the Internal Message Types to be used for an Alleged Trade Processing – values

- PENDING
- SENT
- VERIFIED
- UNCONFIRMED

Domain allegedIndexingMsgType – value DTCC_CONFIRM: To indicate the Internal Message Types to be used for an Alleged Trade Processing

Domain allegedAcceptanceTolerance – value 100: To Specify the tolerance limit (%) for accepting/indexing an alleged external message e.g, 100,80

Domain externalMessageTransactionType : To Define a Matching Config – values

- Trade
- Termination
- Increase
- Exit
- Assignment

5.1.2 Issuer Attributes

[NOTE: in order to import issuer information directly from RED/Markit, please follow the step described in the MarkIt Integration User Guide]

The issuer attributes are used to determine the issuer's name, region, and sector.

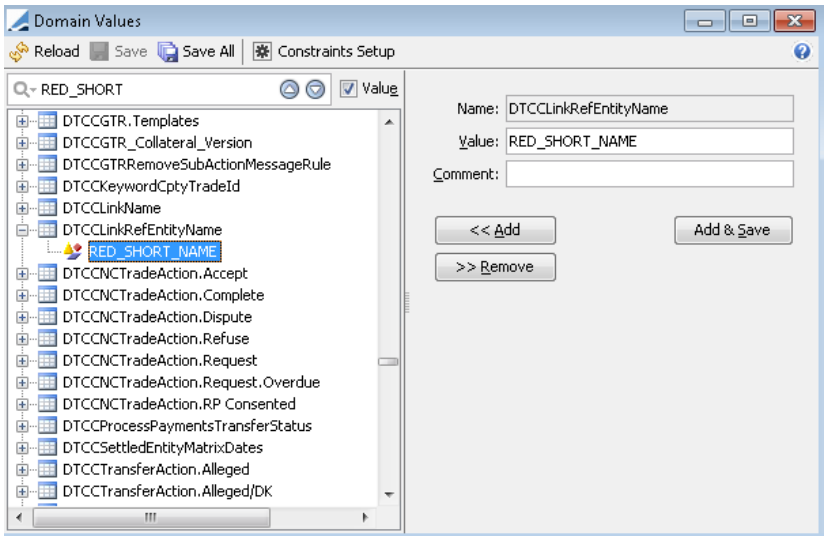
Choose Navigator > Configuration > Legal Data > Legal Entities and load an issuer. Then click **Attributes** to specify the attributes.

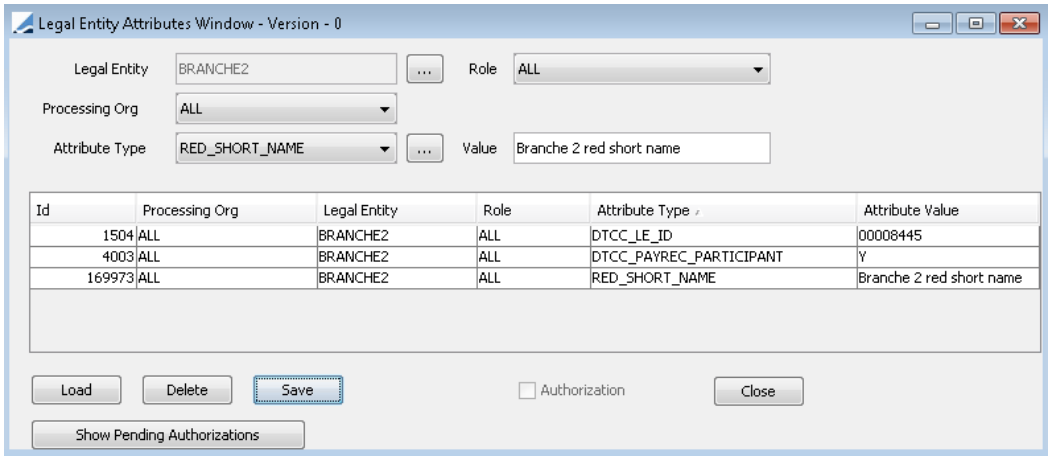
Issuer Name

By default, Reference Entity name in DTCC messages corresponds to issuer's full name.

You can customize this behavior and use any LE attribute you want. For this, create a domain named "**DTCCLinkRefEntityName**" and add to it the name of the attribute you want to be used.

For example:



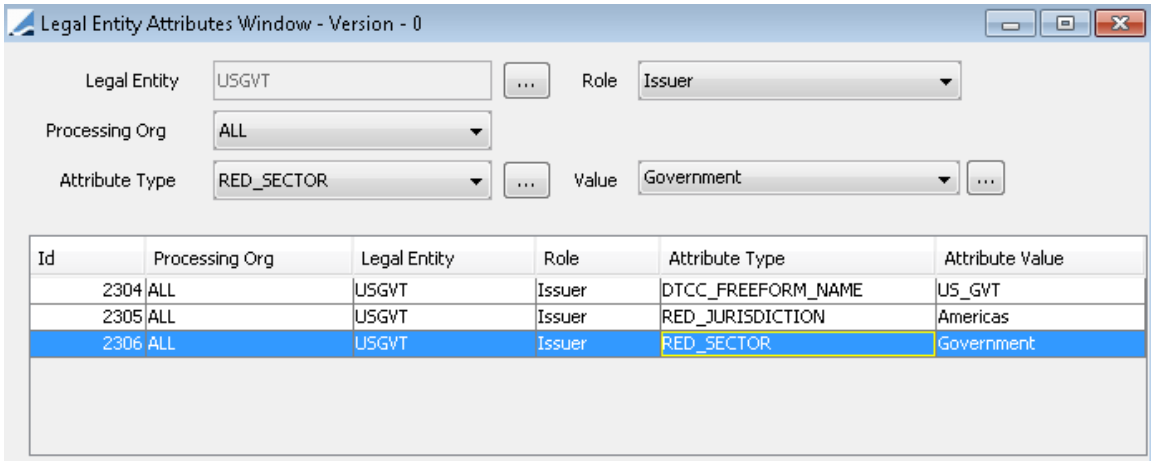


Region and Sector

The following attributes are also required for the issuer: **RED_REGION** and **RED_SECTOR**.

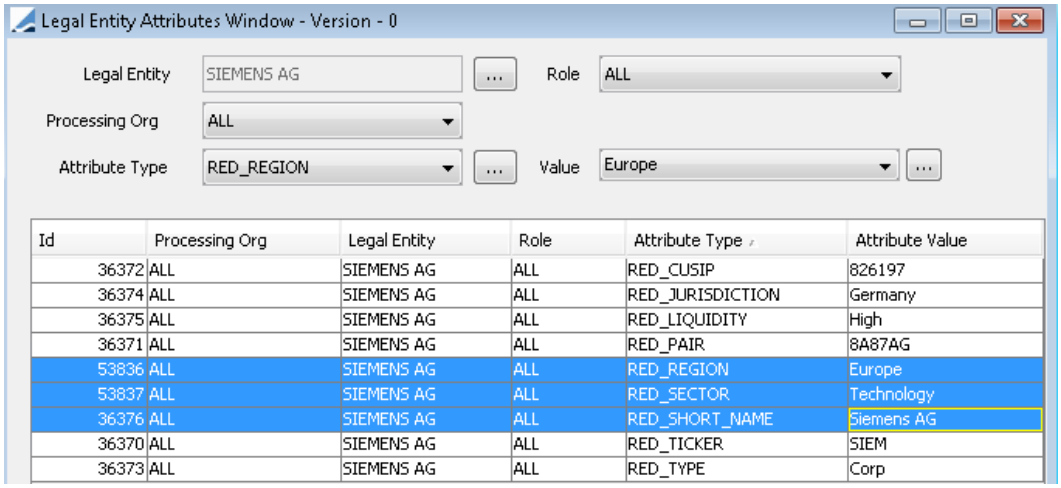
The available values for the RED_REGION attribute must be specified in the issuerRegion domain AND in the "leAttributeType.RED_REGION" domain (AND they must be the same). The available values for the RED_SECTOR attribute must be specified in the issuerSector domain AND in the "leAttributeType.RED_SECTOR" domain. This double-entry requirement will soon be removed, but is currently required.

Sample Issuer Attributes



Id	Processing Org	Legal Entity	Role	Attribute Type	Attribute Value
2304	ALL	USGVT	Issuer	DTCC_FREEFORM_NAME	US_GVT
2305	ALL	USGVT	Issuer	RED_JURISDICTION	Americas
2306	ALL	USGVT	Issuer	RED_SECTOR	Government

In the example above, the issuer name will be provided by the attribute DTCC_FREEFORM_NAME



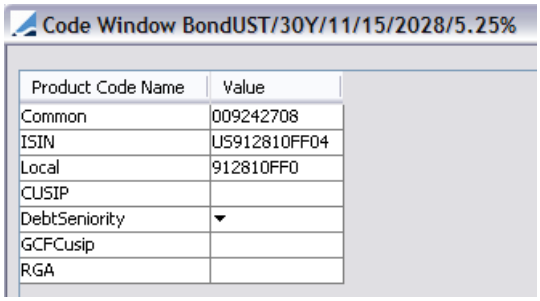
Id	Processing Org	Legal Entity	Role	Attribute Type	Attribute Value
36372	ALL	SIEMENS AG	ALL	RED_CUSIP	826197
36374	ALL	SIEMENS AG	ALL	RED_JURISDICTION	Germany
36375	ALL	SIEMENS AG	ALL	RED_LIQUIDITY	High
36371	ALL	SIEMENS AG	ALL	RED_PAIR	8A87AG
53836	ALL	SIEMENS AG	ALL	RED_REGION	Europe
53837	ALL	SIEMENS AG	ALL	RED_SECTOR	Technology
36376	ALL	SIEMENS AG	ALL	RED_SHORT_NAME	Siemens AG
36370	ALL	SIEMENS AG	ALL	RED_TICKER	SIEM
36373	ALL	SIEMENS AG	ALL	RED_TYPE	Corp

5.1.3 Reference Obligation Attributes

The reference obligation of the issuer should have the following attributes:

- If the attribute **RED_PAIR** is set on the issuer, then the reference obligation should also have the RED_PAIR attribute specified, or the ISIN number.

- If the attribute RED_PAIR is not set on the issuer, then the reference obligation should have the ISIN number.



The reference obligation is set on the issuer by clicking the **Ref Ob** button. To specify attributes for the reference obligation, choose Navigator > Configuration > Fixed Income > Bond Product, and load the reference obligation. Then click **Codes** to set the attributes.

5.1.4 Processing Org and Counterparty Attributes

Choose Navigator > Configuration > Legal Data > Legal Entities and load a legal entity (processing org or counterparty). Then click **Attributes** to specify the attributes.

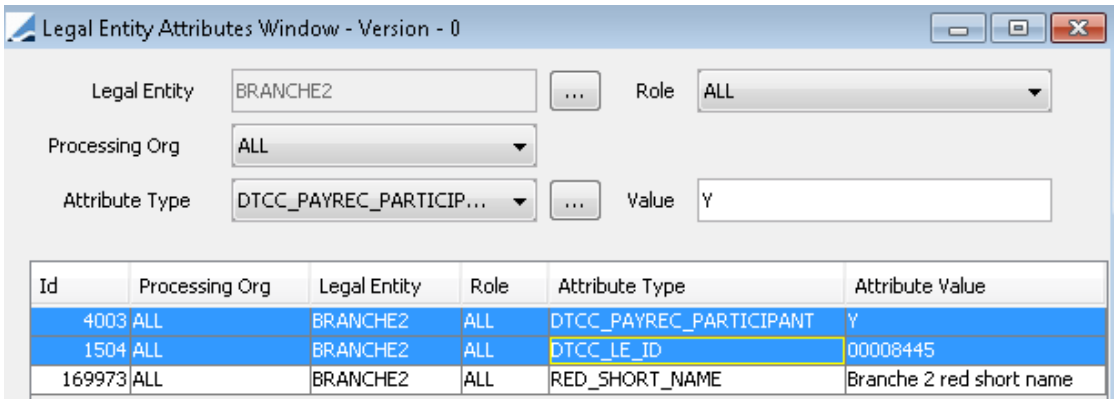
The **DTCC_LE_ID** has been created by the SQL scripts. It should be set to the DTCC member id for the processing organization, AND for the trade counterparty.

The **DTCC_PAYREC_PARTICIPANT** attribute should be added. The value should be set to Y for both the processing organization AND the trade counterparty.

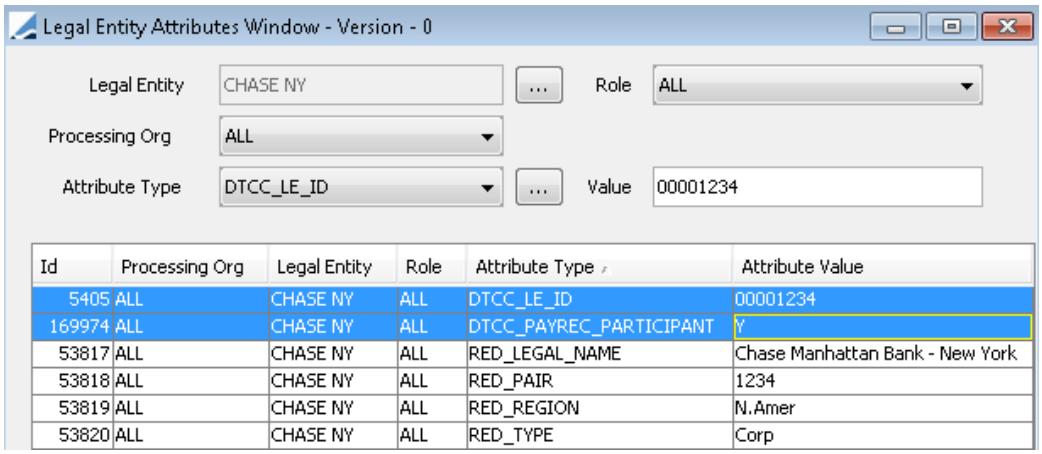
[**NOTE**: A single processing organization or counterparty may have more than one DTCC ID. Therefore Calypso has provided the following logic to provide a simple setup whenever it's possible that can be customized for a particular message contact or trade when needed:

- when trade keyword DTCC_PO_ID (respectively DTCC_LE_ID) is present, it is used as the DTCC PO (respectively counterparty) ID
- otherwise, when DTCC code is present on sender (respectively receiver) contact, it is used as the DTCC PO (respectively counterparty) ID
- otherwise LE attribute DTCC_LE_ID taken as the DTCC PO / counterparty ID]

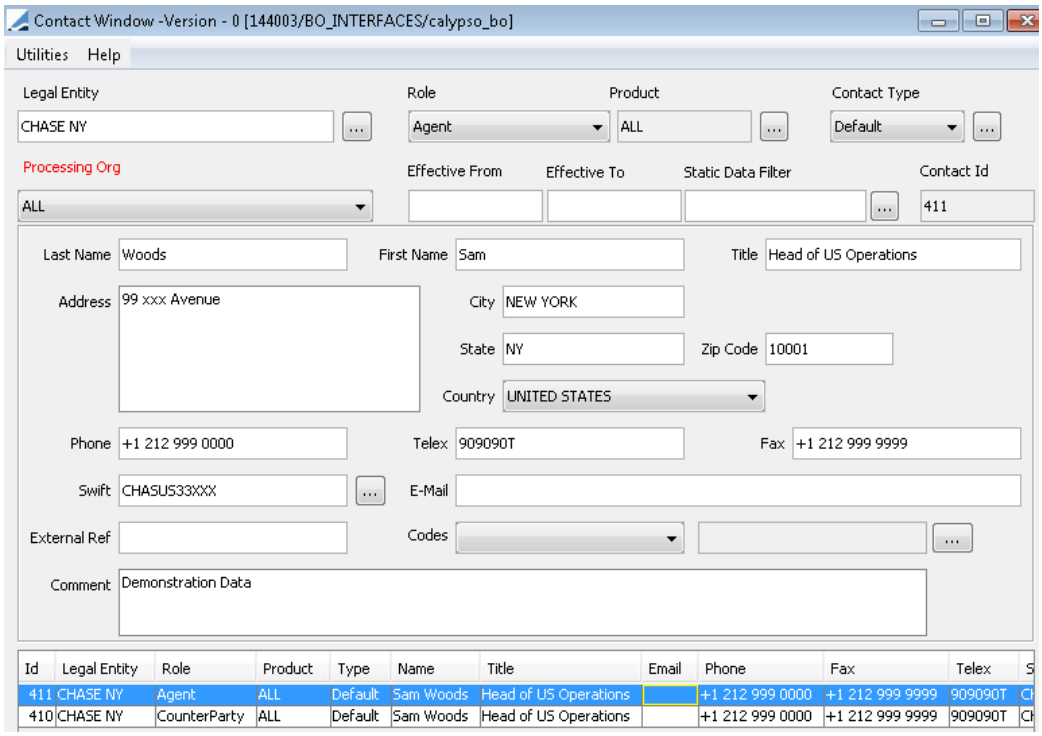
Sample Processing Org Attributes



Sample Counterparty Attributes



Sample Contact Code

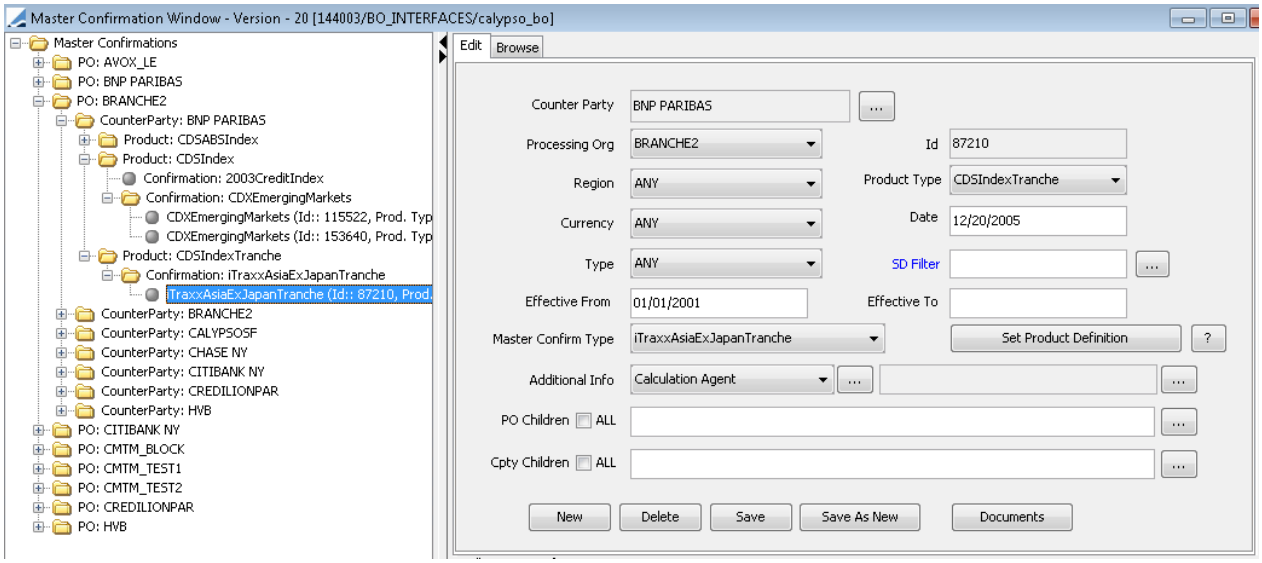


5.1.5 Master Confirmations

You need to create a master confirmation (i.e. a master agreement) between the counterparty and the processing organization. When generating DTCC confirmations, the system will retrieve the master confirmation based on the counterparty, the processing organization, the issuer's region, the issuer's sector, the currency, the product type, and the holidays.

Domain values **masterConfirmationType**, **masterconfirmationTypeMatrix**, **masterConfirmationTypeAPS** have been created by the SQL scripts.

Choose Navigator > Configuration > Messages & Matching > Master Confirmations.



Select a counterparty and a processing org, then enter the fields as applicable.

Region (Optional) — Regions (Markit RED Regions) are specified like in the CDSSettlementMatrix window:

- Either the list comes from the domain specified in the templateLinkRegion
- Or it comes from the RED_REGION LE attribute domain

Type (Optional) — Types (Markit RED Types) are specified like in the CDSSettlementMatrix window:

- Either the list comes from the domain specified in the templateLinkType
- Or it comes from the RED_TYPE LE attribute domain

Note: prior to Calypso V11.0, master confirmations were defined by RED Region and Sector. In order to comply with market standard (mainly settlement matrix), it's now defined by RED Region and Type.

Region and Type can both be either specific or both ANY.

Date (Mandatory) — Enter the effective date of the master confirmation.

Master Confirm Type (Mandatory) — Confirmation types are specified in the masterConfirmationType domain. Types related to Physical Settlement Matrix / Standard Terms must also be specified in masterConfirmationTypeMatrix domain (except for those related to index tranche, for which only standard terms exist).

Click **Set Product Definition** to save the details of the master confirmation to a product template.

The trade window will appear based on the selected product type.

Enter product details as applicable and close the window. The details will be saved as a product template.

This is optional. You can ignore the message indicating that a product definition is not set.

Click the second ... button to set values for the additional information.

Out-of-the-box, the following information is available:

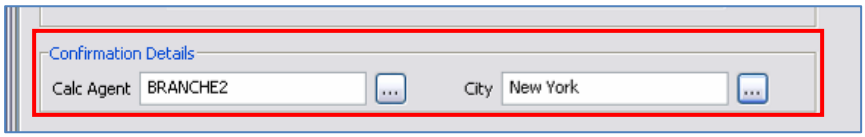
Calculation Agent (Optional) — Used for populating data in the confirmations. Note that the calculation agent entered here must be a legal entity of type Calc_Agent. It should be the same as the calculation agent set on the trade, if specified.

If you want a master confirmation to be valid with every calculation agent, then set its value to "ANY".

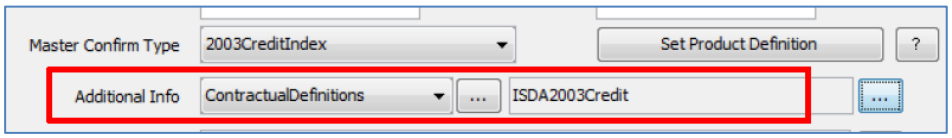
Calculation Agent City (Optional)— Used for populating data in the confirmations of type Corporate Asian or Sovereign or Matrix for CDS, and CDSIndex / tranches Standard Terms for example. Note that the city entered

here must be the same as the city set on the trade, if specified. Cities are defined in the calcAgentCityCode domain.

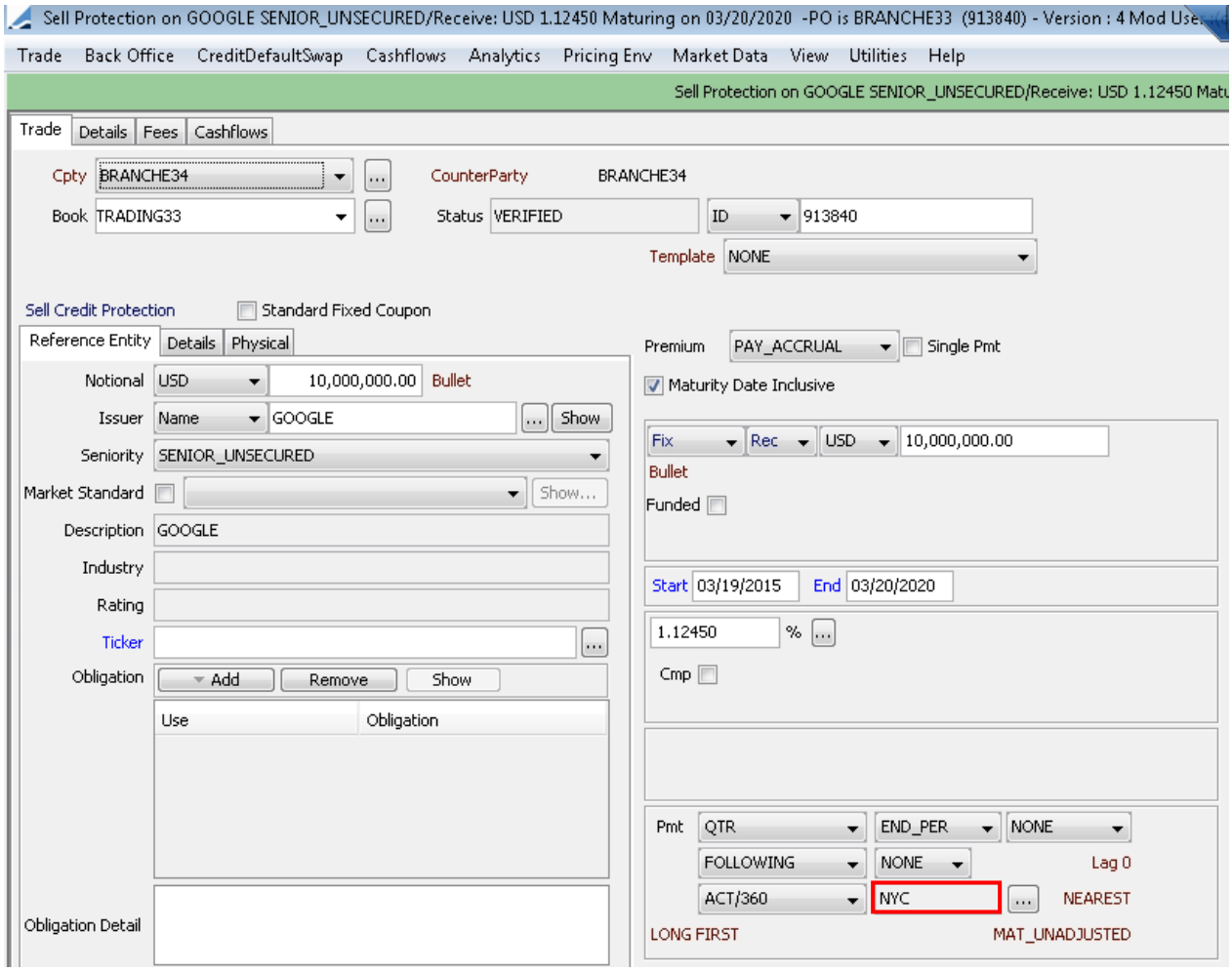
If you want a master confirmation to be valid with every calculation agent city, then set its value to "ANY".



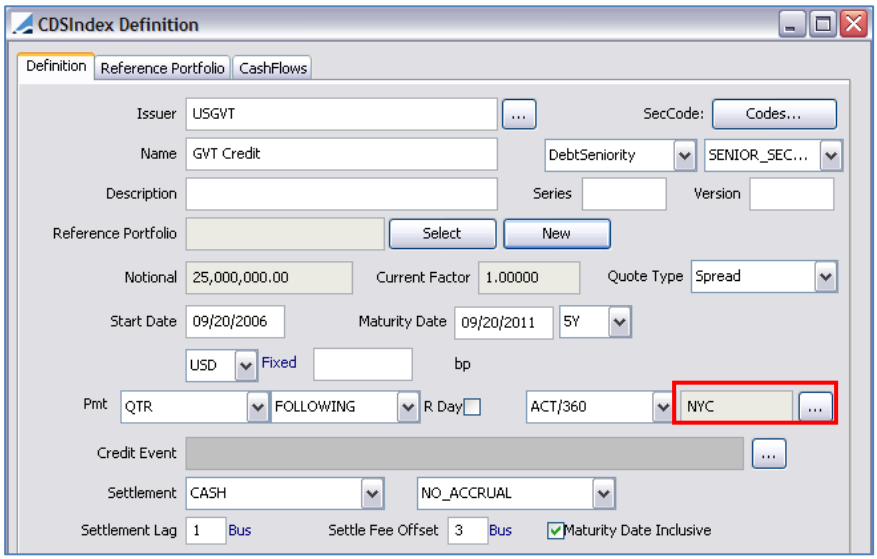
ContractualDefinitions (Mandatory) — Used for populating data in the confirmations. It was introduced by DTCC for 2014 ISDA Credit Definitions. Possible values are: ISDA2003Credit or ISDA2014Credit



Holidays (Mandatory) — the system will control that the holiday calendars in the master confirmation match with the payment holiday calendars on the trade. If you want a master confirmation to be valid with every holidays, then set its value to "ANY".



For CDS Index, matching will be done against the holiday calendars of the index.



isDTCC (Mandatory for DTCC confirmations) — Must be set to Y for generating DTCC confirmations.

You can click the first ... button to add user-defined information.

Then click **Save**.

Note that if the Authorization mode is enabled, another user will have to authorize your entry.

5.2 Message set-up and workflow

5.2.1 Business logic

Message setup and workflow described below are designed to follow this business logic:

➡ Every time a confirm must be generated:

Calypso tries to send it through DTCC using Matrix/Standard Terms form of documentation

If not possible, it tries to send it through DTCC using master confirmation

Otherwise, it is sent by FAX (HTML confirm)

➡ Once a trade reaches VERIFIED status

If trade has already been confirmed in DTCC (keyword CONFIRMED set to "Y")

- If the *only* changes to the trade are
  - Not declared into the *TradeFieldsNotAmendment* domain (see usage of in the messaging documentation)
  - **And** are part of the DTCC workflow update fields (fields listed in the *DTCCWorkflowUpdateFields* domain). Note that **only** the addition / modification / deletion of keyword *DTCC_PO_SuperId* and *DTCC_PO_DeskId* are supported: Please refer to Domain DTCCWorkflowUpdateFields mentioned in page 28

  Then a ***WorkflowUpdate*** message is created

- Else, it means some contractual trade details have been changed and an "Amendment" confirmation is created

An "**Assignment EE**" confirm is produced if the PO is the transferee of a novated trade (new trade where the NovationTransferor keyword is given).

An "**Amendment**" confirmation if trade has already been confirmed in DTCC (keyword CONFIRMED set to "Y")

A "**Trade**" confirmation otherwise

Note : if trade is no more eligible to DTCC confirm process

- If trade has already been confirmed, an Exit message is produced
- Otherwise a Trade message with activity "Cancel" is produced
- In any case, a fax confirm is created

➡ Once a trade reaches TERMINATED status

Depending on termination type and termination reason, transaction type will be: **Termination**, **Assignment** or **Increase**

Depending on message SubAction, activity will be "New" or "Cancel". Note: Termination messages with a "Cancel" SubAction are created when a termination is rejected. If this setup is required in your environment, refer to the Termination Window documentation for the corresponding setup using the "Reject" trade workflow rule.

➡ Once a trade reaches CANCEL status

If trade has already been confirmed, and "**Exit**" message is produced

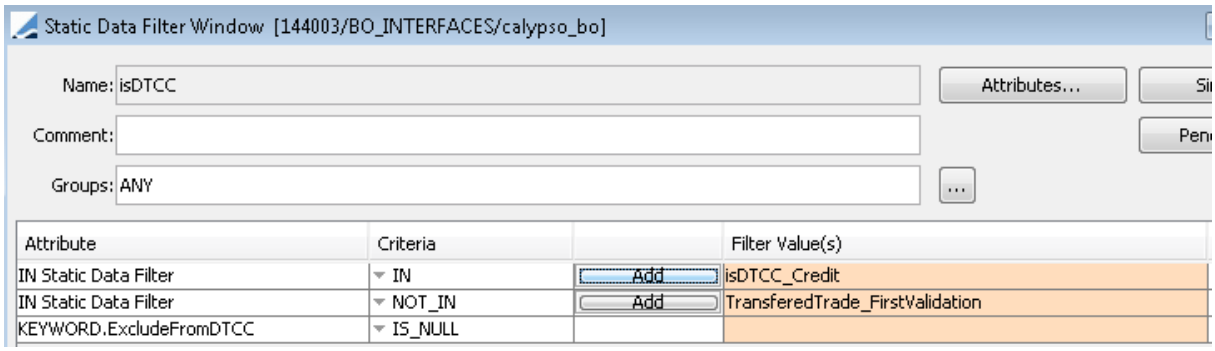
Otherwise a Trade message with activity "Cancel" is produced

➡ Once a trade reaches EXERCISE status

An "**Exercise**" confirm is produced

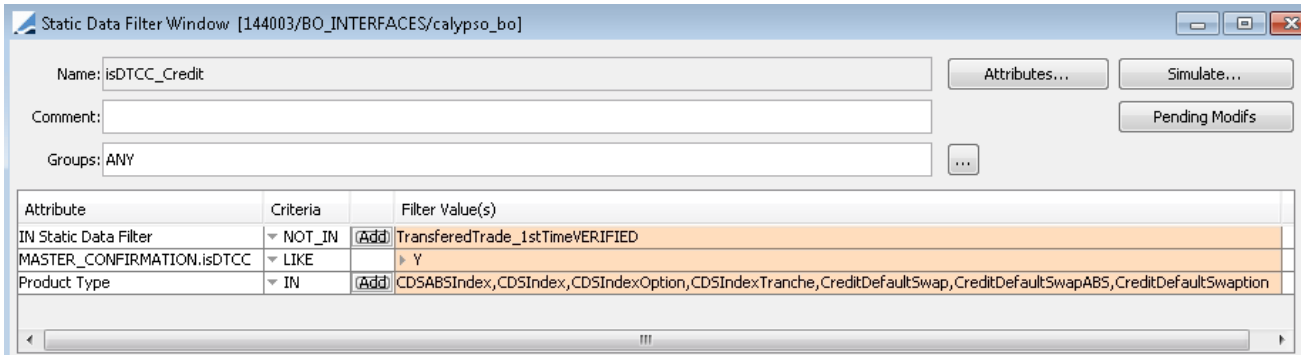
5.2.2 Static Data Filters

isDTCC



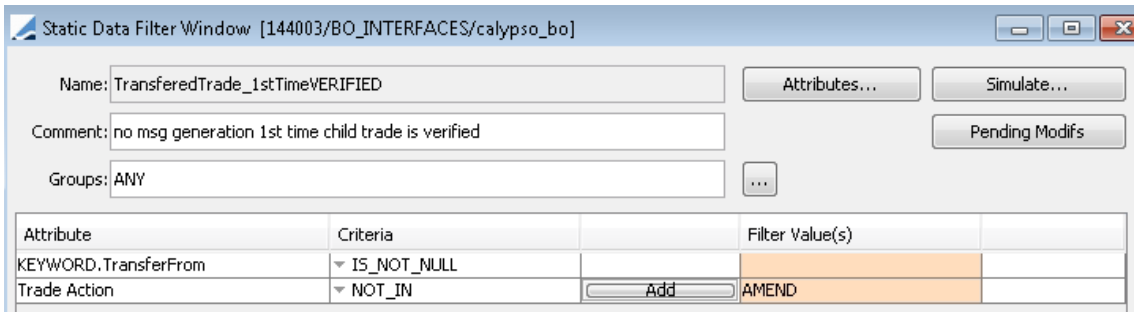
This filter has to be customized according to your exact trades, products and static data setup.

Here the default setup is that for a Swap, any trade can be DTCC eligible whereas for a CreditDefaultSwap, CDSIndex, CDSIndexTranche, CDSABSIndex, CreditDefaultSwaption, CDSIndexOption, and an EquityLinkSwap there must trade must have a Master Confirmation with the *isDTCC* attribute set to Y.



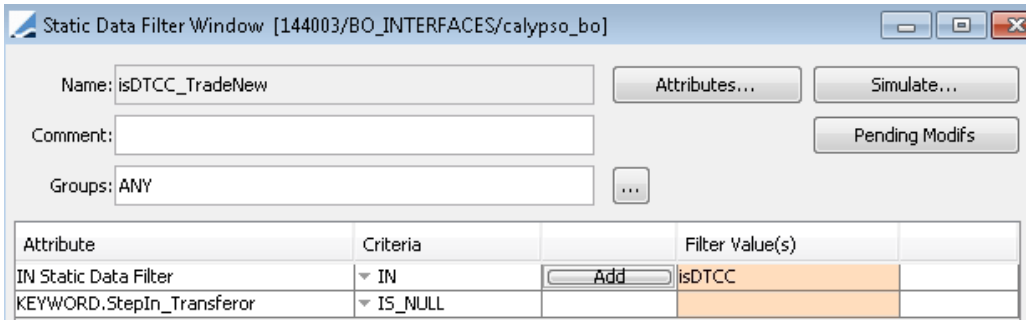
Special case: Second condition, "NOT_IN SD Filter TransferredTrade_1stTimeVERIFIED" has been added to prevent from having two confirmations generated at the same time each time a termination creates a transferred trade: in this case we produce a confirmation triggered by TERMINATED_TRADE event, so we have to prevent message engine from creating a confirmation on VERIFIED_TRADE (event generated by the transferred trade). This explains the first condition "TransferFrom IS_NOT_NULL".

But then, a transferred trade can be amended, and that time an amendment confirmation has to be generated: this explains the need of the "Trade Action NOT_IN AMEND" condition.

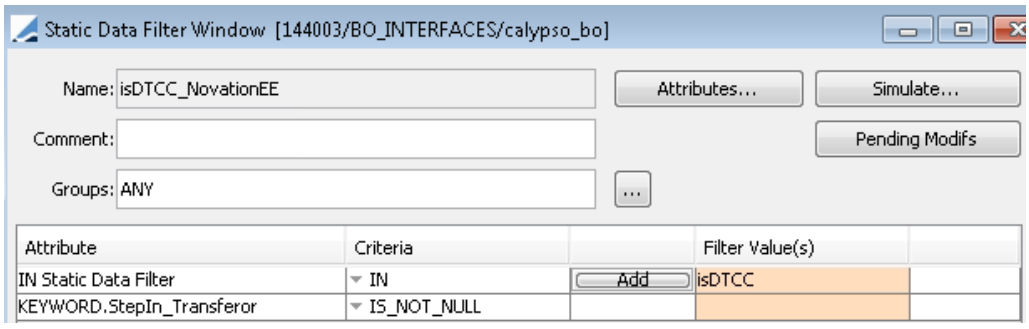


Note: you will certainly need to modify this SD filter according to your actual Trade Workflow setup in order for this filter to match exactly the definition of: "trade is not a transferred trade and trade reaches the VERIFIED status for the first time" (i.e. trade is not coming back to VERIFIED after being amended).

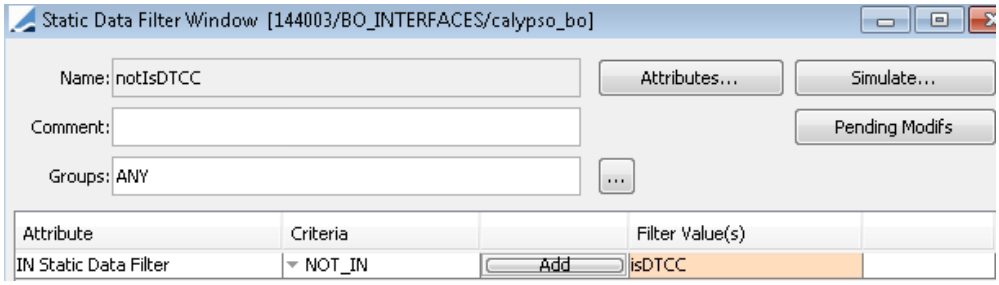
isDTCC_TradeNew



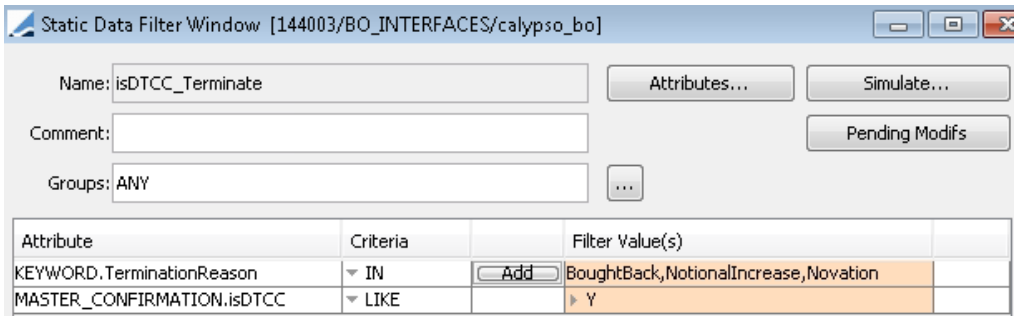
isDTCC_NovationEE



notIsDTCC



isDTCC_Terminate

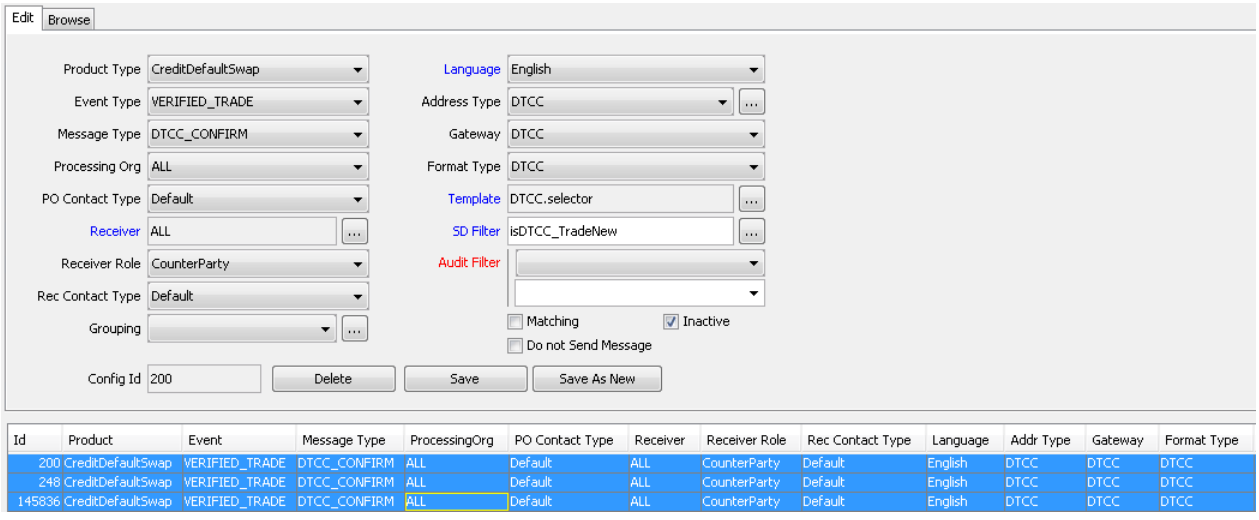


5.2.3 Message Setup

Message configurations are used by the message engine to generate the relevant confirmation in response to a trade event.

Choose Navigator > Navigator > Messages & Matching > Message Set-up to define the message configurations for the different types of confirmations to be sent to DTCC.

Note: html template and setup below is only a sample to illustrate the way to switch from DTCC confirms to FAX. Real FAX setup should be more complex (more templates, filter more detailed …), here it is just being added in order to have some message created when trade is not DTCC eligible.

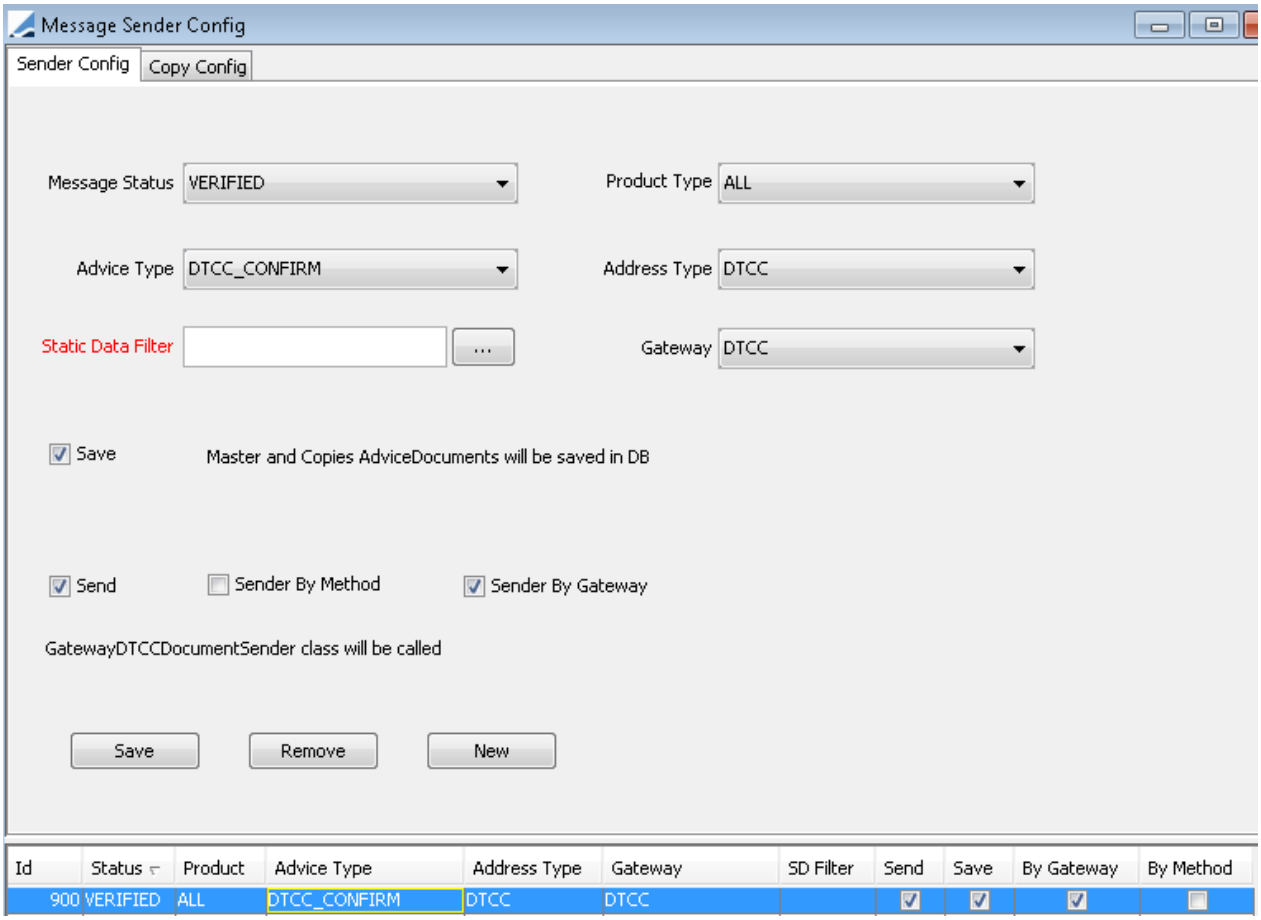


Product	Event	Message Type	Template Name	Static Data Filter
CreditDefaultSwap				
	CANCELED_TRADE	CONFIRM	cdsConfirm.html	notIsDTCC
	CANCELED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_TradeNew
	CANCELED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_NovationEE
	TERMINATED_TRADE	DTCC_CONFIRM	TerminationDTCC.selector	isDTCC_Terminate
	TERMINATED_TRADE	CONFIRM	cdsConfirm.html	notIsDTCC
	VERIFIED_TRADE	CONFIRM	cdsConfirm.html	notIsDTCC
	VERIFIED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_TradeNew
	VERIFIED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_NovationEE
CDSIndexTranche				
	CANCELED_TRADE	CONFIRM	cdsIndexTrancheConfirm.html	notIsDTCC
	CANCELED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_TradeNew
	CANCELED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC
	TERMINATED_TRADE	DTCC_CONFIRM	TerminationDTCC.selector	isDTCC_Terminate
	TERMINATED_TRADE	CONFIRM	cdsConfirm.html	notIsDTCC
	VERIFIED_TRADE	CONFIRM	cdsIndexTrancheConfirm.html	notIsDTCC
	VERIFIED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_TradeNew
	VERIFIED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_NovationEE
CDSIndex / CDSABSIndex				
	CANCELED_TRADE	CONFIRM	cdsIndexConfirm.html	notIsDTCC
	CANCELED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_TradeNew
	CANCELED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_NovationEE
	TERMINATED_TRADE	DTCC_CONFIRM	TerminationDTCC.selector	isDTCC_Terminate
	TERMINATED_TRADE	CONFIRM	cdsConfirm.html	notIsDTCC
	VERIFIED_TRADE	CONFIRM	cdsIndexConfirm.html	notIsDTCC
	VERIFIED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_TradeNew
	VERIFIED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_NovationEE
CreditDefaultSwaption				
	CANCELED_TRADE	CONFIRM	Confirm.html	notIsDTCC
	CANCELED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_TradeNew
	CANCELED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC
	TERMINATED_TRADE	DTCC_CONFIRM	TerminationDTCC.selector	isDTCC_Terminate
	TERMINATED_TRADE	CONFIRM	Confirm.html	notIsDTCC
	VERIFIED_TRADE	CONFIRM	Confirm.html	notIsDTCC
	VERIFIED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_TradeNew
	VERIFIED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_NovationEE

Product	Event	Message Type	Template Name	Static Data Filter
	EXERCISED_TRADE	CONFIRM	Confirm.html	notIsDTCC
	EXERCISED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC
CDSIndexOption				
	CANCELED_TRADE	CONFIRM	Confirm.html	notIsDTCC
	CANCELED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_TradeNew
	CANCELED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC
	TERMINATED_TRADE	DTCC_CONFIRM	TerminationDTCC.selector	isDTCC_Terminate
	TERMINATED_TRADE	CONFIRM	Confirm.html	notIsDTCC
	VERIFIED_TRADE	CONFIRM	Confirm.html	notIsDTCC
	VERIFIED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_TradeNew
	VERIFIED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC_NovationEE
	EXERCISED_TRADE	CONFIRM	Confirm.html	notIsDTCC
	EXERCISED_TRADE	DTCC_CONFIRM	DTCC.selector	isDTCC

5.2.4 Message Sender Configurations

Message sender configurations are used by the sender engine to actually send the DTCC confirmations through the DTCC gateway, once they are in SENT status. Choose Navigator >Configuration > Messages & Matching > Message Sender Configuration to define message sender configurations.



5.2.5 Message Workflow

Choose Navigator > Configuration > Workflow > Workflow Config, or Workflow Graph Config to define workflows.

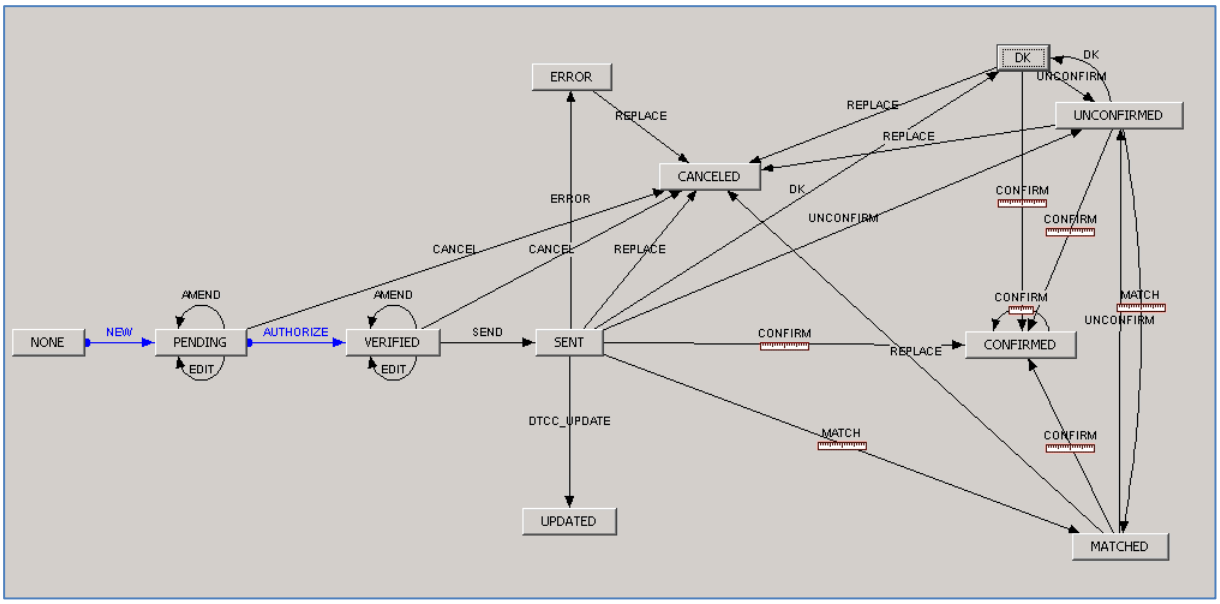
You can import the default Workflows in the folder resources DTCCMsgWorkflows.wf, DTCCNovationsTradeWorkflow.wf, DTCCNovationsMsgWorkflows.wf.

Message type = DTCC_CONFIRM

Orig Status	Action	Resulting Status	Use STP	Subtype	Rules
CONFIRMED	CONFIRM	CONFIRMED	FALSE	DTCC_CONFIRM	CleanupAlleged
DK	CONFIRM	CONFIRMED	FALSE	DTCC_CONFIRM	CleanupAlleged
DK	REPLACE	CANCELED	FALSE	DTCC_CONFIRM	
DK	UNCONFIRM	UNCONFIRMED	FALSE	DTCC_CONFIRM	
ERROR	REPLACE	CANCELED	FALSE	DTCC_CONFIRM	
MATCHED	CONFIRM	CONFIRMED	FALSE	DTCC_CONFIRM	CleanupAlleged
MATCHED	REPLACE	CANCELED	FALSE	DTCC_CONFIRM	
MATCHED	UNCONFIRM	UNCONFIRMED	FALSE	DTCC_CONFIRM	
NONE	NEW	PENDING	TRUE	DTCC_CONFIRM	
PENDING	AMEND	PENDING	FALSE	DTCC_CONFIRM	
PENDING	AUTHORIZE	VERIFIED	TRUE	DTCC_CONFIRM	
PENDING	CANCEL	CANCELED	FALSE	DTCC_CONFIRM	
PENDING	EDIT	PENDING	FALSE	DTCC_CONFIRM	
SENT	CONFIRM	CONFIRMED	FALSE	DTCC_CONFIRM	CleanupAlleged
SENT	DK	DK	FALSE	DTCC_CONFIRM	
SENT	DTCC_UPDATE	UPDATED	FALSE	DTCC_CONFIRM	
SENT	ERROR	ERROR	FALSE	DTCC_CONFIRM	
SENT	MATCH	MATCHED	FALSE	DTCC_CONFIRM	CleanupAlleged
SENT	REPLACE	CANCELED	FALSE	DTCC_CONFIRM	
SENT	UNCONFIRM	UNCONFIRMED	FALSE	DTCC_CONFIRM	
UNCONFIRMED	CONFIRM	CONFIRMED	FALSE	DTCC_CONFIRM	CleanupAlleged
UNCONFIRMED	DK	DK	FALSE	DTCC_CONFIRM	
UNCONFIRMED	MATCH	MATCHED	FALSE	DTCC_CONFIRM	CleanupAlleged
UNCONFIRMED	REPLACE	CANCELED	FALSE	DTCC_CONFIRM	
VERIFIED	AMEND	VERIFIED	FALSE	DTCC_CONFIRM	
VERIFIED	CANCEL	CANCELED	FALSE	DTCC_CONFIRM	
VERIFIED	EDIT	VERIFIED	FALSE	DTCC_CONFIRM	

Orig Status	Action	Resulting Status	Use STP	Subtype	Rules
VERIFIED	SEND	SENT	FALSE	DTCC_CONFIRM	

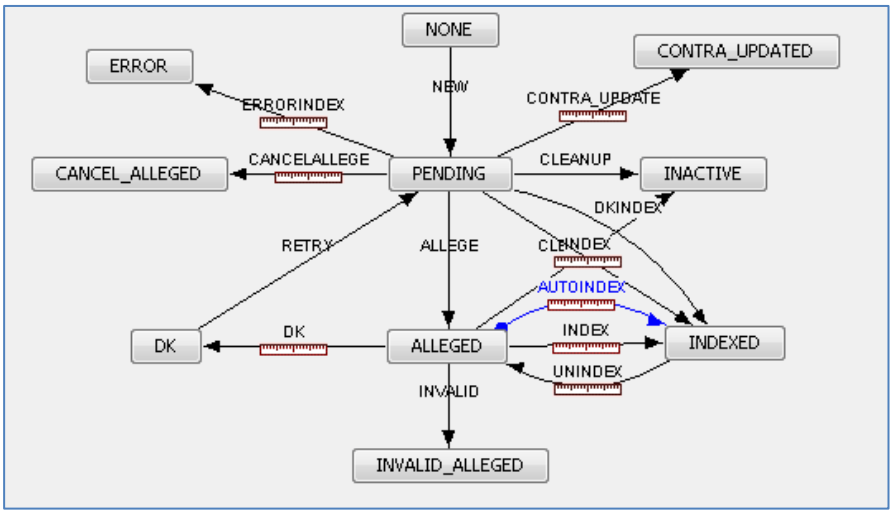
DTCC_UPDATE action is hardcoded. It is used in case of Workflow Update. The message will apply DTCC_UPDATE when the acknowledgment message corresponding to a Workflow Update (sent from PO): the corresponding incoming message is indexed. IncomingReply Message WorkflowRule applies DTCC_UPDATE action on the outgoing message.



Message type = INCOMINGCONF

Orig Status	Action	Resulting Status	Use STP	Subtype	Rules
ALLEGED	AUTOINDEX	INDEXED	true	INCOMINGCONF	IndexAlleged
ALLEGED	CLEANUP	INACTIVE	false	INCOMINGCONF	
ALLEGED	DK	DK	false	INCOMINGCONF	DKAlleged
ALLEGED	INDEX	INDEXED	false	INCOMINGCONF	IndexAlleged
ALLEGED	INVALID	INVALID_ALLEGED	false	INCOMINGCONF	
DK	RETRY	PENDING	false	INCOMINGCONF	
INDEXED	UNINDEX	ALLEGED	false	INCOMINGCONF	UnIndexDTCC
NONE	NEW	PENDING	false	INCOMINGCONF	
PENDING	ALLEGE	ALLEGED	false	INCOMINGCONF	
PENDING	CANCELALLEGE	CANCEL_ALLEGED	false	INCOMINGCONF	CancelAlleged

Orig Status	Action	Resulting Status	Use STP	Subtype	Rules
PENDING	CLEANUP	INACTIVE	false	INCOMINGCONF	
PENDING	DKINDEX	INDEXED	false	INCOMINGCONF	
PENDING	ERRORINDEX	ERROR	false	INCOMINGCONF	SendError
PENDING	CONTRA_UPDATE	CONTRA_UPDATED	false	INCOMINGCONF	IncomingReply
PENDING	INDEX	INDEXED	false	INCOMINGCONF	IncomingReply



After the incoming message is imported, it follows the workflow for the message type INCOMINGCONF:

- If the message does not have a trade/message reference and the DTCC status is alleged, the ALLEGE action is applied.
- If the message does not have a trade/message reference and the DTCC status is cancel alleged, the CANCELALLEGE action is applied.
- If the message has a trade/message reference and the DTCC status is DK acknowledged, the DKINDEX action is applied.
- If the message has a trade/message reference and the DTCC status is different from DK acknowledged, the INDEX action is applied.
- If the message has a trade/message reference and the DTCC status is Contra Updated, the CONTRA_UPDATE action is applied.

  It is an acknowledgement message corresponding to a WorkFlow Update from the counterparty side, the incoming message does not need to be indexed to any outgoing message, but the corresponding trade can be updated with some new information (ex :counterparty SuperId field)

Message type = INCOMING_UNKNOWN

Orig Status	Action	Resulting Status	Use STP	Subtype	Rules
NONE	NEW	PENDING	false	INCOMING_UNKNOWN	

This message type only exists to store incoming messages received from DTCC that are not well formatted and cannot be parsed using FPML/DTCC schemas.

For example you can receive INCOMING_UNKNOWN messages when there is a schema version mismatch between messages you send and messages you receive. In this case this message type allows keeping a track of received messages even if they cannot be processed.

Message type = DKCONF

A DKCONF message is automatically created by the system when the DK action is applied on a message in ALLEGED status.

The DKCONF message does not need a message configuration.

Orig Status	Action	Resulting Status	Use STP	Log	Subtype	Rules
NONE	NEW	PENDING	True	false	DKCONF	
PENDING	SEND	SENT	True	false	DKCONF	
SENT	DKERROR	DK_ERROR	False	false	DKCONF	
SENT	DKACKNOWLEDGE	DK_ACKNOWLEDGED	False	false	DKCONF	

Workflow Rules Description

Rule name	Description
CheckMasterConfirmationTradeRule	This rule returns : - False if there is no Master Confirmation signed for a given issuer seniority and region.
DTCCCleanupGTRUSI	This rule performs the following: clean the USI information keywords for a trade submitted to a Novation Consent
CancelAllegedMessageRule	This rule performs the following: Looks for corresponding Alleged messages from DTCC (same Contra trade id and same contra trade party reference) in the system and applies the action INVALID on it.
CleanUpAllegedMessageRule	This rule performs the following: If the CONFIRM action is applied to a message then it looks for messages in the system with an ALLEGED or INDEXED_ALLEGED status for the same trade (same contra trade id, same contra trade party

Rule name	Description
	reference, same contra trade reference number supplement and same activity), and moves them to INACTIVE status using the action CLEANUP.
CleanupIncomingNovationRequest	This rule performs the following: If the message is an IncomingNovationRequest, it applies the CLEANUP action on all existing DTCCNovationRequest messages for the same novationconsent id, same sender and receiver.
CleanupOutgoingNovationRequest	This rule performs the following: If the message is an DTCCNovationRequest, it applies the CLEANUP action on all existing DTCCNovationRequest messages for the same novationconsent id, same sender and receiver.
DKAllegedMessageRule	This rule performs the following: If the DK action is applied on a message in ALLEGED status it creates a new message with type DKCONF which is sent out to DTCC.
IncomingReplyMessageRule	This rule locates a matching outgoing message in the system and applies the appropriate action depending on the status sent by DTCC. It updates the outgoing message. It also updates the trade keyword CONFIRMED to Y, once you get a confirmed message from DTCC and sets counterparty trade id on the trade with contra trade id when available in the incoming message.
IndexAllegedMessageRule	This rule performs the following: When manual indexing is performed on an Alleged message, it updates the trade's external reference Id with the cpty DTCC id.
UnIndexDTCCMessageRule	This rule performs the following: If the UNINDEX action is applied on an Indexed message (Alleged from Cpty), it locates the indexed trade and un-indexes it.

5.2.6 Indexing Setup

The IncomingReply message rule indexes all INCOMINGCONF message based on:

- Outgoing Calypso message id when provided in the incoming message

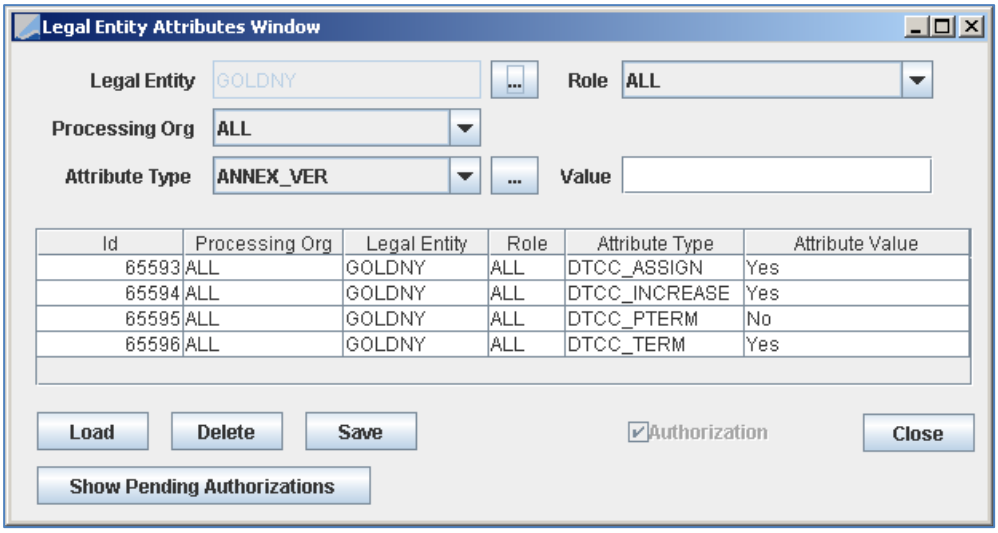
- Trade Reference Number and trade Reference Number supplement, which are stored on any outgoing message as message attributes.

5.2.7 DTCC Transaction Types

All DTCC transaction types are not supported by all counterparties.

If a transaction type is not supported by a particular counterparty, then the confirmation has to be sent by fax instead of DTCC.

This information may be stored on legal entity attributes on the counterparty (one for each transaction type: Trade, Termination, Increase, Exit, and Assignment).



These attributes can be used by the static data filters on message configurations.

5.3 Processing DTCC Confirmations

DTCC communicates with its members via dedicated lines and the message transfer protocol is MQSeries.

The file `modules/dtcc/resources/calypso_dtcc_config.properties` contains MQSeries and JMS messaging properties.

5.3.1 Generating DTCC Confirmations

DTCC confirmations are generated by the message engine based on master confirmations and message configurations. They are then sent to DTCC using the sender engine through the DTCC gateway.

The message engine can be started using `com.calypso.apps.startup.StartMessageEngine`.

The message engine publishes message events (PSEventMessage) and saves the messages to the database (BOMessage objects). It also publishes task events. All exceptions are created with the source "MessageEngine" and the exception type "MESSAGE_SETUP".

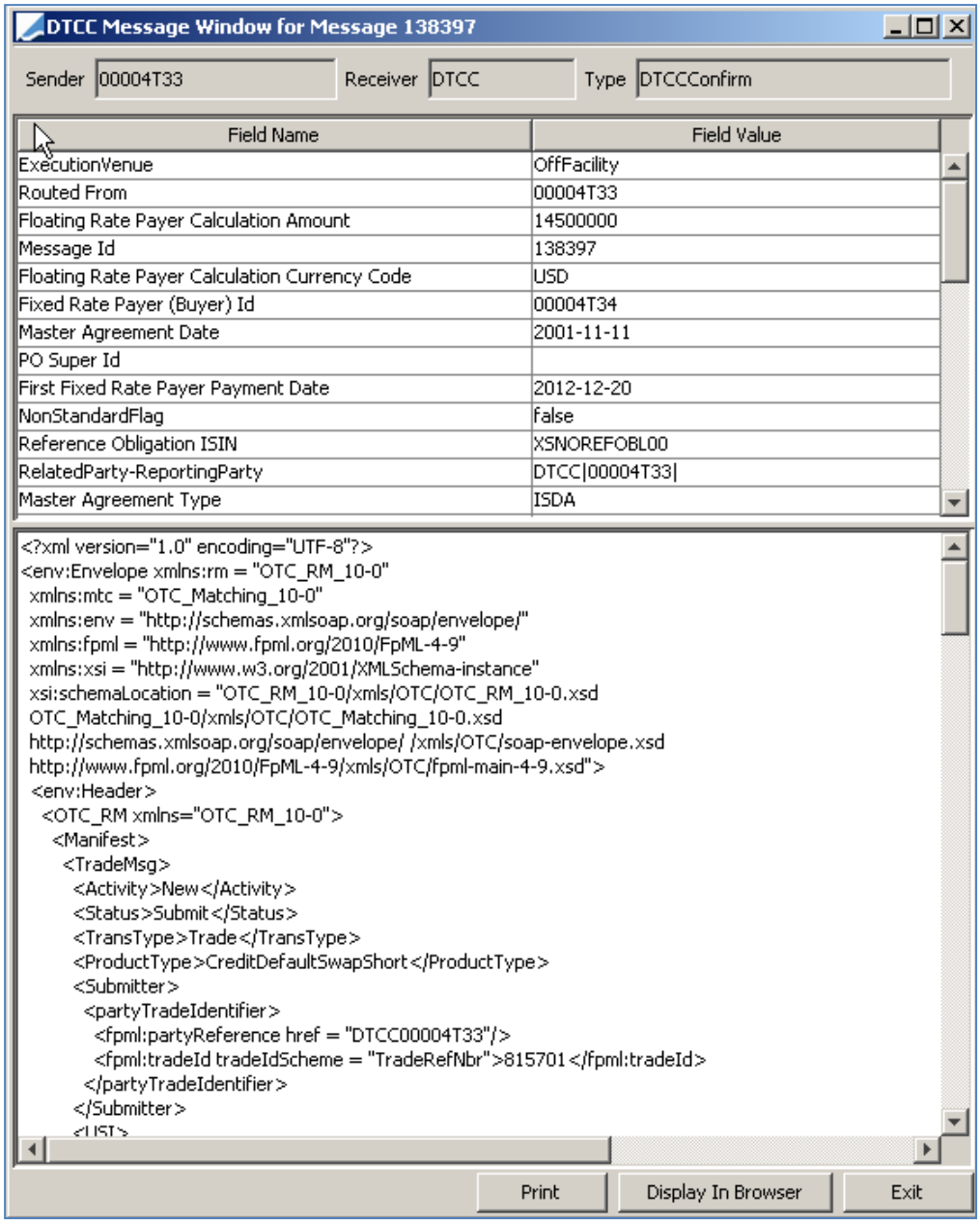
Whenever the message engine is started, it processes all outstanding events to which it subscribes that were published while it was inactive.

The sender engine can be started using `com.calypso.apps.startup.StartSenderEngine`.

Reviewing DTCC Confirmations

You can review the generated DTCC confirmations from the Back Office Window (on a trade), or using the Message report.

Double-click a message to view its details.



You can click **Display in Browser** for a full view of the XML file.

Note: Depending on the default XML editor of your computer, the file may open in Internet Explorer or another application.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:rm = "OTC_RM_10-0"
  xmlns:mtc = "OTC_Matching_10-0"
  xmlns:env = "http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:fpml = "http://www.fpml.org/2010/FpML-4-9"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "OTC_RM_10-0/xmls/OTC/OTC_RM_10-0.xsd
  OTC_Matching_10-0/xmls/OTC/OTC_Matching_10-0.xsd
  http://schemas.xmlsoap.org/soap/envelope/ /xmls/OTC/soap-envelope.xsd
  http://www.fpml.org/2010/FpML-4-9/xmls/OTC/fpml-main-4-9.xsd">
    <env:Header>
        <OTC_RM xmlns="OTC_RM_10-0">
            <Manifest>
                <TradeMsg>
                    <Activity>New</Activity>
                    <Status>Submit</Status>
                    <TransType>Trade</TransType>
                    <ProductType>CreditDefaultSwapShort</ProductType>
                    <Submitter>
                        <partyTradeIdentifier>
                            <fpml:partyReference href = "DTCC00004T33"/>
                            <fpml:tradeId tradeIdScheme = "TradeRefNbr">815701</fpml:tradeId>
                        </partyTradeIdentifier>
                    </Submitter>
                    <USI>
                        <USIIssuer issuerIdScheme="http://www.fpml.org/coding-scheme/external/cftc/issuer-identifier">BRANCHE33_Calypso</USIIssuer>
                        <USITradeId tradeIdScheme="http://www.fpml.org/coding-scheme/external/unique-transaction-identifier">815701</USITradeId>
                    </USI>
                </TradeMsg>
                <MsgId>138397</MsgId>
            </Manifest>
            <Delivery>
                <RouteInfo>
                    <From>DTCC00004T33</From>
                    <To>DTCC</To>
```

5.3.2 Importing Confirmations from DTCC

DTCC confirmations are dynamically imported using the import message engine listening to an IEAdapter.

The file `modules/dtcc/resources/calypso_dtcc_config.properties` contains MQSeries and JMS messaging properties.

You can set CalypsoToDTCC.queue.setContext=false to turn off MQOO_SET_ALL_CONTEXT and MQPMO_SET_IDENTITY_CONTEXT. It is true by default.

You can start the import message engine using `com.calypso.apps.startup.StartImportMessageEngine` with the argument `-config DTCC`.

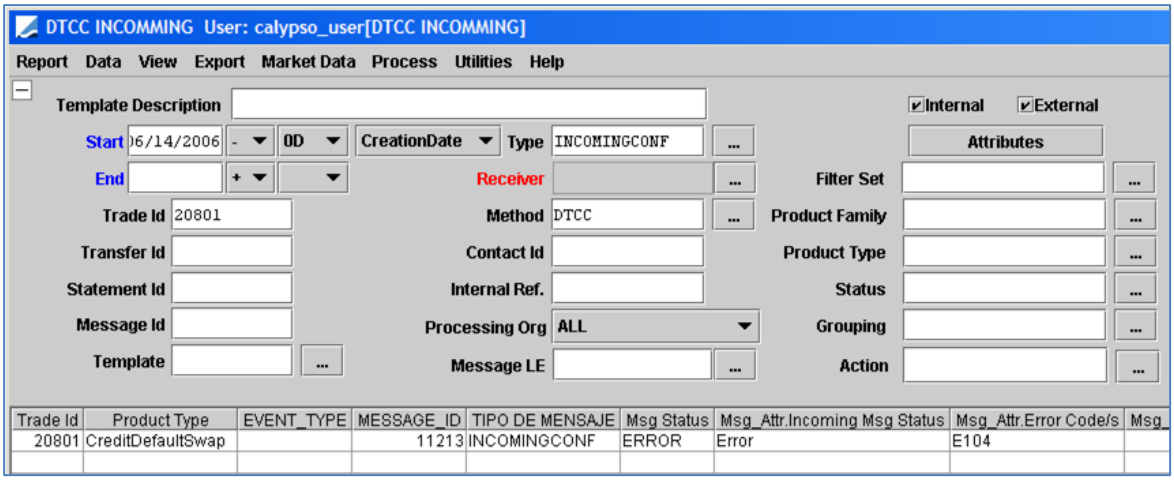
Based on the status of the incoming confirmations, Calypso confirmations will be confirmed or not. The various cases are described below.

Transaction Type = Trade

Enter a new trade, a CONFIRM will be sent to DTCC. The outgoing message will follow the workflow for the CONFIRM message type.

The incoming message will follow the workflow for the INCOMINGCONF message type.

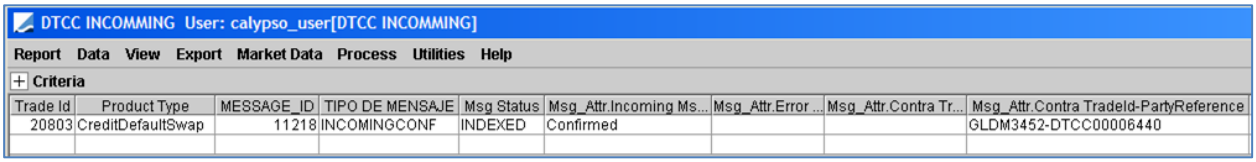
- Error Message — The incoming message indicates that the outgoing message is invalid.



The incoming rule SendError applies the action ERRORINDEX on the incoming message, and it moves to status ERROR. The DTCC error code is stored in the message attribute "Error Code/s".

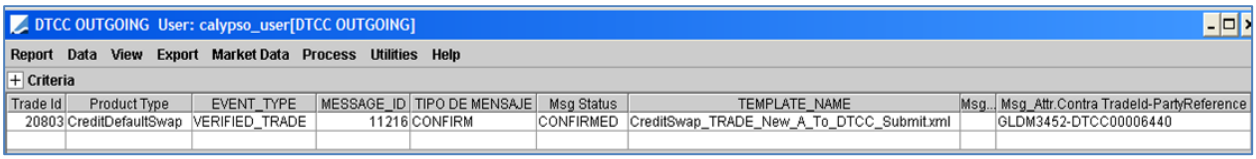
The incoming rule SendError applies the action ERROR on the outgoing message, and it moves to status ERROR. You can modify the trade, so that the outgoing message is amended and regenerated.

- Confirmed Message — The incoming message indicates that the outgoing message is valid and matched 100%.



The incoming rule IncomingReply applies the INDEX action to the incoming message, and it moves to status INDEXED.

The incoming rule IncomingReply applies the CONFIRM action to the outgoing message, and it moves to status CONFIRMED.



The trade keyword CONFIRMED is set to Y.

- Unconfirmed Message — The incoming message indicates that the outgoing message is valid but matched less than 100%. The outgoing message will move to UNCONFIRMED.

  The incoming rule IncomingReply applies the INDEX action to the incoming message, and it moves to status INDEXED.

  The incoming rule IncomingReply applies the UNCONFIRM action to the outgoing message, and it moves to status UNCONFIRMED.

Transaction Type = Amendment

Note that the transaction type Amendment only applies to trades that have been confirmed by DTCC (CONFIRMED trade keyword = Y).

Any modification will generate an DTCC message of type AMENDMENT that will be sent to DTCC.

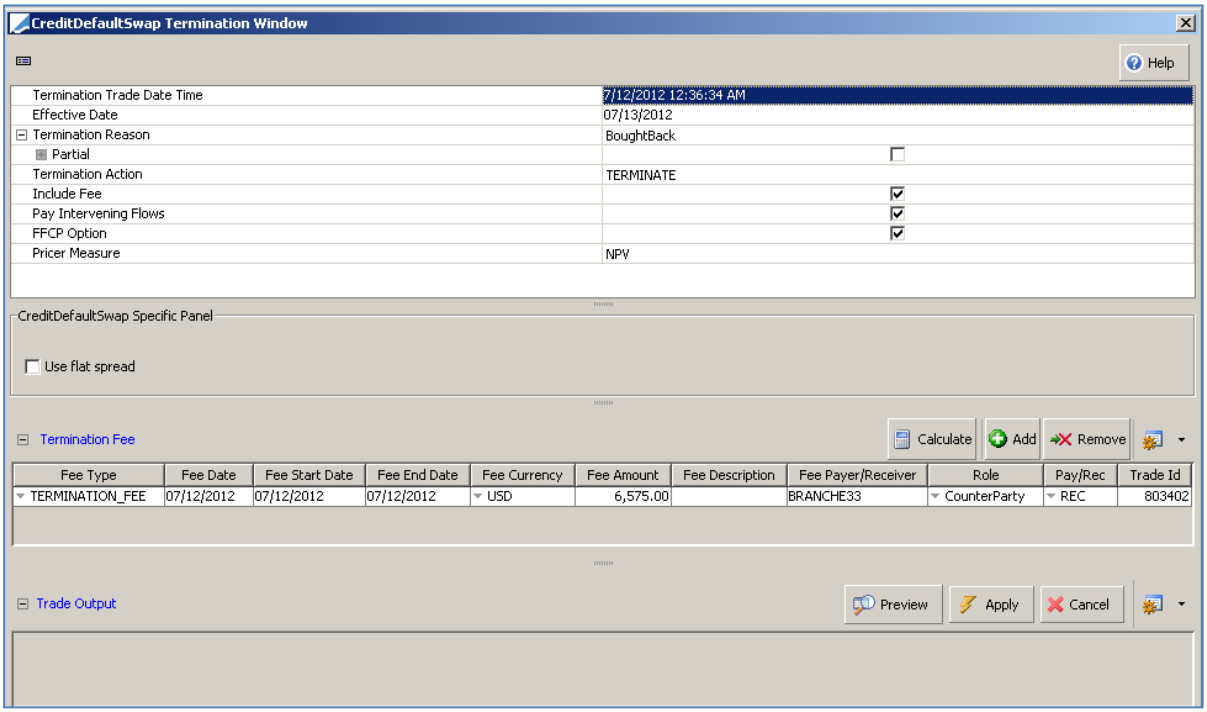
For a notional decrease/increase, see the transaction types Partial Termination and Increase.

The incoming message will follow the workflow for the INCOMINGCONF message type.

The incoming message may return an error, a confirmed message, or an unconfirmed message as for the transaction type Trade.

Transaction Type = Termination (full and partial termination)

To terminate a trade, bring up the Termination window from the trade.



Select BoughtBack from the Term Reason field, check the "Partial Termination" checkbox in case of a partial termination and populate the termination trade date, termination effective date and fee details.

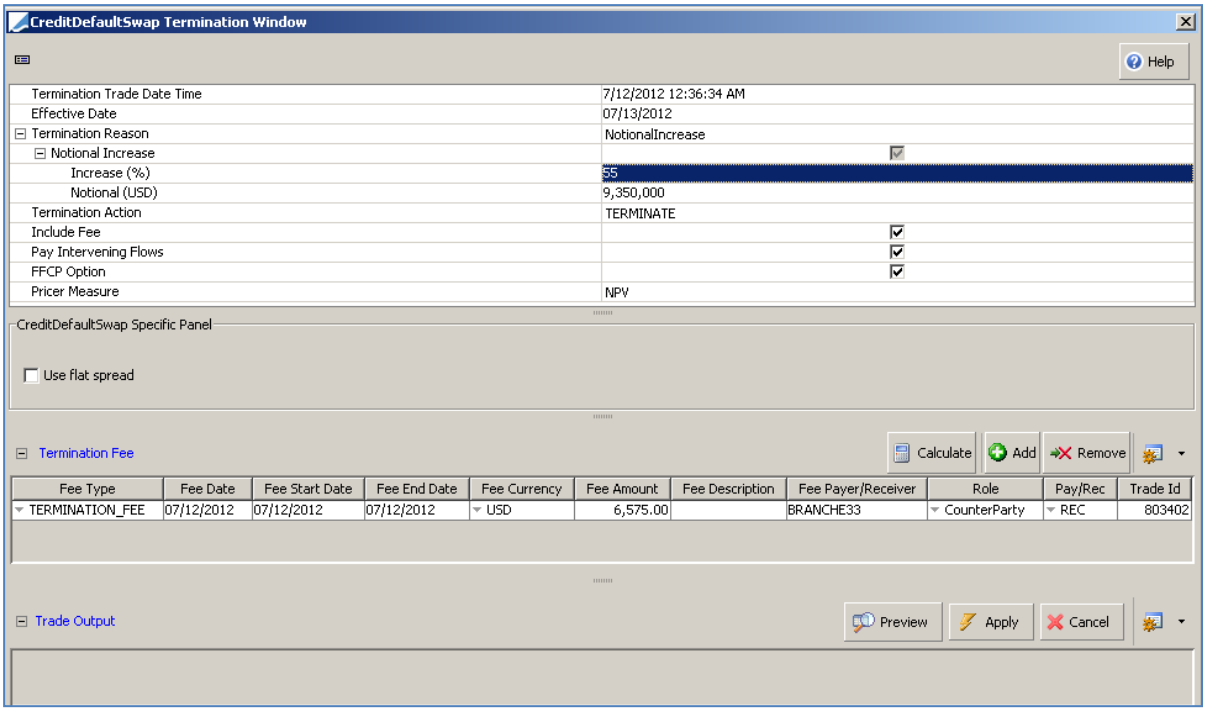
A new DTCC message of type TERMINATION will be created once the trade reaches the TERMINATED status.

The incoming message will follow the workflow for the INCOMINGCONF message type.

The incoming message may return an error, a confirmed message, or an unconfirmed message as for the transaction type Trade.

Transaction Type = Notional Increase

To partially increase the amount of a trade, bring up the Termination window from the trade.



Select NotionalIncrease from the Term Reason field, and click the Notional Increase checkbox.

Enter a percentage of increase in the % field.

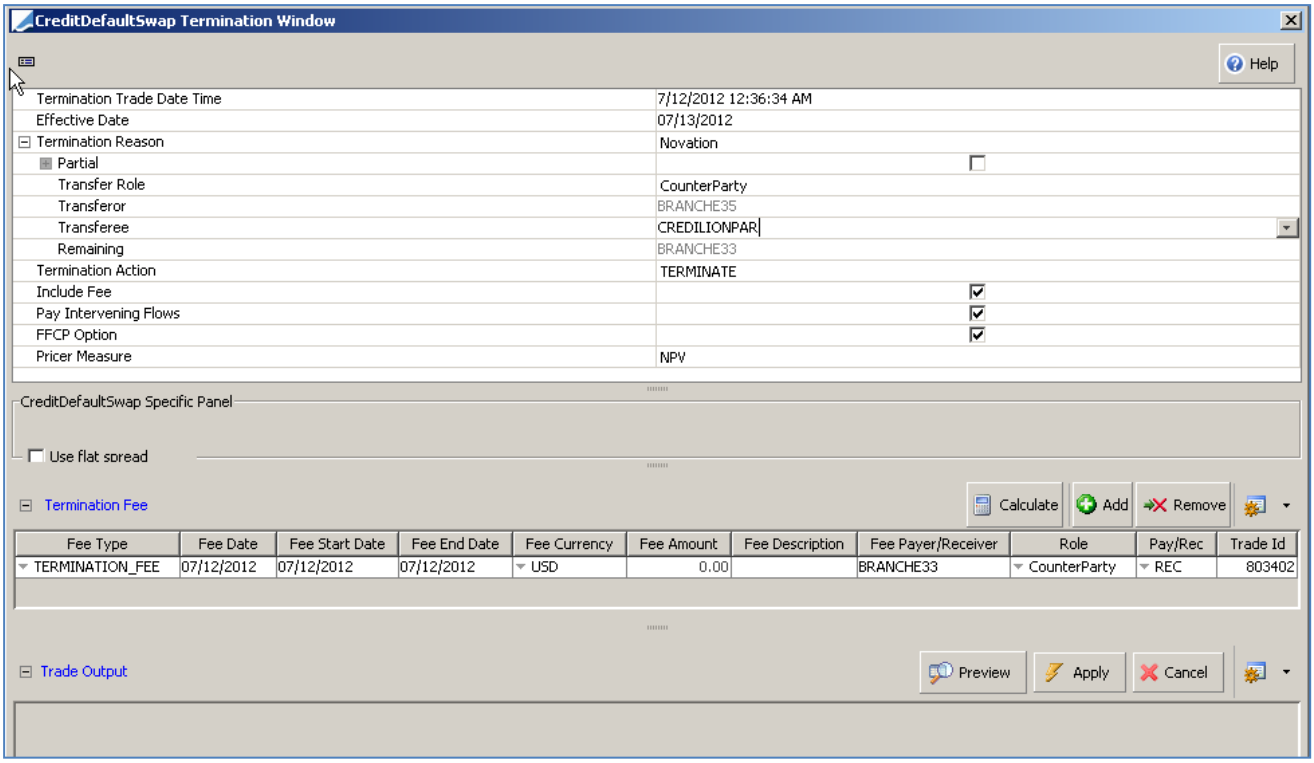
The incoming message will follow the workflow for the INCOMINGCONF message type.

The incoming message may return an error, a confirmed message, or an unconfirmed message as for the transaction type Trade.

Transaction Type = Assignment (Novation)

- PO is either remaining party or transferor

To novate a trade, bring up the Termination window from the trade.

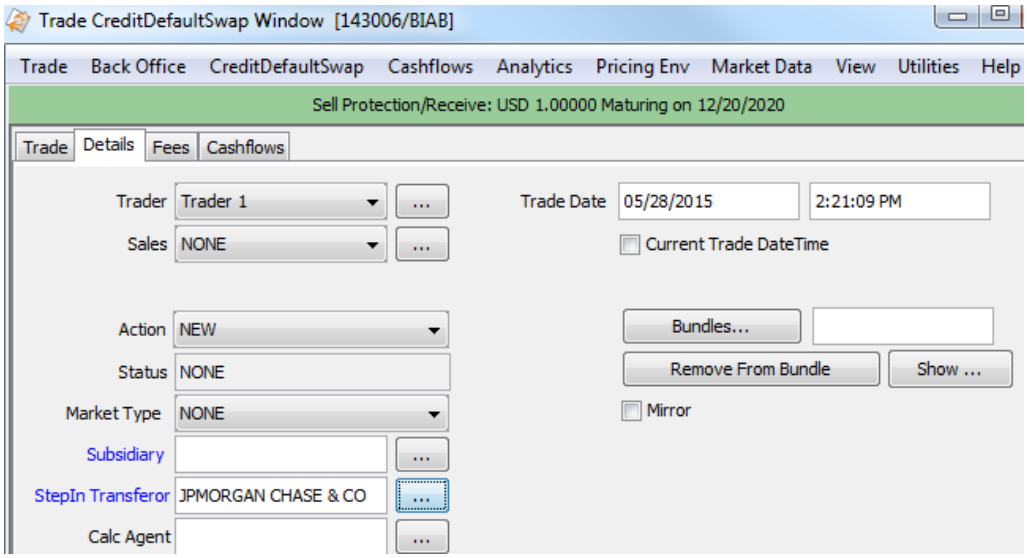


Select Novation from the Term Reason field, and select the transfer role and the transferee.

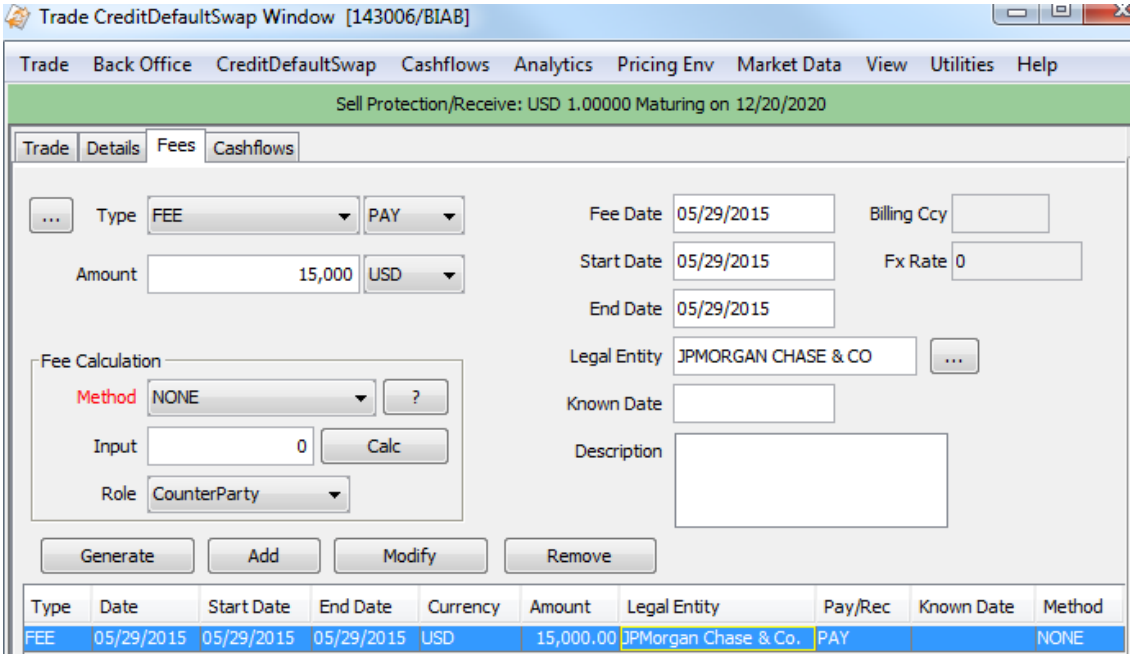
PO is transferee

This case is similar to a new trade, with additional information specific to a step-in novation:

enter the name of transferor in the "Transfer From" field (Detail Tab of trade window)



if necessary, enter novation fee. In order to see it in DTCC messagen fee type has to be "FEE"



The incoming message will follow the workflow for the INCOMINGCONF message type.

The incoming message may return an error, a confirmed message, or an unconfirmed message as for the transaction type Trade.

Alleged Messages

These are messages from DTCC notifying a counterparty's intention to match a specific activity on a specific transaction type.

For alleged messages, there will be a reference to a Calypso trade_id when the transaction type of the alleged message is a post-trade transaction (trade is confirmed already in DTCC) but there won't be any reference to a Calypso trade_id in case of a new trade.

- Alleged message is a New Trade transaction

In this case, the user will use DTCC website and reporting tool to find the possible matching trades that best correspond to the alleged message.

- Alleged Message is a post trade transaction

In this case, the alleged message is indexed to its corresponding Calypso trade so that the user can easily act on the corresponding Calypso trade in order to process the counterparty request.

If you cannot find a matching outgoing message, and think the message you received should have been sent to somebody else, apply the DK action to the incoming message. The rule DKAlleged will create a message of type DKCONF and send it to DTCC. When the incoming message for the DKCONF is imported, it will have the DTCC status DK acknowledged or error. The outgoing DKCONF message can then be moved to status DK_ACKNOWLEDGED or DK_ERROR (currently, the action has to be manually applied).

If the DTCC status is cancel alleged, the CANCELALLEGE action is applied to the incoming message, and it moves to CANCEL_ALLEGED status. The incoming rule CancelAlleged looks for the corresponding alleged message already in the system and applies the INVALID action.

5.4 TriParty Model

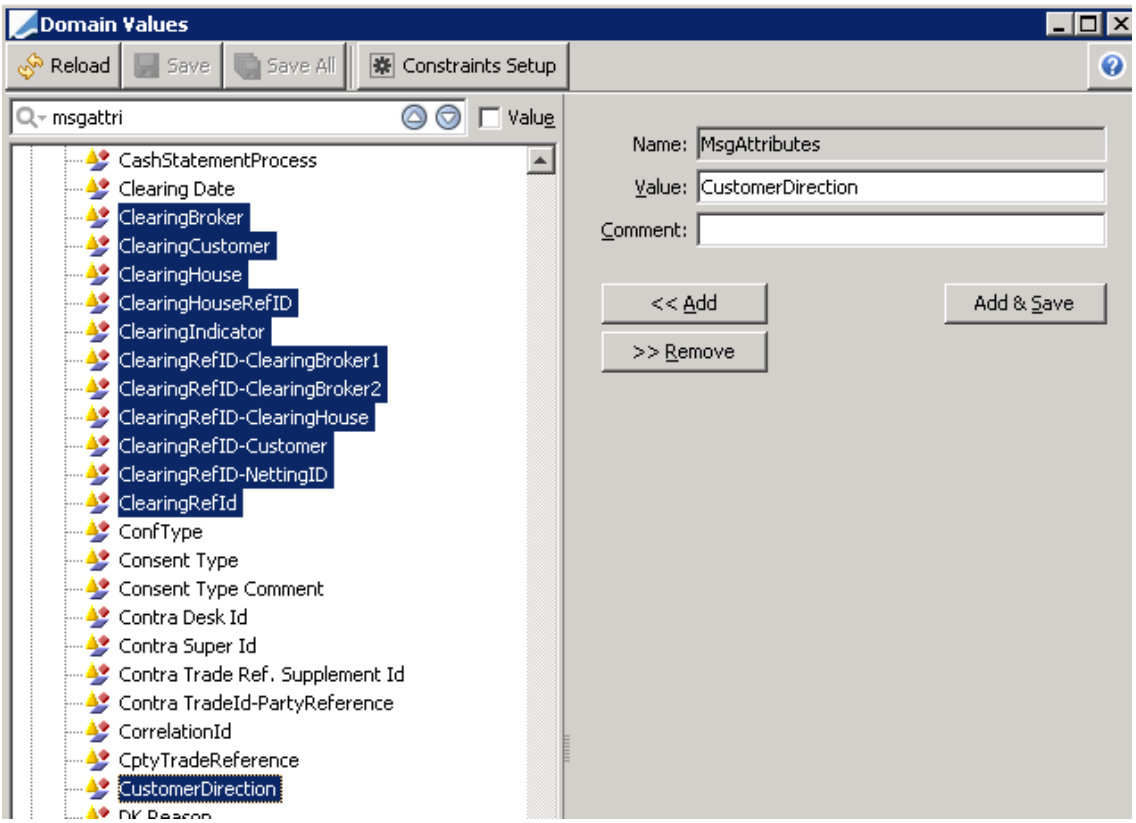
The DTCC V10.0 Credit Derivative release incorporates support for Triparty ticket model.
DTCC's triparty model provides a new set of messages dedicated to the process of clearing for CreditDefaultSwapShort and CreditDefaultSwapIndex

The table below explains the steps of the workflow:
Light green lines are not supported, and not covered in this module.

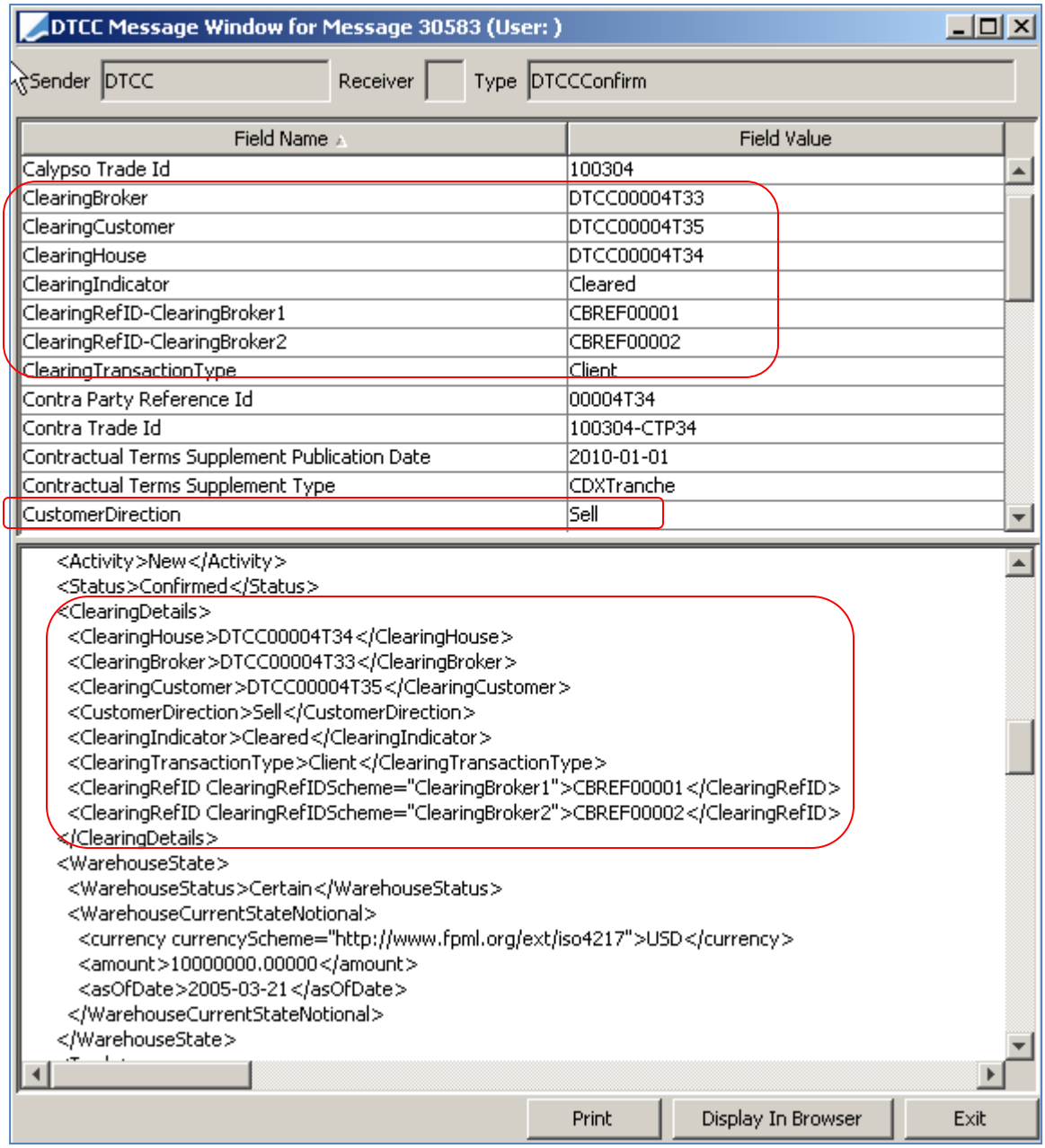
Trade	Step		Event	Transaction type	Initiator (Direction)
Request	1	1-1	New trade & Request to clear	New	Members → DTCC Affiliates → DTCC
		1-2	Request to clear	CCP report (FTP)	DTCC → CCP
Terminate (VIT) (vendor initiated termination)	2	2-1	Terminate original bilateral trades	VIT Termination	CCP → DTCC
		2-2	Termination confirmed	VIT Termination	DTCC → CCP DTCC → Members or Affiliates
Register	3	3-1	Registration CCP-facing trade	PreConfirmed	CCP → DTCC
		3-2	Registration confirmed	PreConfirmed	DTCC → CCP DTCC → Members
Post Clearing	4	4-1	TRI change – incoming	TRI change	Members → DTCC → CCP?
		4-2	TRI change - outgoing	TRI change	CCP → DTCC
	5	5-1	Early termination	VIT Termination	CCP → DTCC
		5-2	Termination confirmed	VIT Termination	DTCC → CCP DTCC → Members or Affiliates
	6		Novation	Novation	Members -> DTCC -> CCP?
	7		Amendment (VIA)	VIA Amendment	Members -> DTCC -> CCP?
	8		Workflow field change	WorkflowUpdate	Members -> DTCC -> CCP?
	9	9-1	Credit event of restructuring split-off single name trade input - outgoing	PreConfirmed	CCP → DTCC -> Members
		9-2	Credit event of restructuring split-off single name trade input - incoming	PreConfirmed	DTCC → CCP

5.4.1 Triparty messages / ClearingDetails

These TRIParty messages are comparable to regular messages of same transaction type, with the addition of the <ClearingDetails> tag. When such messages are imported in Calypso as a BO message, these fields are saved as message attributes, and then have to be declared in domain values MsgAttributes, as following:



For ClearingRefID which has multiple occurrences, we will name them as ClearingRefID-ClearingBroker1, ClearingRefID-ClearingBroker2, ClearingRefID-Customer, ClearingRefID-ClearingHouse



5.4.2 TRIChange

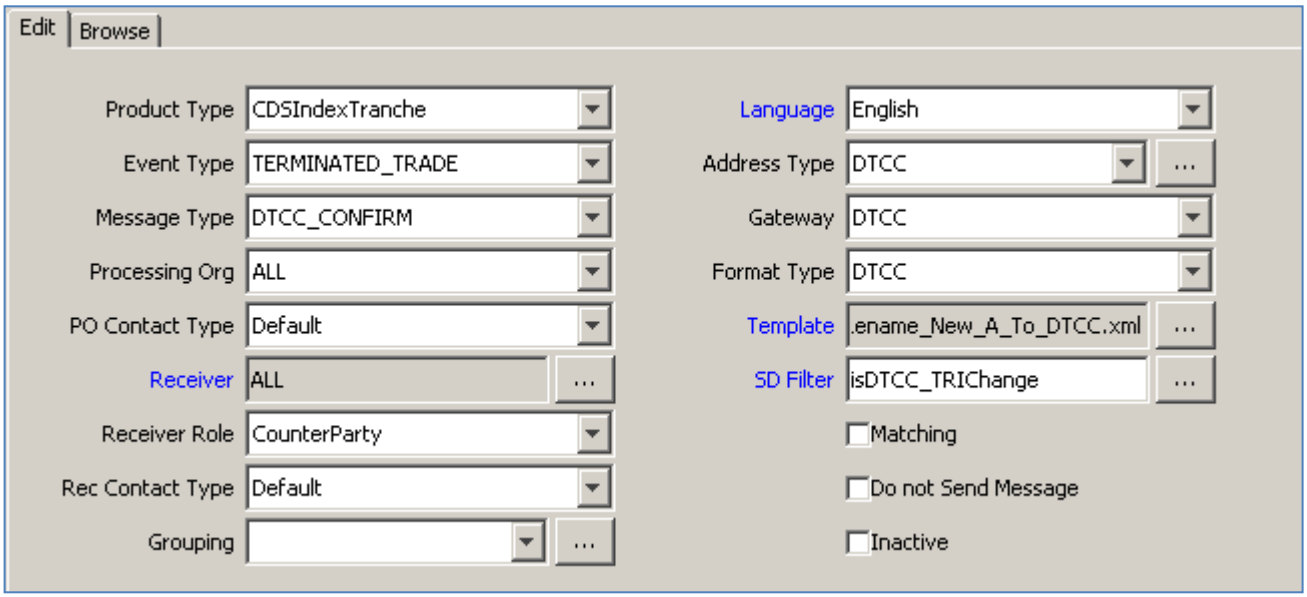
INCOMING TRICHANGE

TRI changes will become very common with the clearing process, because when a cleared trade is first booked as a Preconfirmed trade by a CCP, DTCC will assign it a reference for the clearing member.
But because the clearing member didn't choose this reference, he may be very likely to change it for a more convenient one.
When a TRI change message is received, IncomingReply message rule should update DTCC_CPTY_ID trade keyword if Counterparty TRI has changed: it will take in account 2 new dtcc message statuses: "YourTRIChanged" and "ContraTRIChanged".

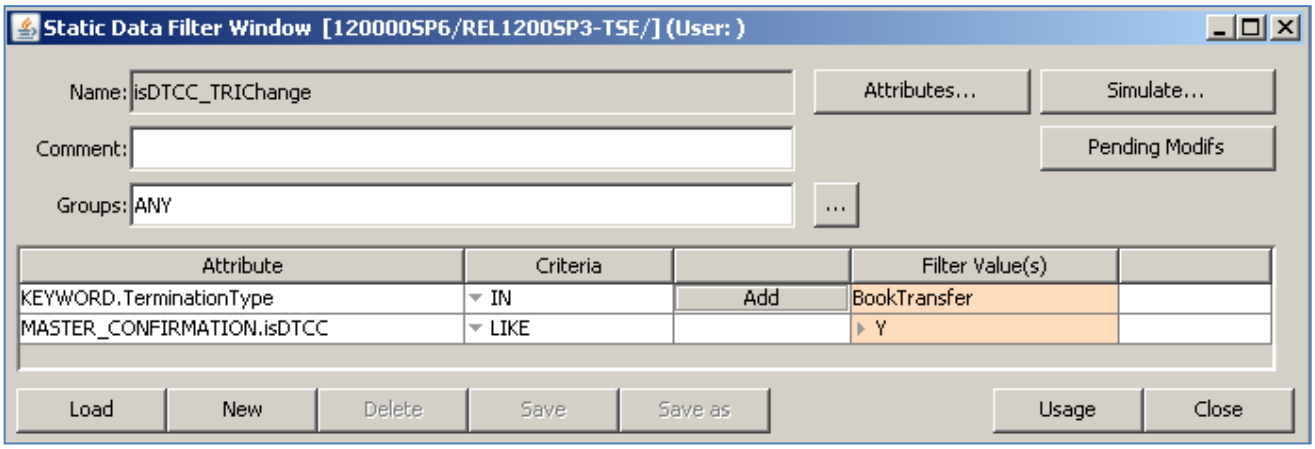
OUTGOING TRICHANGE

Trade Attribute (keyword) 'TransferTo' (on trade in status Terminated) will trigger the TRI change message.
The TRIChange will be triggered by a trade termination with the termination type **BookTransfer**.
This is only a request from one client, so Calypso does not change the termination template selector (**TerminationDTCCTemplateSelector**). Client will have to add a new line in its message setup referring to the template **TRIChange_Rename_New_A_To_DTCC.xml** combined with an appropriated Static Data Filter filtering **BookTransfer** termination type.

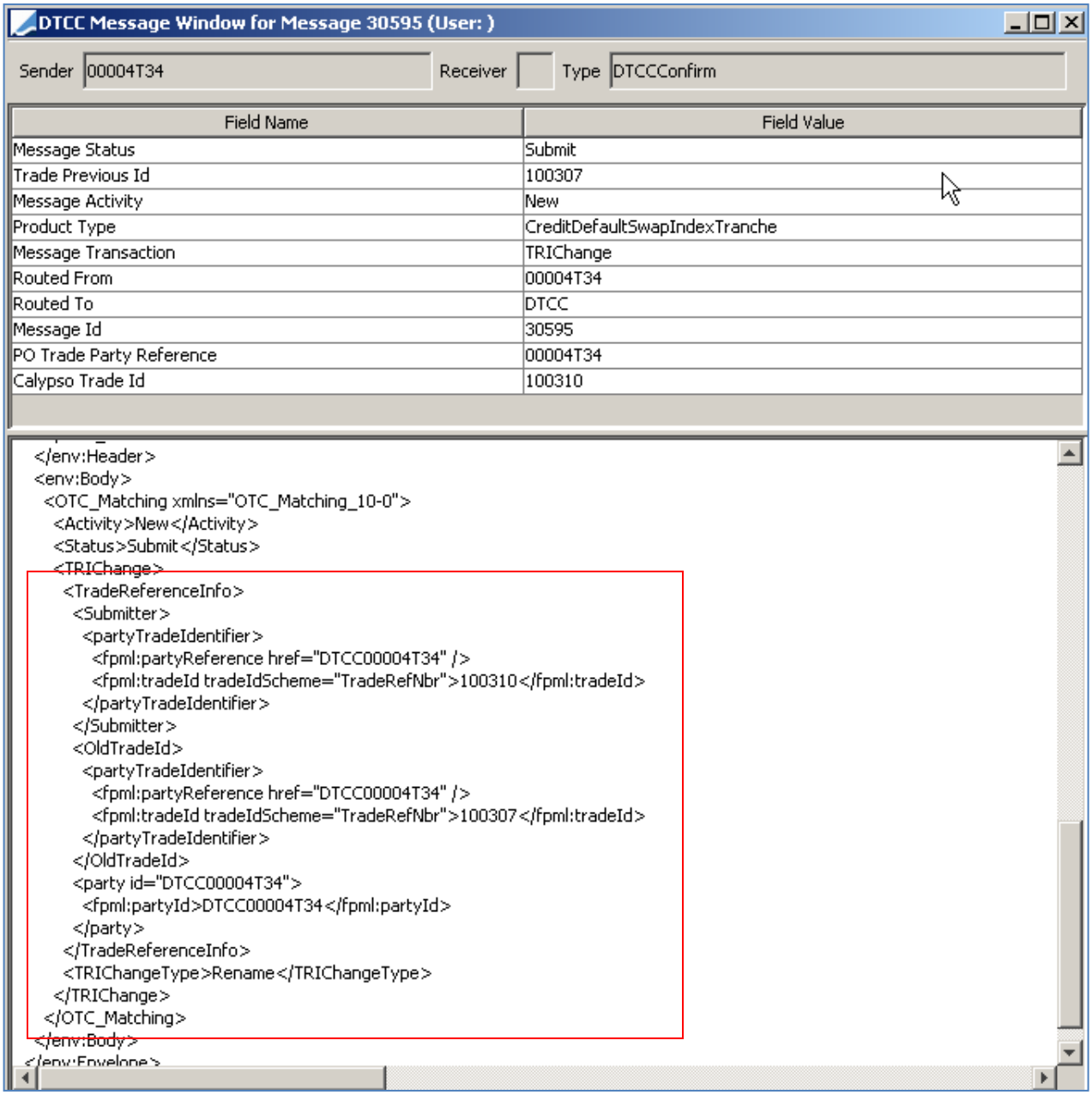
Sample Message set up:



With SD Filter:



TRIChange message example:



Section 6. DTCC Novation Consent

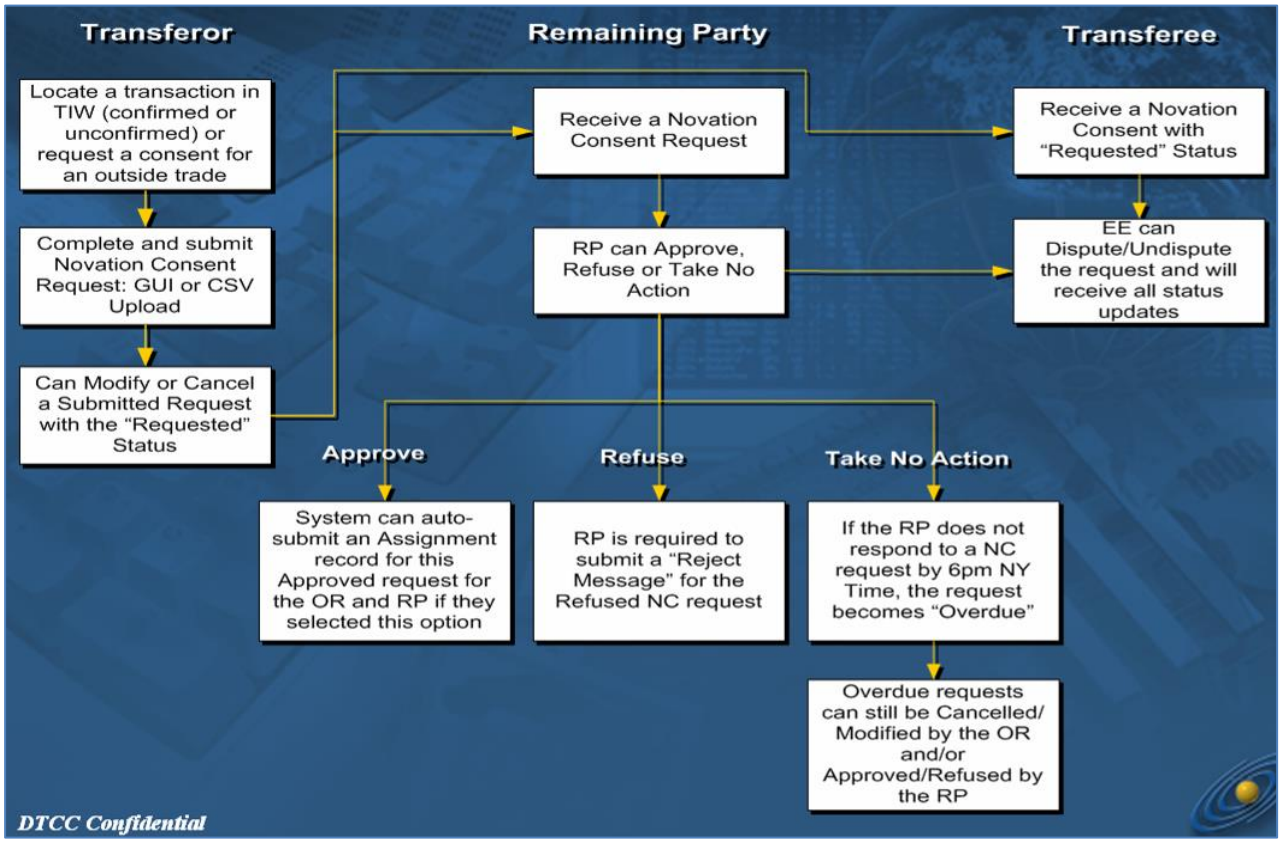
This section describes Calypso interface to DTCC Novation Consent service, an electronic novation consent service based on 2005 ISDA Novation Protocol.

Calypso module supports most of DTCC features, with some limitations:

- When the Processing Org is Transferor, and the novation request is created from Calypso, then Calypso only supports "inside" novation request (novation of trades confirmed in DTCC Trade Information Warehouse). Outside novation are supported in other cases (outside novation requests created from DTCC GUI when PO is Transferor, and all kind of request when PO is Remaining Party or Transferee.
- When the Processing Org is transferee, once the novation request is accepted, Calypso doesn't create the step-in trade automatically. Trade needs to be captured manually.

6.1 Presentation

6.1.1 DTCC general flowchart



Calypso replicates this flowchart, with the Processing Org being Transferor, Remaining Party or Transferee.

6.1.2 Procedure when the Processing Org is Transferor

Starting from Calypso

1. Trade is terminated through the termination window. At this stage, trade is NC_PENDING and not yet TERMINATED because of the pending novation request
2. NC_PENDING trade event triggers generation of a DTCCNovationRequest message, which is sent to DTCC
3. Remaining Party and Transferee receive the novation request and reply
4. Any status change of the novation request in DTCC is received in Calypso as an IncomingNovationRequest message. Original DTCCNovationRequest message status changes accordingly (REQUESTED, APPROVED, REFUSED, DISPUTED, OVERDUE)
5. Once the request is approved, trade is moved to NC_ACCEPTED, where a manual (or STP) approval can move it to the TERMINATED status.
6. TERMINATED trade event will trigger a Novation DTCC Confirmation message.
    a. Message is sent to DTCC as usual if novation request has been approved with AutoAssignement = false
    b. Message is never sent to DTCC is novation request has been approved with AutoAssignement = true (in this case, message setup can be designed in order not create any message at all).

Starting from DTCC Novation Consent GUI

1. Novation request is created from DTCC Novation Consent GUI. At this stage trade is VERIFIED in Calypso.
2. Remaining Party and Transferee receive the novation request and reply
3. Any status change of the novation request in DTCC is received in Calypso as an IncomingNovationRequest message.
4. Once the request is approved, a manual (or STP) approval on the IncomingNovationRequest terminates the trade.
5. TERMINATED trade event will trigger a Novation DTCC Confirmation message.
    a. Message is sent to DTCC as usual if novation request has been approved with AutoAssignement = false
    b. Message is never sent to DTCC is novation request has been approved with AutoAssignement = true (in this case, message setup can be designed in order not create any message at all).

6.1.3 Procedure when the Processing Org is Remaining Party

1. Remaining Party receives novation request from Transferor (Calypso IncomingNovationRequest with status RP_ALLEGED)
2. Remaining Party accepts, disputes or refuses the request, or takes no action.
3. REFUSE, DISPUTE and ACCEPT actions on the RP_ALLEGED message trigger the creation of a DTCCNovationReply message (with corresponding consent type)
4. Once the request is approved, a manual (or STP) approval on the IncomingNovationRequest terminates the trade.
5. Any status change of the novation request in DTCC is received in Calypso as an IncomingNovationRequest message. DTCCNovationReply message (if any) status changes accordingly.
6. Once the request is approved, a manual (or STP) approval on the IncomingNovationRequest terminates the trade.
7. TERMINATED trade event will trigger a Novation DTCC Confirmation message.
    c. Message is sent to DTCC as usual if novation request has been approved with AutoAssignement = false
    d. Message is never sent to DTCC is novation request has been approved with AutoAssignement = true (in this case, message setup can be designed in order not create any message at all).

6.1.4 Procedure when the Processing Org is Transferee

1. Transferee receives novation request from Transferor (Calypso IncomingNovationRequest with status EE_ALLEGED)
2. Transferee can dispute the request from DTCC GUI, otherwise wait until it's accepted / refused.
3. Once the request is approved, step-in trade can be captured manually in Calypso.

6.2 Calypso Setup

6.2.1 Additional settings

In order to use DTCC novation consent, run `DTCCSchemaData.xml` with the "DTCC novations" option checked.

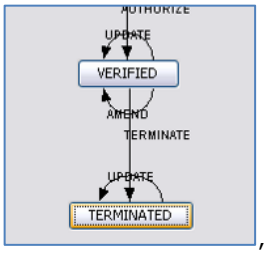
In order to create the DTCC transfer workflow, choose Navigator > Configuration > Workflow > Workflow Configuration. Then import the workflow contained in the `DTCCNovationsMsgWorkflow.wf` file.

6.2.2 Trade workflow

Note: this setup can be imported automatically in Calypso using the `DTCCNovationsTradeWorkflow.wf` file. This file doesn't create a full Trade workflow, but only create some additional statuses and actions in order to handle novation consent. Please note that this file will modify Trade workflow defined for PO=ALL, Product Type=ALL, Product SubType=ALL: if this is not the workflow used in your implementation, then modify the file accordingly before using it.

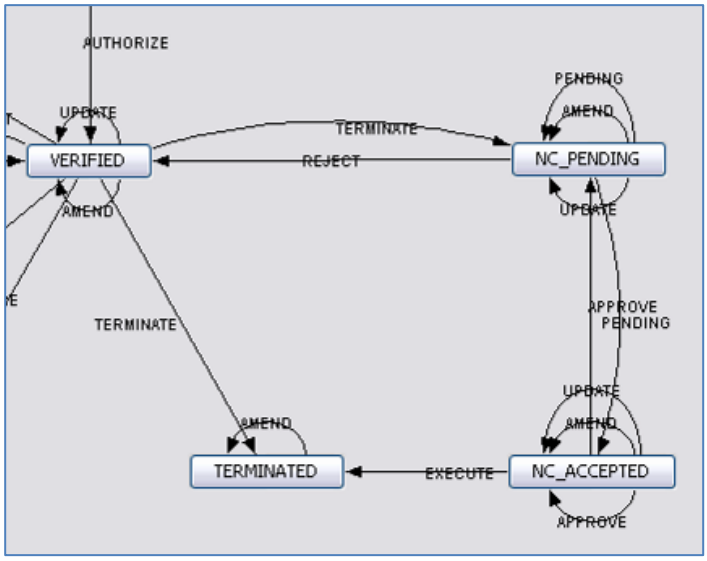
Trade workflow needs to be modified in order to handle DTCC novation request.

From original workflow like this one:



, add some statuses related to the novation consent lifecycle:

- NC_PENDING: novation consent in any status but accepted.
- NC_ACCEPTED: novation request is accepted



From VERIFIED, trade is terminated using the TERMINATE action. Resulting status is NC_PENDING when there is a novation consent to be processed, and TERMINATED otherwise.
The choice of one of these 2 options is driver by the CheckDTCCNovation and CheckNotDTCCNovation rules:

- CheckDTCCNovation is set on the VERIFIED – TERMINATE – NC_PENDING transition. This rule returns true if: Trade is confirmed in DTCC, has been novated, no novation request already exist and the PO is the transferor
- CheckNotDTCCNovation is set on the VERIFIED – TERMINATE – TERMINATED transition. This rule is the opposite rule of CheckDTCCNovation

With the implementation of reporting data, the DTCCCleanupGTRUSITradeRule has to be added in case of Novation RP, full or partial (Termination Reason= Novation and Transfer Role=Counterparty) that will remove USI/USIIssuer and USI/USITradeId on the new trade created between RP and EE.

Note about customizing the trade workflow: Actions resulting from incoming messages are applied by the DTCCApplyNovation Message Rule.

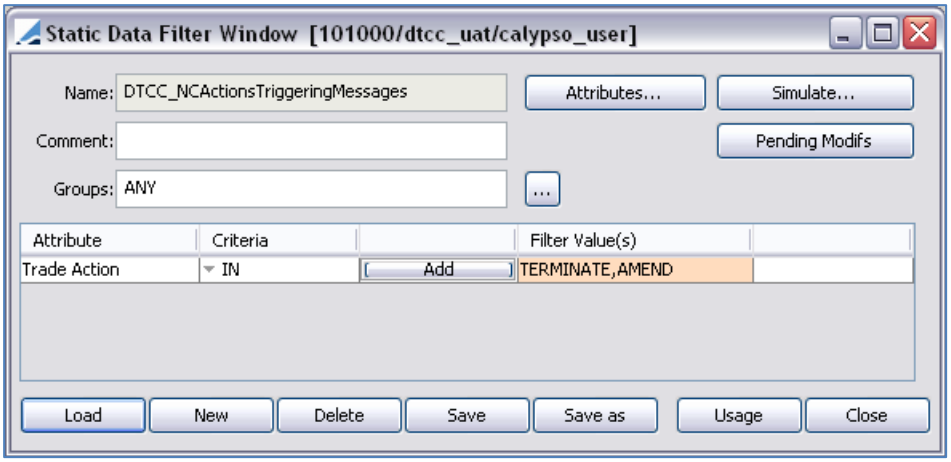
In the default setup, we kept the trade workflow as simple as possible with only 2 statuses: NC_PENDING and NC_ACCEPTED. Using domains **DTCCNCTradeAction.<consent type>.<consent status> or DTCCNCTradeAction.<consent type>** you can change this workflow to make it more accurate (but also more complex).

6.2.3 DTCCNovationRequest message setup and workflow

Message Setup

DTCCNovationRequest messages are triggered by NC_PENDING trade event:

Product	Event	Template Name	Static Filter
CreditDefaultSwap	NC_PENDING_TRADE	NovationConsent_New_Request_OR_To_DTCC.xml	DTCC_NCActionsTriggeringMessages
CDSIndex	NC_PENDING_TRADE	NovationConsent_New_Request_OR_To_DTCC.xml	DTCC_NCActionsTriggeringMessages
CDSIndexTranche	NC_PENDING_TRADE	NovationConsent_New_Request_OR_To_DTCC.xml	DTCC_NCActionsTriggeringMessages



Message Workflow

Message created and validated reach the VERIFIED status.

Then the sender engine processes them to DTCC.

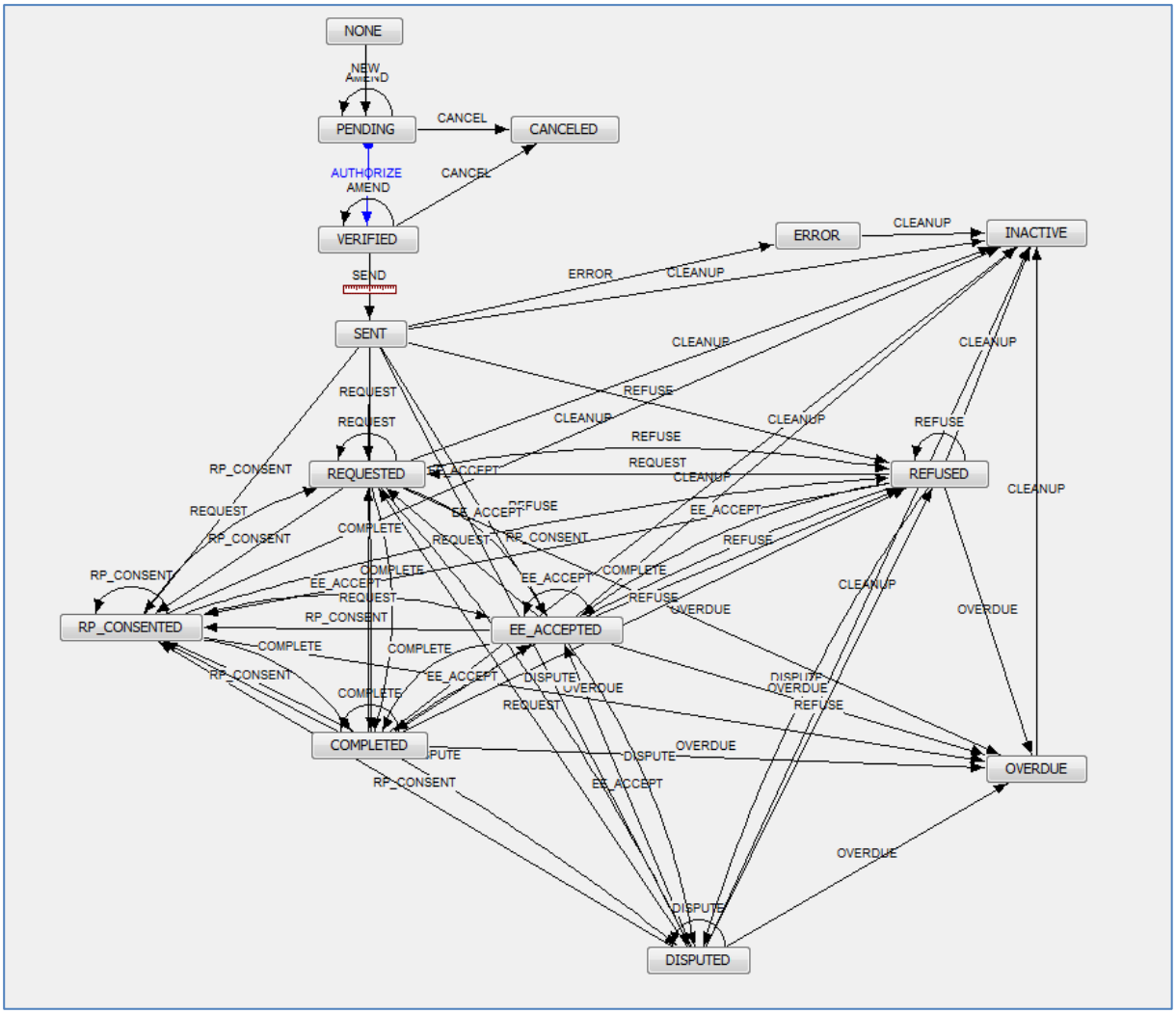
All statuses after SENT reflect the status of the last received incoming message (REQUESTED, APPROVED, REFUSED, DISPUTED, OVERDUE, or ERROR).

(the workflow definition is given in DTCCNovationMsgWorkflow.wf)

Orig Status	Action	Resulting Status	Use STP	Subtype	Rules
COMPLETED	CLEANUP	INACTIVE	FALSE	DTCCNovationRequest	
COMPLETED	COMPLETE	COMPLETED	FALSE	DTCCNovationRequest	
COMPLETED	EE_ACCEPT	EE_ACCEPTED	FALSE	DTCCNovationRequest	
COMPLETED	OVERDUE	OVERDUE	FALSE	DTCCNovationRequest	
COMPLETED	REFUSE	REFUSED	FALSE	DTCCNovationRequest	
COMPLETED	REQUEST	REQUESTED	FALSE	DTCCNovationRequest	
COMPLETED	RP_CONSENT	RP_CONSENTED	FALSE	DTCCNovationRequest	
DISPUTED	CLEANUP	INACTIVE	FALSE	DTCCNovationRequest	
DISPUTED	DISPUTE	DISPUTED	FALSE	DTCCNovationRequest	
DISPUTED	EE_ACCEPT	EE_ACCEPTED	FALSE	DTCCNovationRequest	
DISPUTED	OVERDUE	OVERDUE	FALSE	DTCCNovationRequest	
DISPUTED	REFUSE	REFUSED	FALSE	DTCCNovationRequest	
DISPUTED	REQUEST	REQUESTED	FALSE	DTCCNovationRequest	
DISPUTED	RP_CONSENT	RP_CONSENTED	FALSE	DTCCNovationRequest	

Orig Status	Action	Resulting Status	Use STP	Subtype	Rules
EE_ACCEPTED	CLEANUP	INACTIVE	FALSE	DTCCNovationRequest	
EE_ACCEPTED	COMPLETE	COMPLETED	FALSE	DTCCNovationRequest	
EE_ACCEPTED	DISPUTE	DISPUTED	FALSE	DTCCNovationRequest	
EE_ACCEPTED	EE_ACCEPT	EE_ACCEPTED	FALSE	DTCCNovationRequest	
EE_ACCEPTED	OVERDUE	OVERDUE	FALSE	DTCCNovationRequest	
EE_ACCEPTED	REFUSE	REFUSED	FALSE	DTCCNovationRequest	
EE_ACCEPTED	REQUEST	REQUESTED	FALSE	DTCCNovationRequest	
EE_ACCEPTED	RP_CONSENT	RP_CONSENTED	FALSE	DTCCNovationRequest	
ERROR	CLEANUP	INACTIVE	FALSE	DTCCNovationRequest	
NONE	NEW	PENDING	FALSE	DTCCNovationRequest	
OVERDUE	CLEANUP	INACTIVE	FALSE	DTCCNovationRequest	
PENDING	AMEND	PENDING	FALSE	DTCCNovationRequest	
PENDING	AUTHORIZE	VERIFIED	TRUE	DTCCNovationRequest	
PENDING	CANCEL	CANCELED	FALSE	DTCCNovationRequest	
REFUSED	CLEANUP	INACTIVE	FALSE	DTCCNovationRequest	
REFUSED	COMPLETE	COMPLETED	FALSE	DTCCNovationRequest	
REFUSED	DISPUTE	DISPUTED	FALSE	DTCCNovationRequest	
REFUSED	EE_ACCEPT	EE_ACCEPTED	FALSE	DTCCNovationRequest	
REFUSED	OVERDUE	OVERDUE	FALSE	DTCCNovationRequest	
REFUSED	REFUSE	REFUSED	FALSE	DTCCNovationRequest	
REFUSED	REQUEST	REQUESTED	FALSE	DTCCNovationRequest	
REFUSED	RP_CONSENT	RP_CONSENTED	FALSE	DTCCNovationRequest	
REQUESTED	CLEANUP	INACTIVE	FALSE	DTCCNovationRequest	
REQUESTED	COMPLETE	COMPLETED	FALSE	DTCCNovationRequest	
REQUESTED	DISPUTE	DISPUTED	FALSE	DTCCNovationRequest	
REQUESTED	EE_ACCEPT	EE_ACCEPTED	FALSE	DTCCNovationRequest	
REQUESTED	OVERDUE	OVERDUE	FALSE	DTCCNovationRequest	

Orig Status	Action	Resulting Status	Use STP	Subtype	Rules
REQUESTED	REFUSE	REFUSED	FALSE	DTCCNovationRequest	
REQUESTED	REQUEST	REQUESTED	FALSE	DTCCNovationRequest	
REQUESTED	RP_CONSENT	RP_CONSENTED	FALSE	DTCCNovationRequest	
RP_CONSENTED	CLEANUP	INACTIVE	FALSE	DTCCNovationRequest	
RP_CONSENTED	COMPLETE	COMPLETED	FALSE	DTCCNovationRequest	
RP_CONSENTED	DISPUTE	DISPUTED	FALSE	DTCCNovationRequest	
RP_CONSENTED	EE_ACCEPT	EE_ACCEPTED	FALSE	DTCCNovationRequest	
RP_CONSENTED	OVERDUE	OVERDUE	FALSE	DTCCNovationRequest	
RP_CONSENTED	REFUSE	REFUSED	FALSE	DTCCNovationRequest	
RP_CONSENTED	REQUEST	REQUESTED	FALSE	DTCCNovationRequest	
RP_CONSENTED	RP_CONSENT	RP_CONSENTED	FALSE	DTCCNovationRequest	
SENT	CLEANUP	INACTIVE	FALSE	DTCCNovationRequest	
SENT	COMPLETE	COMPLETED	FALSE	DTCCNovationRequest	
SENT	DISPUTE	DISPUTED	FALSE	DTCCNovationRequest	
SENT	EE_ACCEPT	EE_ACCEPTED	FALSE	DTCCNovationRequest	
SENT	ERROR	ERROR	FALSE	DTCCNovationRequest	
SENT	REFUSE	REFUSED	FALSE	DTCCNovationRequest	
SENT	REQUEST	REQUESTED	FALSE	DTCCNovationRequest	
SENT	RP_CONSENT	RP_CONSENTED	FALSE	DTCCNovationRequest	
VERIFIED	AMEND	VERIFIED	FALSE	DTCCNovationRequest	
VERIFIED	CANCEL	CANCELED	FALSE	DTCCNovationRequest	
VERIFIED	SEND	SENT	FALSE	DTCCNovationRequest	CleanupOutgoingNovationRequest



6.2.4 IncomingNovationRequest workflow

IncomingNovation message are created by the ImportMessage engine when messages are received from DTCC.

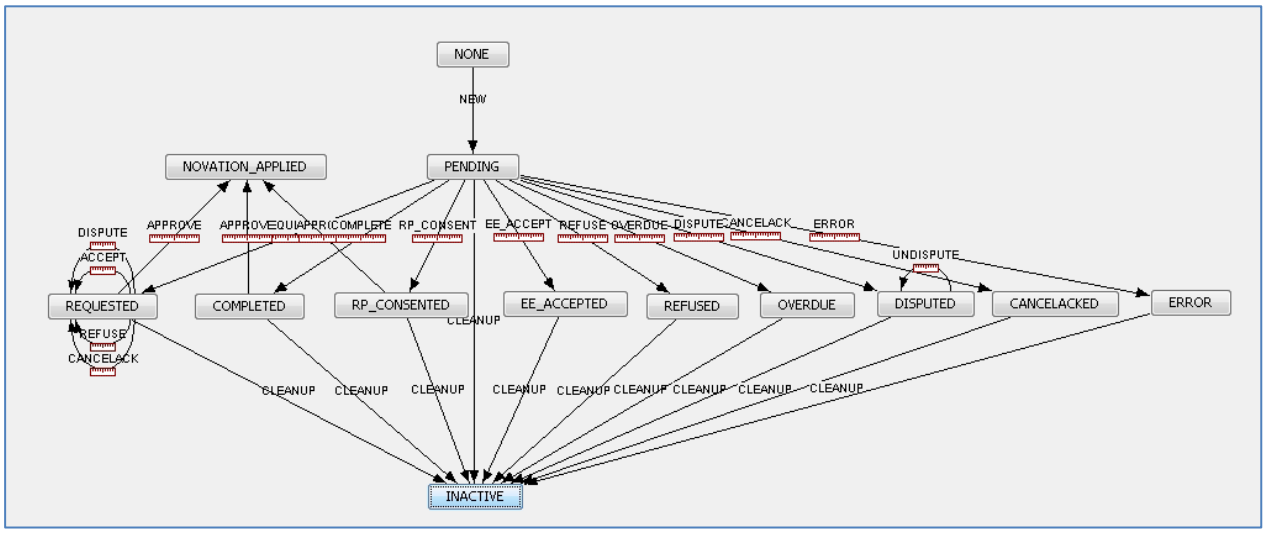
Based on its content, indexed message and/or trade, when exists, is modified.

For alleged message, user can act on the incoming message in order to create a DTCCNovationReply message.

Orig Status	Action	Resulting Status	Use STP	Subtype	Rules
CANCELACKED	CLEANUP	INACTIVE	FALSE	IncomingNovationRequest	
COMPLETED	APPROVE	NOVATION_APPLIED	FALSE	IncomingNovationRequest	DTCCApplyNovation
COMPLETED	CLEANUP	INACTIVE	FALSE	IncomingNovationRequest	
DISPUTED	CLEANUP	INACTIVE	FALSE	IncomingNovationRequest	

Orig Status	Action	Resulting Status	Use STP	Subtype	Rules
DISPUTED	UNDISPUTE	DISPUTED	FALSE	IncomingNovationRequest	DTCCNovationReply
EE_ACCEPTED	CLEANUP	INACTIVE	FALSE	IncomingNovationRequest	
ERROR	CLEANUP	INACTIVE	FALSE	IncomingNovationRequest	
NONE	NEW	PENDING	FALSE	IncomingNovationRequest	
OVERDUE	CLEANUP	INACTIVE	FALSE	IncomingNovationRequest	
PENDING	CANCELACK	CANCELACKED	FALSE	IncomingNovationRequest	CleanupIncomingNovationRequest IncomingNovationRequest
PENDING	CLEANUP	INACTIVE	FALSE	IncomingNovationRequest	
PENDING	COMPLETE	COMPLETED	FALSE	IncomingNovationRequest	CleanupIncomingNovationRequest IncomingNovationRequest
PENDING	DISPUTE	DISPUTED	FALSE	IncomingNovationRequest	CleanupIncomingNovationRequest IncomingNovationRequest
PENDING	EE_ACCEPT	EE_ACCEPTED	FALSE	IncomingNovationRequest	CleanupIncomingNovationRequest IncomingNovationRequest
PENDING	ERROR	ERROR	FALSE	IncomingNovationRequest	IncomingNovationRequest
PENDING	OVERDUE	OVERDUE	FALSE	IncomingNovationRequest	CleanupIncomingNovationRequest IncomingNovationRequest
PENDING	REFUSE	REFUSED	FALSE	IncomingNovationRequest	CleanupIncomingNovationRequest IncomingNovationRequest
PENDING	REQUEST	REQUESTED	FALSE	IncomingNovationRequest	CleanupIncomingNovationRequest IncomingNovationRequest
PENDING	RP_CONSENT	RP_CONSENTED	FALSE	IncomingNovationRequest	CleanupIncomingNovationRequest IncomingNovationRequest
REFUSED	CLEANUP	INACTIVE	FALSE	IncomingNovationRequest	
REQUESTED	ACCEPT	REQUESTED	FALSE	IncomingNovationRequest	DTCCNovationReply
REQUESTED	APPROVE	NOVATION_APPLIED	FALSE	IncomingNovationRequest	DTCCApplyNovation
REQUESTED	CANCELACK	REQUESTED	FALSE	IncomingNovationRequest	DTCCNovationReply
REQUESTED	CLEANUP	INACTIVE	FALSE	IncomingNovationRequest	

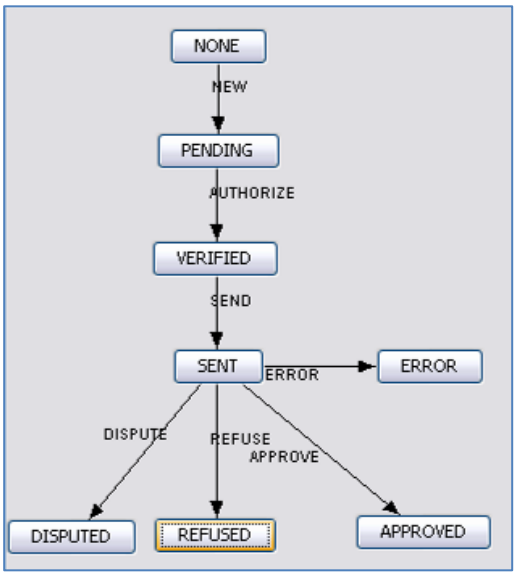
Orig Status	Action	Resulting Status	Use STP	Subtype	Rules
REQUESTED	DISPUTE	REQUESTED	FALSE	IncomingNovationRequest	DTCCNovationReply
REQUESTED	REFUSE	REQUESTED	FALSE	IncomingNovationRequest	DTCCNovationReply
RP_CONSENTED	APPROVE	NOVATION_APPLIED	FALSE	IncomingNovationRequest	DTCCApplyNovation
RP_CONSENTED	CLEANUP	INACTIVE	FALSE	IncomingNovationRequest	



Workflow rules used:

- *IncomingNovationRequest*: This rule sets attributes on the corresponding indexed message (if any), and changes the index message status according to incoming consent type and consent status.
- *DTCCApplyNovation:* This rule acts on the corresponding trade (if any). This rule uses the domains DTCCNCTradeAction.<consent type>.<consent status> and DTCCNCTradeAction.<consent type> in order to decide which action should be applied on the corresponding trade. This rule
    - Adds DTCCNovationConsentId as a keyword
    - If the consent type is Accept:
        - DTCCAutoGenerateAssignment is added as a trade keyword, based on corresponding message attribute value.
        - If trade is terminated, trade is just saved. If not terminated, then trade is terminated (novated) using message details (fee amount, notional, new counterparty, …)
- *DTCCNovationReply*: this rule creates the DTCCNovationReply message with consent type based on the action performed (Approve, Refuse or Dispute)
- *CleanupIncominNovationRequest*: applies the CLEANUP action on all existing DTCCIncomingNovationRequest messages for the same novation consent id, same sender and receiver. The goal here is to have only the last incoming message not being INACTIVE for ease of reporting and monitoring.

6.2.5 DTCCNovationReply workflow



6.2.6 AutoGenerateAssignment option

When sending a novation consent message to DTCC (DTCCNovationRequest or DTCCNovationReply), AutoGenerateAssignment option is mandatory.

When a novation consent is confirmed with AutoGenerateAssignment = true, then no confirmation message is required for this novation transaction, whereas AutoGenerateAssignment = false means that a novation confirmation message has to be send as usual.

Setting AutoGenerateAssignment option during the novation consent lifecycle

Default value of AutoGenerateAssignment in Calypso is *false*. You can set this value to *true* using message rule SetAttributes: (note that the transition comment is important, it is used by the rule SetAttributes in order to know which attribute and value to set).

For this, you can replace this workflow transition (in both DTCCNovationRequest and DTCCNovationReply workflows):

Orig Status	Action	Resulting Status	Rules	Comment
PENDING	AUTHORIZE	VERIFIED		

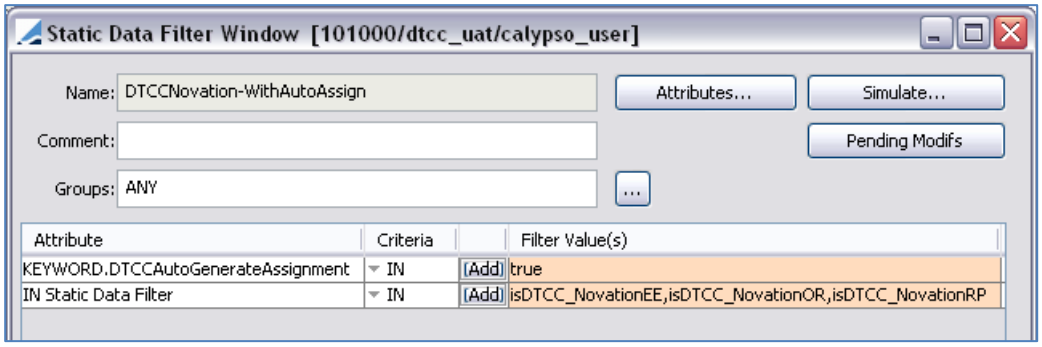
By:

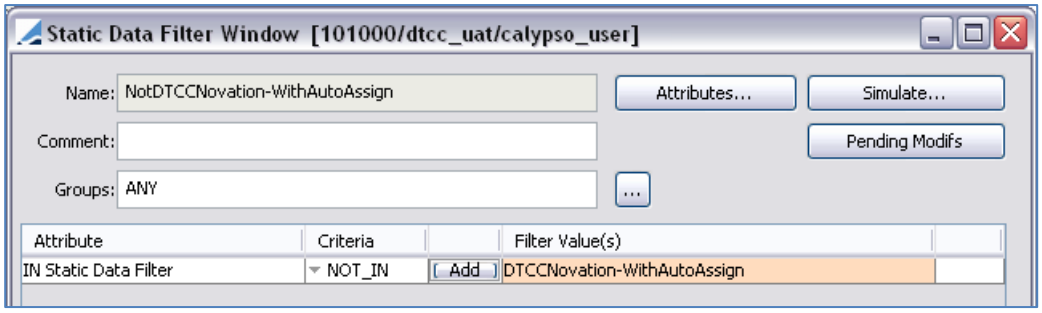
Orig Status	Action	Resulting Status	Rules	Comment
PENDING	AUTHORIZE	VERIFIED	SetAttributes	{AutoGenerateAssignment = true}

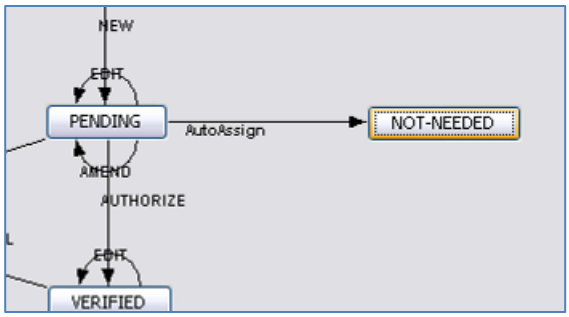
Using AutoGenerateAssignment option to manage confirmation messages

The DTCCApplyNovation message rule, when it updates the trade once the novation request is approved, will set the DTCCAutoGenerateAssignment keyword with the corresponding value.

In the DTCC_CONFIRM workflow, this keyword can be used to manage the lifecycle of the novation DTCC_CONFIRM. Example:



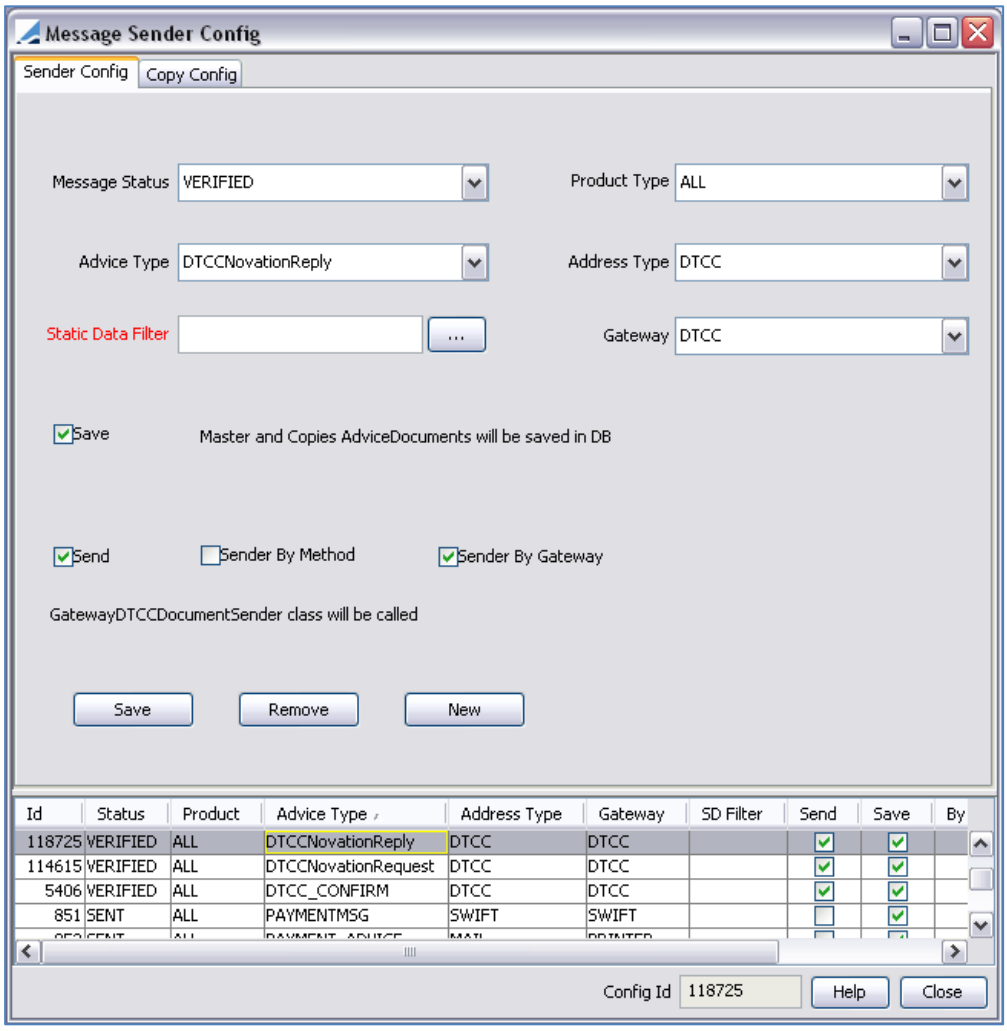




Orig Status	Action	Resulting Status	Rules	Filter
PENDING	AutoAssign	NOT-NEEDED		DTCCNovation-WithAutoAssign
PENDING	AUTHORIZE	VERIFIED		NotDTCCNovation-WithAutoAssign

6.2.7 Sender config

DTCCNovationRequest and DTCCNovationReply message are sent by sender engine when message is VERIFIED:



6.2.8 Reporting and monitoring

Many message statuses should be monitored in the message report or the Task station.

Among them, some of the most meaningful ones are:

Msg Type	Status	Meaning and action to take
All	ERROR	Message has been rejected by DTCC because of bad content. See error explanation in the message itself
All	REFUSED or DISPUTED	A pending novation request has been refused or disputed by remaining party or transferee. See consent comment in the message.
All	OVERDUE	A pending novation request is overdue
IncomingNovationrequest	RP_ALLEGED	PO is remaining party and is alleged a novation request. User should reply back with the APPROVE or REFUSE action, or take no action.
IncomingNovationrequest	EE_ALLEGED	PO is transferee and is alleged a novation request. User should reply back with the DISPUTE action, or take no action.
IncomingNovationrequest	APPROVED	On an APPROVED incoming novation request, user should accept manually (action APPROVE) in order to update the corresponding trade

Section 7. DTCC Message Keywords

A number of DTCC message templates are provided out-of-the-box under `resources/com/calypso/templates/dtcc`. They can be customized or you can create your own templates. DTCC Message templates should be registered in the "DTCC.Templates" domain.

Templates contain XML tags as well as template keywords to retrieve information from trades, messages and transfers. All available message keywords are described below. Keywords have the format `|keyword name|`, they are bracketed using the "|" symbol.

Keyword Names	CDS Master Conf	CDS Matrix Names	CDS Index CDS ABS Index	CDS Index CDS ABS Index Standard Terms	CDS Index Tranche	CDS option	CDX option	Paymt
ACTIVITY If message sub-action is "amend", then ACTIVITY is "Modify", otherwise it's "New".	x	x	x	x	x	x	x	
AMEND_EFFECTIVE_DATE Effective date.	x	x	x	x	x	x	x	
AMEND_FEE_ELEMENT Adds the element with the information based on the "AMENDMENT_FEE" if present.	x	x	x	x	x	x	x	
AMEND_TRADE_DATE Trade date.	x	x	x	x	x	x	x	
ATTACHMENT_POINT ATTACHMENT_POINT#ASSIGNED (assigned trade) CDSIndexTranche attachment point.					x			
CALC_AGENT_BUSINESS_CENTER CALC_AGENT_BUSINESS_CENTER#ASSIGNED (assigned trade) Only for CreditDefaultSwap, when transaction type is Matrix or is part of the AsiaPacific&Sovereign list, adds the tag with code of the calculation agent city of the trade which is stored in the comment of the related domain value.	x	x	x	x	x	x	x	
CALCULATION_AGENT CALCULATION_AGENT#ASSIGNED (assigned trade) DTCC ID of the SELLER.				x	x	x	x	
CDSINDEX_ADDITIONAL_TERM_ELEMENT CDSINDEX_ADDITIONAL_TERM_ELEMENT#ASSIGNED (assigned trade) Always empty. For Tranches, <additionnalTerms> is always omitted.					x			

Keyword Names	CDS Master Conf	CDS Matrix Names	CDS Index CDS ABS Index	CDS Index CDS ABS Index Standard Terms	CDS Index Tranche	CDS option	CDX option	Paymt
CDSINDEX_ADDITIONAL_TERM CDSINDEX_ADDITIONAL_TERM#ASSIGNED (assigned trade) Always "N"			x	x				
CDSINDEX_REF_ENTITY_ID_ELEMENT CDSINDEX_REF_ENTITY_ID_ELEMENT_N (assigned trade) RED sec code of the index definition.			x	x	x			
CDSINDEX_REF_ENTITY_RED_NAME_ELEMENT CDSINDEX_REF_ENTITY_RED_NAME_ELEMENT#ASSIGNED (assigned trade) Basket name of the index definition. Note that for CDS indices, index version is removed from the name. Ex: basket named CDX.NA.HY.4-V1 will become CDX.NA.HY.4			x	x	x			
COLLATERAL COLLATERAL#ASSIGNED (assigned trade) Tag populated with the information of the COLLATERAL fee: **Calypso supports only_PaymentPercent**.	x	x	x	x	x			
CONTRA_PARTY_ID DTCC ID of message receiver	x		x					
CONTRA_TRADE_ID Counterparty trade id (DTCC's Contra Trade Id)	x		x					
CREDIT_EVENTS_RESTRUCTURING CREDIT_EVENTS_RESTRUCTURING#ASSIGNED (assigned trade) Credit Event Restructuring.	x	x						
CURRENCY CURRENCY#ASSIGNED (assigned trade) Trade currency.	x	x	x	x	x			x
DESTINATION_SSI_ELEMENT Adds the tag with information of sender contact attribute "DESTINATION_SSI"								x
DK_REASON Value of message attribute "DK Reason".	x		x					
EFFECTIVE_DATE	x	x	x	x	x			x

Keyword Names	CDS Master Conf	CDS Matrix Names	CDS Index CDS ABS Index	CDS Index CDS ABS Index Standard Terms	CDS Index Tranche	CDS option	CDX option	Paymt
EFFECTIVE_DATE#ASSIGNED (assigned trade) For CreditDafeultSwap, trade start date For CDSIndex and CDSIndexTranche, start date of the underlying Index.								
EFFECTIVE_RATE Attribute "Reset Rate" of related BOTransfer, rounded to 5 decimal digits.								x
EXHAUSTION_POINT EXHAUSTION_POINT#ASSIGNED (assigned trade) CDSIndexTranche exhaustion point.					x			
FIRST_PAYMENT_DATE FIRST_PAYMENT_DATE#ASSIGNED (assigned trade) Date of first cashflow.					x			
FIXED_RATE_PAYER#ASSIGNED (assigned trade) See FIXED_RATE_PAYER applied to assigned trade.	x	x	x	x	x			
FIXED_RATE_PAYER DTCC ID of trade buyer.	x	x	x	x	x			
FIXED_RATE FIXED_RATE#ASSIGNED (assigned trade) For CDSIndex, fixed rate of the underlying Index. For CreditDefaultSwap and CDSIndexTranche, fixed rate of the trade.			x	x	x			
FLOATING_RATE_PAYER FLOATING_RATE_PAYER#ASSIGNED (assigned trade) DTCC ID of trade seller.	x	x	x	x	x			
FULL_FIRST_CALCULATION_PERIOD If trade keyword TerminationFullFirstCalculationPeriod = "false", then "NotApplicable". Otherwise "Applicable".	x	x	x	x	x			
INCREASE_EFFECTIVE_DATE See INCREASE_TRADE_DATE.	x	x	x	x	x			
INCREASE_FEE_AMOUNT Amount of the termination fee.	x	x	x	x	x			
INCREASE_FEE_CURRENCY Currency of the termination fee.	x	x	x	x	x			

Keyword Names	CDS Master Conf	CDS Matrix Names	CDS Index CDS ABS Index	CDS Index CDS ABS Index Standard Terms	CDS Index Tranche	CDS option	CDX option	Paymt
INCREASE_FEE_DATE Payment date of the termination fee.	x	x	x	x	x			
INCREASE_FEE_PAYER DTCC ID of termination fee payer.	x	x	x	x	x			
INCREASE_FEE_RECEIVER DTCC ID of termination fee receiver.	x	x	x	x	x			
INCREASE_NOTIONAL_AMT Amount of notional increased (Current notional - old notional).	x	x	x	x	x			
INCREASE_TRADE_DATE Termination trade date.	x	x	x	x	x	x	x	
INDEX_ANNEX_DATE_ELEMENT INDEX_ANNEX_DATE_ELEMENT#ASSIGNED (assigned trade) Tag populated with index annex date.					x			
INDEX_PUBLICATION_DATE_ELEMENT INDEX_PUBLICATION_DATE#ASSIGNED (assigned trade) • "<publicationDate>yyyy-mm-dd</publicationDate>" When trade is a CDSIndexTranche on a CDX index and trade date >= 2008-07-14 "yyyy-mm-dd" is the date defined as the greatest date listed in the "DTCCSettledEntityMatrixDates" domain that is on or before trade date. (This domain has to be maintained by users with the relevant publication dates.) ex: - if the domain is populated with: "2008-02-09", "2008-12-12", "2009-05-08" - then a trade with trade date - "2008-06-10" has no publication date - "2008-10-10" has a publication date = "2008-02-09" - "2008-12-12" has a publication date = "2008-12-12" • Element is empty otherwise					x			
INITIAL_PAYMENT INITIAL_PAYMENT#ASSIGNED (assigned trade) Tag populated with the information of the UPFRONT fee.			x	x	x			
IS_FULL_TERMINATION Adds the tag is termination type is "FullTermination".	x	x	x	x	x			
MASTER_AGREEMENT_DATE		x		x	x			

Keyword Names	CDS Master Conf	CDS Matrix Names	CDS Index CDS ABS Index	CDS Index CDS ABS Index Standard Terms	CDS Index Tranche	CDS option	CDX option	Paymt
MASTER_AGREEMENT_DATE#ASSIGNED (assigned trade) Date of legal agreement applying to this trade.								
MASTER_AGREEMENT_TYPE MASTER_AGREEMENT_TYPE#ASSIGNED (assigned trade) Type of legal agreement applying to this trade.		x		x	x			
MASTER_CONFIRMATION_ANNEX_DATE_ELEMENT MASTER_CONFIRMATION_ANNEX_DATE_ELEMENT#ASSIGNED (assigned trade) Tag populated with index annex date Business convention: this tag is populated for CDX 1 index only.			x					
MASTER_CONFIRMATION_DATE MASTER_CONFIRMATION_DATE#ASSIGNED (assigned trade) Date of the master confirmation applying to this trade.	x		x	x	x			
MASTER_CONFIRMATION_TYPE MASTER_CONFIRMATION_TYPE#ASSIGNED (assigned trade) Type of the master confirmation applying to this trade.	x		x	x	x	x	x	
MATRIX_CONTRACTUAL_SUPPLEMENT_ELEMENT MATRIX_CONTRACTUAL_SUPPLEMENT_ELEMENT#ASSIGNED (assigned trade) Adds the tag filled with "ISDA2003CreditMonolineInsurers2005" when Master confirmation type is NorthAmericanCorporate.		x						
MATRIX_PUBLICATION_DATE_ELEMENT MATRIX_PUBLICATION_DATE_ELEMENT#ASSIGNED (assigned trade) Not yet defined.		x						
MATRIX_SOURCE MATRIX_SOURCE#ASSIGNED (assigned trade) • "Publisher" for CDSIndexTranche on a CDX1 index when trade date >= 2008-07-14 • "NotApplicable" otherwise					x			
MATRIX_TERM		x						

[1] In order to distinguish Index families (CDX, ITRAXX, …), the DTCC uses the sec code « IndexType » with value « CDX » (to be added on each relevant index definition).

Keyword Names	CDS Master Conf	CDS Matrix Names	CDS Index CDS ABS Index	CDS Index CDS ABS Index Standard Terms	CDS Index Tranche	CDS option	CDX option	Paymt
MATRIX_TERM#ASSIGNED (assigned trade) See master confirmation type.								
MATRIX_TYPE MATRIX_TYPE#ASSIGNED (assigned trade) Always "CreditDerivativesPhysicalSettlementMatrix".		x						
MATURITY_DATE Attribute "ScheduledMatDate" of the transfer.								x
MESSAGE_ID Message id.	x	x	x	x	x	x	x	x
MESSAGE_TRANSFERID Transfer id.								x
MODIFIED_EQUITY_DELIVERY_ELEMENT MODIFIED_EQUITY_DELIVERY_ELEMENT#ASSIGNED (assigned trade) Adds the tag with value "Applicable". Business convention: this tag is added only for CDSIndextranches on CDX1 indexes.					x			
NOTIONAL NOTIONAL#ASSIGNED (assigned trade) Trade notional.	x	x	x	x	x	x	x	x
NOVATED_AMOUNT See NOTIONAL.	x	x	x	x	x	x	x	
NOVATION_DATE Novation effective date, see TERMINATION_EFFECTIVE_DATE.	x	x	x	x	x	x	x	
NOVATION_EE_PAYMENT_ELEMENT Tag populated with the information of the "FEE" fee.	x	x	x	x	x	x	x	
NOVATION_OR_PAYMENT_ELEMENT Tag populated with the information of the termination fee.	x	x	x	x	x	x	x	
NOVATION_RP_PAYMENT_ELEMENT Tag populated with the information of the termination fee.	x	x	x	x	x	x	x	
OBLIGATIONS Not yet defined.	x	x						

Keyword Names	CDS Master Conf	CDS Matrix Names	CDS Index CDS ABS Index	CDS Index CDS ABS Index Standard Terms	CDS Index Tranche	CDS option	CDX option	Paymt
OUTSTANDING_NOTIONAL_AMT In case of a notional increase / decrease, new notional.	x	x	x	x	x	x	x	
PARTIAL_ELEMENT In case of a partial termination, adds the tag, with information of decreased and outstanding notional.	x	x	x	x	x	x	x	
PAYMENT_AMOUNT Transfer settlement amount.								x
PAYMENT_CURRENCY Transfer settlement currency.								x
PAYMENT_DATE Transfer settle date.								x
PAYMENT_EE If trade has a single payment, adds the tag with relevant currency and amount Otherwise adds the periodicPayment tag without the optional tag.	x							
PAYMENT_END Attribute "EndDate" of BOTransfer.								x
PAYMENT_PAY_RECEIVE_STR If settlement amount of BOTRansfer is negative, then "PAY", otherwise "REC".								x
PAYMENT_REASON Transfer type.								x
PAYMENT_START Attribute "StartDate" of BOTransfer.								x
PAYMENT PAYMENT#ASSIGNED (assigned trade) If trade has a single payment, adds the tag with relevant currency and amount Otherwise adds the periodicPayment tag including the optional tag.	x	x						
RECEIVE_TIME Attribute "StartDate" of BOTransfer.	x	x	x	x	x	x	x	x
RECEIVER_CODE Counterparty DTCC Id.	x	x	x	x	x	x	x	x

Keyword Names	CDS Master Conf	CDS Matrix Names	CDS Index CDS ABS Index	CDS Index CDS ABS Index Standard Terms	CDS Index Tranche	CDS option	CDX option	Paymt
REF_ENTITY_ID_ELEMENT REF_ENTITY_ID_ELEMENT#ASSIGNED (assigned trade) Adds the tag populated with reference entity RED_PAIR when present.	x	x						
REF_ENTITY_NAME_ELEMENT REF_ENTITY_NAME_ELEMENT#ASSIGNED (assigned trade) Adds the with value: - If trade reference entity has a RED_PAIR attribute, then the RED_LEGAL_NAME attribute is retrieved - this case corresponds to standard case, when reference entity is known in MarkIt/RED - Otherwise it retrieves the DTCC_FREEFORM_NAME attribute - this attributes should be entered manually when reference entity is not known in RED/MarkIt - Otherwise it retrieves the RED_LEGAL_NAME attribute - this is done for clients who input legal names of reference entity in this field, whether it's in RED/MarkIt or not.	x	x						
REF_OB_ISIN REF_OB_ISIN#ASSIGNED (assigned trade) Get ISIN code for reference obligation. Business rule: If there is no reference obligation or if reference obligation has no ISIN code, dummy ISIN code "XSNOREFOBL00" is used. In case of EmergingEuropeanCorporateLPN trades, dummy ISIN code “XSUNKNOWN000” is used	x	x						
REFERENCE_ENTITY Attribute RefEntity of BOTransfer.								x
RELEASE_TIME Current date time.	x	x	x	x	x	x	x	x
SENDER_CODE DTCC ID of trade PO.	x	x	x	x	x	x	x	x
SOURCE_SSI_ELEMENT Adds the tag with information of sender contact attribute "SOURCE_SSI".								x
TERMINATION_DATE TERMINATION_DATE#ASSIGNED (assigned trade) Trade maturity date.	x	x	x	x	x	x	x	
TERMINATION_EFFECTIVE_DATE Termination effective date.	x	x	x	x	x	x	x	
TERMINATION_FEE_AMOUNT					x			

Keyword Names	CDS Master Conf	CDS Matrix Names	CDS Index CDS ABS Index	CDS Index CDS ABS Index Standard Terms	CDS Index Tranche	CDS option	CDX option	Paymt
Termination fee amount.								
TERMINATION_FEE_CURRENCY Termination fee currency.					x			
TERMINATION_FEE_DATE Termination fee date.					x			
TERMINATION_FEE_ELEMENT Tag populated with the information of the "TERMINATION FEE" fee.	x	x	x	x		x	x	
TERMINATION_FEE_PAYER DTCC ID of termination fee payer.					x			
TERMINATION_FEE_RECEIVER DTCC ID of termination fee receiver.					x			
TERMINATION_TRADE_DATE Termination trade date.	x	x	x	x	x	x	x	
TRADE_DATE TRADE_DATE#ASSIGNED (assigned trade) Trade date.	x	x	x	x	x	x	x	x
TRADE_ID_AMEND_SUPPLEMENT Deprecated, see TRADE_ID_SUPPLEMENT.	x	x	x	x	x	x	x	
TRADE_ID_SUPPLEMENT TradeRefNbrSupplement is populated with: - trade internal reference + "F" in case of a full termination - trade internal reference + TransferTo keyword + "P" in case of a partial termination - trade internal reference + TransferTo keyword + "N" in case of a novation - trade internal reference + TransferTo keyword + "I" in case of a notional increase - trade internal reference + "A" + message linkedId in case of an amendment (trade is confirmed already).	x	x	x	x	x	x	x	
TRADE_ID TRADE_ID#ASSIGNED (assigned trade) Trade internal reference.	x	x	x	x	x	x	x	x
TRADE_START_DATE See EFFECTIVE_DATE.	x	x	x	x	x	x	x	
TRADE_TYPE								x

Keyword Names	CDS Master Conf	CDS Matrix Names	CDS Index CDS ABS Index	CDS Index CDS ABS Index Standard Terms	CDS Index Tranche	CDS option	CDX option	Paymt
Product Type.								
TRANSFEREE#ASSIGNED DTCC Id of Novation_Transferee.	x	x	x	x	x			
TRANSFEROR#ASSIGNED DTCC Id of Novation_Transferor.	x	x	x	x	x			
XSITYPE If message sub action is AMEND then "ModifyTradeConfirmation", otherwise "RequestTradeConfirmation".	x	x	x	x	x	x	x	