

# Nasdaq Calypso IceLink Integration Guide

Version 5.3.0

Revision 32.0 April 2024 Approved



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

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

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

## **Document History**

Revision	Published	Summary of Changes
1.0	May 2013	Initial Version for 2.0.0 Release
2.0	May 2013	Update ICELink API Compatibility
3.0	May 2013	Tweak installation instructions
4.0	June 2013	Add Dealer-side allocation support details & keyword changes
5.0	July 2013	Add Client-side allocation support details
6.0	September 2013	Add Parent LE mapping details
7.0	October 2013	Add setup details for FCM allocation support, partial clearing for allocations support, and multi-engine support
8.0	October 2013	Add setup details for FCM bidirectional
9.0	November 2013	Add setup details for FCM bidirectional with allocations, as well as Dealer and Client bidirectional support
10.0	December 2013	Add setup details for Client allocation bidirectional support
11.0	January 2014	Update keyword details
12.0	March 2014	Update Data Uploader dependency
13.0	June 2014	Add ICELink Custom Fields details
14.0	July 2014	Update keyword details
15.0	September 2014	Update CDS Settlement Matrix Mapping
16.0	April 2015	Add setup details for ICELink engine for V-14





Revision	Published	Summary of Changes
17.0	July 2015	Add Performance and Stability Improvement details
18.0	July 2015	Add ICELink Connection
19.0	November 2015	Update compatible ICELink API version
20.0	June 2016	Update compatible ICELink API version
21.0	August 2016	Supported JRE Version and ExecuteSQL setup
22.0	February 2017	Update compatible ICELink API version
23.0	July 2017	Added Netting/Compression lifecycle and its supported keywords
24.0	October 2017	Update Bi-Directional FCM section
25.0	December 2017	Update Legal Entity
26.0	December 2017	Update ICELink API Compatibility and Keywords section with MIFID keywords.
27.0	February 2018	Update Netting/Compression lifecycle, Update icelink config file and Added Workflow setup [FCM]
28.0	March 2018	Update Data Uploader dependency
29.0	November 2018	Update Data Uploader dependency
30.0	February 2020	Updated for version 3.0.0.
31.0	February 2022	Updated for version 4.0.1 (compatibility with version 17)
32.0	January 2024	Updated for version 5.3.0 (compatibility with version 18)

# This document describes the setup requirements to use the IceLink interface with Calypso.



# Table of Contents

Introduct	tion		6
1.1	Calypso	ICELink Interface Features	6
	1.1.1	Supported	6
	1.1.2	Not Supported	7
	1.1.3	Known Issues / Limitations	8
	1.1.4	ICELink API: Known Issues / Limitations	8
Setup Ins	structions	5	9
2.1	ICELink	Persistent Messaging Setup	9
2.2	Installat	ion Instructions	9
	2.2.1	Calypso Components	9
	2.2.2	ICELink Distribution	10
	2.2.3	ICELink Connection	10
	2.2.4	Setup Config Data using Execute SQL	11
	2.2.5	Message Workflows	11
	2.2.6	Task Station	13
	2.2.7	Calypso Mapping Window Setup	13
Legal En	tity Mapp	ing	14
3.1	Legal Er	ntity Mapping	14
3.2	Parent L	egal Entity Mapping	15
3.3	Process	ingOrg-CounterParty Mapping	16
Book Ma	pniaa		17
4.1	Book Ma	apping (Dealer / IA)	17
	411	ICEI ink Custom Field	17
	4.1.2	Default Book	18
4.2	Book Ma	apping (FCM)	18
Product	Mapping		20
5.1	Sinale N	ame CDS Mapping	20
	E 1 1		
	5.1.1 5.1.2	Issuel Mapping	20
	0.1.Z	REIOD Midphilig	ו∠ 10
	5.1.5		21
5.2	CDS Ind	ex Mapping	22
Trade W	orkflow		24
6.1	ICELink	Platform: Clearing Workflow Overview	24



	6.2	Calypso Trade Creation Overview (Dealer / IA)	25
	6.3	Calypso Trade Creation Overview (FCM)	25
	6.4	Calypso Trade Workflow Setup	26
	6.5	Novation Requirements	26
		6.5.1       Message Rule         6.5.2       ClearingKeywords Domain         6.5.3       Fees	26 26 27
	6.6	Allocation Requirements	27
		<ul> <li>6.6.1 Allocation Support (Dealer)</li> <li>6.6.2 Allocation Support (IA)</li> <li>6.6.3 Allocation Support (FCM)</li> <li>6.6.4 Netting/Compression</li> </ul>	27 28 32 33
	6.7	Bidirectional Requirements	35
		<ul> <li>6.7.1 Bidirectional Overview</li></ul>	35 35 36 36 37
	6.8	Workflow Keywords	39
ICEL	.ink Er	ngine Setup	41
	7.1	Configure the Engine	41
	7.2	Setup the IceLink config file	42
	7.3	Setup the gatewayservice config file	42
	7.4	Launching the ICELink Engine	42
	7.5	Troubleshooting	43
	7.6	Running Multiple Engines	43
Perf	orman	ce and Stability Improvement	44
Key	words		46
	9.1	General Keywords	46
	9.2	MIFID Keywords	49
		,	



# Introduction

This document describes the Calypso ICELink Interface setup. The ICELink interface is used to communicate with the ICELink platform for clearing trades through supported CCPs.

The interface builds upon the Data Uploader framework (Data Uploader module) and Calypso's Engine interface. It allows a Dealer, IA, or FCM to receive messages for trades booked through the ICELink web interface using ICELink's clearing workflow.

The ICELink engine connects to ICELink using its API listener interface and subscribes for messages. Once an ICELink message is received, the message flows through the configured Calypso workflows that route the message through the appropriate stages until completion, and a Calypso trade is created.

The interface also has support for bidirectional functionality, which allows a Dealer, IA, or FCM to send outgoing messages to ICELink from within Calypso. This is explained in more detail in the appropriate section below.

This document describes the configuration required to setup the workflows, etc. for the ICELink interface to run successfully. Please note that setup steps specific to the Dealer, IA, or FCM use-case are in their own separate sub-sections.

# 1.1 Calypso ICELink Interface Features

### 1.1.1 Supported

The ICELink interface supports trades cleared through the following CCPs:

- ICE Clear Credit LLC (US)
- ICE Clear Europe (UK)

Other CCPs may work if they are compatible with ICELink processes but have not been tested.

The ICELink interface supports the following trade types:

- Single Name CDS (Corp/Sov/Muni)
- CDS Index

The ICELink interface supports trade messages for the following workflows:

• Client Clearing (incl Block & Partial Clearing for Allocations)

The ICELink interface supports the following ICELink life cycle events for the Clearing workflow (i.e. these message types can be consumed by Calypso):

• ALLEGE: As the Dealer, IA, and FCM move the trade through the clearing workflow, ICELink will send ALLEGE messages through the API interface. A Bilateral trade will be created in Calypso to represent



the trade between the two parties (Dealer vs. IA), with the Calypso user's party as the PO. As each party affirms, keywords will be updated to track its progress in the clearing workflow.

- AFFIRM: When the CCP affirms the trade for clearing the original Bilateral trade will be Novated to the CCP as counterparty. The Bilateral will be terminated, and a new Cleared trade created.
- REJECT: If any party (IA, FCM, CCP) reject the trade, the Bilateral trade will be cancelled.
- RECALL: If the Dealer recalls the trade, the Bilateral trade will be cancelled. A RECALL action can only be applied by the Dealer who initiated the trade, and only before the client's FCM has approved.
- NETTING/COMPRESSION: At the end of the day/week Clearing house send the net or compressed offset values of all the booked deals between the parties. These offsets are received only for cleared deals via BulkTransactionListener.batchBrokeredTerminationChanged(). For this action, trade(s) in Calypso will be terminated and new trade(s) will be created for each offset value.
- PORTFOLIO TRANSFER [Between client]: In case of Portfolio Transfer, FCM can transfer the cleared trade from one client sub-account to another client sub-account. At the time Portfolio Transfer user can decide the amount and upfront fee to be transferred. These offsets are received only for cleared deals via BulkTransactionListener.batchBrokeredTerminationChanged(). For this action, trade(s) in Calypso will be terminated and new trade(s) will be created for each offset value.

The ICELink interface supports the following bidirectional Calypso actions for the Clearing workflow (i.e. these actions can be executed from within Calypso and sent to ICELink):

- Dealer: A Dealer may action an Allege or Recall from within Calypso.
- IA: An IA may action an Affirm or Reject from within Calypso. This includes the ability to allocate a trade from within Calypso and send those allocation details to ICELink, similar to the ICELink Web GUI functionality.
- FCM: An FCM may action an Affirm or Reject from within Calypso.

### 1.1.2 Not Supported

The ICELink interface does **NOT** support the following trade types:

• Index Tranche

The ICELink interface does **NOT** support the following workflows:

- Bilateral
- Termination
- Transfer/Novation
- Backload
- SEF (certain features may not be fully implemented)



### 1.1.3 Known Issues / Limitations

- ICELink and Calypso have some common 3rd party packages that are in conflict. Please ensure that you follow the ICELink installation instructions exactly as described in the sections below to ensure that the Calypso environment works as expected.
- All Pending ICELink Engine messages **must** be fully processed before upgrading to a later ICELink API version.
- There is currently a known issue with rounding of Fees in Calypso when performing allocations, which leaves a small Fee showing on the block trade. Please contact Help Desk support for a fix if you experience this issue.
- Currently ICELink life cycle events MISCLEAR & ENRICH are not supported in all use-cases.

### 1.1.4 ICELink API: Known Issues / Limitations

The ICELink API has the following limitations and hence not supported by Calypso:

• The API does not support receiving messages for Credit Events.



# Setup Instructions

# 2.1 ICELink Persistent Messaging Setup

The ICELink platform has a feature called 'Persistent Messaging'. When this is switched on for your ICELink login id, any messages to the id will be queued on their server when the interface is not connected and will be delivered upon reconnection. Otherwise, any messages resulting from trade activity while the ICELink Engine is down will be lost.

# (1) Mandatory. This feature *must* be turned on for any login ids which will be used for connecting to the ICELink platform through the ICELink Engine.

Please contact ICELink Support to activate this feature.

## 2.2 Installation Instructions

## 2.2.1 Calypso Components

The use of the ICELink interface requires the Calypso Data Uploader.

ICELink is installed as part of the Calypso Installer when you select the Icelink Interface and the Data Uploader optional module.





Optional Modules (Please consult with your Calypso Account Executive for licensing questions)
 Calypso Business Services Layer (CBSL)
 CCP eClearing Web Portal
 CMF OTC Clearing
 Collateral
 Colla

All subsequent instructions assume that all Data Uploader installation steps have been completed successfully. This includes:

- Setting up the GATEWAYMSG & UPLOADESOURCEMSG workflows
- Setting up Task Station tabs for the workflows above
- If you intend to use the bidirectional features of the ICELink interface, please be sure to check the 'bidirectional' category when applying GatewaySchemaData, and setup the PLATFORMMSG workflow along with appropriate Task Station tabs.

Please refer to the Data Uploader Guide for specific installation and configuration information. You **must** configure the Data Uploader prior to configuring ICELink.

## 2.2.2 ICELink Distribution

Important: Due to conflicts between some 3<sup>rd</sup>-party jars used by ICELink and those used by Calypso, it is very important that the next steps are followed exactly as detailed. Not doing so could prevent the Calypso Data Server from starting correctly.

Download the ICELink distribution zip (v 8.6.3.0 or higher) and uncompress it to a temporary directory. [Details on where to get the ICELink distribution from should be provided to you by ICELink; please do not contact Calypso support for this.]

Take the following jars from the ICELink distribution and place them directly into the <calypso home>\jars directory:

- fpml-jaxb2\icelink-fpml-api.jar
- lib\bouncycastle-xxx.jar
- lib\commons-httpclient-xxx.jar
- lib\jackson-\*.jar
- lib\ICELinkAPI.jar

Finally, ensure that all jars listed above are included in the CLASSPATH.

### 2.2.3 ICELink Connection

The ICELink interface can connect to platform via two ways.

• <u>Direct Internet Connection</u>: A direct internet connection can be established by providing the valid host, username and password in icelink.properties. ICELink interface will establish a direct connection with the platform.



• <u>Internet Connection via a Proxy</u>: For a connection via proxy, apart from the host, username and password in icelink.properties, Proxy details also need to be provided. Proxy information must be provided in calypso environment property file against:

HTTP\_PROXY\_HOST = Proxy server host address

HTTP\_PROXY\_PORT = Port number, its optional field by default port is 80

HTTP\_PROXY\_USER = Proxy UserName, its optional depending on user proxy set-up

HTTP\_PROXY\_PASSWORD = Proxy Password, its optional depending on user proxy set-up

If HTTP\_PROXY\_HOST is present, then connection will be made via Proxy Server else direct internet connection will be used.

Note: The connection API provided by ICELink first tries to connect via proxy using proxy details and if it fails then as a fall back it will try to connect directly via internet.

### 2.2.4 Setup Config Data using Execute SQL

When you run Execute SQL, the following files should already be loaded:

- SchemaBase.xml
- GatewaySchemaBase.xml
- ICELinkSchemaData.xml
- FpMLSchemaData.xml

### 2.2.5 Message Workflows

The ICELink interface uses the UPLOADSOURCEMSG and GATEWAYMSG workflows when importing messages, and the PLATFORMMSG workflow for outgoing messages (i.e. bidirectional). These should have been setup as part of the Data Uploader Setup Guide.

Messages from the UPLOADSOURCEMSG workflow are translated from the external message format into Calypso's internal format and placed in the GATEWAYMSG workflow. The GATEWAYMSG workflow then translates the internal format, performs verifications, and saves the trade to the database.

For outgoing messages (ie. bidirectional mode), the PLATFORMMSG workflow converts the Calypso trade into the external message format and tracks sending it to the external system as well as message acceptance.



#### Message Workflow Screenshots

The UPLOADSOURCEMSG workflow:



#### The GATEWAYMSG workflow:





#### The PLATFORMMSG workflow:



### 2.2.6 Task Station

The ICELink interface uses the Data Uploader Framework to create task station entries for all the messages and exceptions that are encountered. The user can view / reprocess the messages that failed in validation from the task station.

Please see the Data Uploader Setup Guide for how to add the appropriate messages and exceptions to the Task Station.

### 2.2.7 Calypso Mapping Window Setup

The Calypso Mapping Window is required to map the ICELink values used in the incoming messages to the Calypso values. Some mappings are provided for you out-of-the-box, such as Frequency and Seniority.

# NOTE: These mappings will also be used – in the opposite direction – for outgoing messages in bidirectional mode.

#### Setup Calypso Mapping Window

You can bring up the Calypso Mapping window using menu action "mapping.CalypsoMappingWindow".

💋 Calypso Mapping Window (Use	r: calypso_user)	
Interface Mappings	Name:	ICELink/Frequency
CME ICELink Eees	Interface Value:	3
Frequency	Calypso Value:	QTR
	Reverse Default:	
G □	<< Add	
	>> Remove	
	Configure Interfaces	
E SalesPerson E Seniority	Configure Types	
SNR SNRFOR SUB		
Traders		



# Legal Entity Mapping

# 3.1 Legal Entity Mapping

When a trade is alleged in the ICELink Web GUI, the Dealer is required to select a Legal Entity for both the Party and Counterparty; those ICELink entity names are then included in the ICELink message. The same is true for the FCM entity name (which the IA must always select when affirming, and a Dealer may select if they are not self-clearing). In addition, when you select a CCP, the underlying legal entity name for that CCP is included in the ICELink message as well.

The Legal Entity for the Party, Counterparty, FCM, and CCP for incoming trade messages will be identified in Calypso using the Legal Entity Attribute 'ICELinkParticipant', and the value would match the legal entity value provided in the message (which will match the name picked in the ICELink web screen).

The fetching logic will first search for a Legal Entity with the ICELinkParticipant attribute value specified in the message, and if not found then it will search for a Legal Entity having a matching Short Name (case-sensitive or all uppercase).

🗾 Legal Entity- Version - 2 [130100/icelink_rel2/calypso_user] (User: calypso_user)							
Utilities Help							
Short Name	DEALER			Status	Enabled	•	
Full Name	Dealer for ICELink P	atform		Role(s)	CounterPart	У	
Parent					ProcessingO	rg	
Country	NONE						
	r			1	I		
📈 Legal Entity /	Attributes Window	(User: calypso_u	ser)			_ 🗆 ×	
Legal Ent	ity DEALER		F	tole ALL		<b>v</b>	
Processing Or	g ALL	•	[				
Attribute Ty				alue			
Id	Processing Org	Legal Entity	Role	Attribut	е Туре	Attribute Value	
55696 A	LL	DEALER	ALL	ICELinkBook		DEALER_DEFBOOK	
54704 A	LL	DEALER	ALL	ICELinkPartic	ipant 🛛	cyp_dl1	

This lookup logic will be applied to PO, Counterparty, FCM, and CCP lookups. If no Calypso Legal Entity is found using the rules above, an error will be raised.

#### Note: The legal entity code for ICE Clear Credit LLC (US) is 'icetrust', and the code for ICE Clear Europe (UK) is 'iceclear'.



## 3.2 Parent Legal Entity Mapping

In some cases, there might be a need to map several separate ICELink Legal Entities to a single parent LE in Calypso.

To support this, first follow the procedure described above to setup separate Calypso Legal Entities each mapped to their corresponding ICELink Legal Entity. Once that is complete, simply add the 'ICELinkParent' attribute on each of the Calypso Legal Entities and populate it with the Calypso LE Short Name that you wish to be used as the parent.

When the lookup logic encounters a Legal Entity that has a parent defined, it will use the parent LE instead for all LE-related logic afterwards.

🗾 Legal Entity- V	ersion/	- 2 [130100/	icelink_rel2/calyp	oso_user](L	Jser: calyp	so_user)		
Utilities Help								
Short Name	ICE US				Status	Enabled		•
Full Name	Interco	ntinental Excha	ange US	_	Role(s)	CounterParty		=
Parent	<u> </u>				1			
Country		STATES	•	1	_			
To a block All France	<u> </u>		 					
Legal Entity A	ttribut	es Window (	User: calypso_us	er)				
	. F.							T
Legal Enti	ity p	CE US		Ro	Ie ALL		<u> </u>	1
Processing Org	a  4	ALL	•					
Attribute Typ	be 🖡	ACCOUNTING	-	Valu	Je l			
Id	Proce	ssing Org	Legal Entity	Role	Attri	oute Type	Attribute Va	lue
54698 Al	LL		ICE US	ALL	ICELinkPar	rent	ICE	
54697 Al	LL		ICE US	ALL	ICELinkPar	rticipant	icetrust	
🔀 Legal Entity- V	ersion/	- 2 [130100/	'icelink_rel2/calyp	oso_user](L	Jser: calyp	so_user)		
Utilities Help								
Short Name	ICE				Status	Enabled		-
Full Name	Interco	ntinental Excha	ange		Role(s)	CounterParty		
Parent						MarketPlace		
Country	UNITED	STATES	•	] [				
Teaching Ac From								





## 3.3 ProcessingOrg-CounterParty Mapping

Calypso uses the Legal Entity Attribute window to determine the counterparty of a trade based on the processing organization. The Calypso Mapping window is not used.

Configuration and maintenance at the level of the Legal Entity is more efficient and provides more flexibility for entry and reporting.

For example, if the Processing Organization is 'LONDON' then counterparty is 'GIGA1'. If Processing Org is 'NEWYORK' then counterparty will be 'GIGA2':

🗾 Legal Entity Attributes Window - Version - O								
Q- Search								
Legal Entity GIGA1	Ð	Role ALL	Processing Org	LONDON				
Attribute Group	🝸 🔁 Attrib	ute Type ICELinkPa 💽	<b>) Ə</b> Value	cyp_dl1 🛛 🕀				
Id Processing Org	Legal Entity Role	Attribute Group	Attribute Type	Attribute Value				
524503 LONDON GI	GA1 ALL		ICELinkParticipant	cyp_dl1				
🖊 Legal Entity Attributes Window - Version - O								
Q- Search								
Legal Entity GIGA2		Role ALL -	Processing Org	NEWYORK -				

Attribute Group		<b>▼</b> ∋	Attribute Type	ELinkPa 🚽 ∋	Value <mark>cyp</mark>	o_dli ව
Id	Processing Org	Legal Entity	Role	Attribute Gro 🛆	Attribute Type	Attribute Value
524504	NEWYORK	GIGA2	ALL		ICELinkParticipant	cyp_dl1



# **Book Mapping**

Once the ICELink interface determines the appropriate Legal Entity to use as the PO, it must then choose a Calypso Book for the trade. The sections below outline the logic used for the Dealer / IA and FCM use cases.

If no Calypso Book is found using the rules below, an error will be raised.

(1) Note: When a Calypso Book is determined based on the lookup solutions described below, the interface will verify that the Book's Legal Entity matches the Calypso Legal Entity found using the rules specified in the previous section. This is especially important to note when using a parent LE; the Book must belong to the parent LE, not the child.

# 4.1 Book Mapping (Dealer / IA)

When a trade is booked in the ICELink Web GUI, the user is NOT required to set any Book or Account value. However, in Calypso we require the trade to be assigned to a Calypso Book for the PO. In addition, we realize that Calypso clients might want to assign trades to different books. These requirements have been addressed with two different solutions that can be used interchangeably.

Note: In the use case where a trade is booked in ICELink by another user with you as the counterparty, the first ALLEGE message received by Calypso will have no custom field defined, hence the default book rule below \*must\* be used to create the trade. When you then affirm the trade, please ensure that you choose a book in the 'Book' Custom Field within the ICELink Web GUI if you wish to use a different book from the default.

## 4.1.1 ICELink Custom Field

ICELink provides users with the ability to create custom fields ('Client Specific Fields') with a corresponding pick list of valid values; this feature will be used to support mapping to different Calypso Books. Users will have to create a Custom Field named 'Book' within their ICELink login if they wish to use this feature. [See below for defaulting rules if such a field is not setup.]

On the Calypso side, the Custom Field 'Book' value will be identified in Calypso using the Book Attribute 'ICELinkBook'. If no Book is found in Calypso with that attribute value, the fetching logic will look for a Book with the short name matching the Custom Field 'Book' value (case-sensitive).



Book Window - Version -0 [130100/icelink_rel2/calypso_user] (User: calypso_user)								
View Help								
Book Id 54707	Attributes							
Name DEALER_BOOK1	Name	Value						
	Drawn MM Book	<b>▼</b> ▲						
Activity	FEE_RECOGNITION_LAG							
	ICELink Book							
Accounting Link AFS	ICELinkBook	Book 1						
Logal Eptity DEALED	ICE_BOOK							
	MARKITWIRE_PARTY_ID							

### 4.1.2 Default Book

For use cases where a Calypso client wishes to use a default book for all incoming trades from a particular Legal Entity, that value can be set using the Legal Entity Attribute 'ICELinkBook'. The ICELink interface will use that attribute value – from the Legal Entity determined using the lookup rules described in the previous section – to perform the book lookups described in the sub-section above (again, first by Book attribute and then by Book short name).

🕖 Legal Entity	Legal Entity Attributes Window (User: calypso_user)						
Legal E	ntity	DEALER		F	Role ALL	<b>~</b>	
Processing C	Drg	ALL	•				
Attribute T	уре	ACCOUNTING	•	Va	alue		
Id	Proc	essing Org	Legal Entity	Role	Attribute Type	Attribute Value	
55696	ALL		DEALER	ALL	ICELinkBook	DEALER_DEFBOOK	
54704	ALL		DEALER	ALL	ICELinkParticipant	cyp_dl1	

To use this method, do **not** specify a value in the Custom Field 'Book' within the ICELink Web GUI. This also means that if the value is mistakenly omitted when booking or affirming the trade within the ICELink Web GUI, the Default Book logic will kick in as well.

# (i) Note: If the Custom Field 'Book' contains a value, but it cannot be matched to a valid Calypso Book, an error will be thrown. The interface will not fall back to the Default Book in this case.

## 4.2 Book Mapping (FCM)

When the ICELink interface is running in FCM mode, the Calypso Book is determined using the Legal Entity Attribute 'Client Clearing Book' on the Legal Entity retrieved using the lookup rules described in the previous section.



The value in the 'Client Clearing Book' attribute will be matched to a Calypso Book using the Book Attribute 'ICELinkBook'. If no Book is found in Calypso with that attribute value, the fetching logic will look for a Book with the short name matching the 'Client Clearing Book' attribute value (case-sensitive).

🔀 Legal Entity- ¥	'ersion - 1 [13010	0/icelink_rel2/ca	lypso_us	er] (Use	er: calyp		_	
Utilities Help								
Short Name	CMF_1				Status	Enabled	-	-
Full Name	Clearing Member Fir	m 1			Role(s)	ProcessingOr	]	_
Parent								
Country	UNITED STATES		<b>•</b>	<u> </u>				
Inactive As From		User calypso_u	iser					
🗾 Legal Entity A	ttributes Window	(User: calypso_	user)				_	
Legal Entit	y CMF_1		- ]	Role	ALL		<b>*</b>	
Processing Org	ALL	•	]					
Attribute Typ		;	- 1	Value				
	- <u> </u>							
Id	Processing Org	Legal Entity	Role		Attribu	ite Type	Attribute Value	•
63201 AL	L	CMF_1	ALL	Clier	nt Clearir	ng Book	CMF1_DEFBOOK	
Book Window	- Version -1 [130	100/icelink_rel2	/calypso	user]	(User:	calypso_use	er)	
View Help								
Boo	k Td 59198		 Attrib	utes	[			
							1	
N	ame CMF1_DEFBO	ОК	Drawn	n MM Boo	Name ok		Value	
Act	ivity		FEE_F	RECOGN	ITION_L/	AG		
Accoupting			ICELir	nk Book	DADTY	TD		_
Accounting			Marke	t Index	FARII_	10		
Legal Er	ntity CMF_1		··· ORIG	IN				
Loca	tion America/New	Vork	POSI	TON_AC	COUNT	ID		



# **Product Mapping**

When importing product-based trade types into Calypso, mappings must be setup so that the ICELink interface can match the incoming trade details to objects contained within the client's instance of Calypso. The sections below outline the various mappings required for the different trade types supported by the ICELink interface.

## 5.1 Single Name CDS Mapping

When importing Single Name CDS trades, the ICELink interface needs a way to map the incoming CDS Issuer & RefOb key with the corresponding Issuer and RefOb in Calypso.

### 5.1.1 Issuer Mapping

For mapping the Issuer on the incoming trade to a Calypso Legal Entity, the ICELink interface makes use of the issuer RED code provided by ICELink on the incoming message.

The interface first attempts to find a Calypso Legal Entity with a matching Short Name (case-sensitive or all uppercase). If no matching Legal Entity is found it attempts to find a Legal Entity using the Legal Entity Attribute 'RED\_PAIR', and then finally using the Legal Entity Attribute 'RED\_TICKER'.

If no Calypso Legal Entity is found using the rules above, an error will be raised.

# Note: If the Calypso MarkIT Interface is used to import Issuers, the 'RED\_PAIR' and 'RED\_TICKER' attributes will automatically be set for you on the Legal Entity.

	Legal Entity	Attrib	utes Window (	(User: calypso_us	ser)			
Legal Entity SAMPLE ISSUER				R		Role	ALL	-
	Processing C	Drg	ALL	-				
	Attribute T	уре	ACCOUNTING	-		Value		
	Id	Pro	cessing Org	Legal Entity		Role	Official Type	Attribute Value
	57729	ALL		SAMPLE ISSUER		ALL <	RED_TICKER	897ED6
	60697	ALL		SAMPLE ISSUER		ALL	RED_REGION	maner
	60698	ALL		SAMPLE ISSUER		ALL	RED_TYPE	Corp
	Load Show F	ending a	Delete	Save			Authorization	Close



## 5.1.2 RefOb Mapping

For mapping the RefOb on the incoming trade, the ICELink interface makes use of the 'ISIN' Product Code. On the incoming trade message, the ISIN of the CDS underlying is provided by ICELink, and the matching product is found in Calypso by matching that value to the 'ISIN' code on a Calypso product.

Note: RefOb is not a required field when creating a CDS trade in Calypso. Therefore, if the ICELink interface detects that there is no Calypso product setup for the 'ISIN' code provided by ICELink, it will create the trade successfully without the RefOb.

Bond Window (User: calypso_user)	
<u>File H</u> elp	
📑 🚔   🖬 📓   🎽   🗙   🚔   🥹	
Name SNAC Test	Generic 💌
Security Code ISIN	Load
Convertible Call Schedule Brady Schedule Bond Coupon Market	Credit Events AB5 CLN Special CashFlows
Bond Class Bond 💌 Bon	d Type UST
Issue Date         Dated Date         Maturity Date           03/16/2009         03/15/2009         03/15/2012         3Y	Issuer SAMPLE ISSUER Dummy entity for Marki
	Country UNITED STATES
Issue Price Issue Yield Currency Redem. 99.667005 0 USD	Price Redem. Curr. Total Issued
C& e ISIN US912810FF04	Codes

### 5.1.3 CDS Settlement Matrix Mapping

For mapping the CDS Settlement Matrix on the incoming trade, the ICELink interface makes use of the 'ISDA Transaction Type' field on the incoming trade message from ICELink.

A user can map this ICELink 'ISDA Transaction Type' with an existing Calypso CDS Settlement Matrix using the Calypso Mapping window, as shown below.





For an incoming trade, the ICELink interface checks for the CDS Settlement Matrix mapping. If the mapping for the given 'ISDA Transaction Type' is present, then a CDS trade will be booked using that settlement matrix.

If no mapping is present, then a default settlement matrix will be set on the CDS trade. Please review the Calypso Documentation on how to setup default CDS Settlement Matrix mappings for issuers within Calypso. The ICELink interface uses the standard logic for assigning a default CDS Settlement Matrix to incoming trades.

Note: This mapping is required for ICELink trades since ICELink does not provide the frequency or holidays for Single Name CDS, so the ICELink interface must take these from the Calypso CDS Settlement Matrix.

## 5.2 CDS Index Mapping

When importing CDS Index trades, the ICELink interface needs a way to map the incoming CDS Index key with the corresponding CDS Index Definition in Calypso. To accomplish this, it uses the Product Code in Calypso.

The ICELink interface is able to map the incoming index to the Calypso index using the 'RED' code. Please ensure that this code is setup as a valid Product Code type for CDSIndex using the Product Code window under Configuration > Product > Code (screenshot below).



Product Code Window (Use	r: calypso_user)				_ []
Name	<b>•</b>	Туре	string		•
🗖 Unique	🗌 Search	able		Mandatory	
Product CDSIndex					
Name	Туре	Unique	Searchable	Mandatory	
BB_UNIQUE	string		<u> </u>		Bond,Equity
BB_IS_SUBORDINATED	string		<b>N</b>		Bond
RED	string				CDSIndex
RED_PAIR	string				Bond
BB_IS_SECURED	string				Bond
BB_SECURITY_TYP	string		<b>v</b>		Bond
BB_TICKER	string		<b>v</b>		Bond
CUSIP	string	•	<b>v</b>		ALL
Common	string	~	<b>v</b>		ALL
DebtSeniority	string				ALL
GCFCusip	string	<b>V</b>	<b>V</b>		ALL
ISIN	string		V		ALL
Loçal	strina			Π	ALL 🚬
<b>▲</b>					► I

Then for each index you intend to receive messages for, you will need to populate the 'RED' Product Code value. Below is a screenshot of the code being set on a fictitious index called 'ICE Index EM 14V1-5Y'.

JID:12800 Desc:ICE	Index EM 14¥1-5Y.Dec.2015 (User: calypso_user)	<u>_   ×</u>
Definition Reference Po	ortfolio CashFlows Next Version	
Issuer	GOVT. OF USA SecCode: Codes	
Name	ICE Index EM 14V1-5Y RED 2165BZAN2	$\supset$
Description	Series Version	
Reference Portfolio	TestBasket 🔽 New	
Notional	25,000,000 Current Factor 1.00000 Quote Type Spread	•
Start Date	09/20/2010 Maturity Date 12/20/2015 5Y 💌 Annex Date	
	USD Fixed 0.000000 bp	
Pmt QTR	FOLLOWING R Day 30/360 NYC	
Credit Event	BANKRUPTCY, FAILURE TO PAY, RESTRUCTURING	
Settlement	CASH   NO_ACCRUAL  Status  ENABLED	
Settlement Lag	Image: Bus         Settle Fee         Image: Bus         Image: Maturity Date Inclusive         Default Event         30         C           Offset         Settle Lag         Settle Lag <th>al</th>	al

Once the code type & value is identified, the interface will search for a corresponding index definition with the matching code type/value combination and the same Maturity Date as the trade. If found, the interface will create the trade using that index as the underlying. Otherwise, it will create an error message in the Task Station clearly identifying the code type, code value, and maturity date it tried to lookup.





# Trade Workflow

This section describes the ICELink interface trade workflow. It is important to understand these details so that the Calypso Trade Workflow can be customized accordingly. Please read & follow all setup instructions carefully to ensure a successful installation.

# 6.1 ICELink Platform: Clearing Workflow Overview

This sub-section contains a brief overview of the ICELink platform clearing workflow. For further details please consult ICELink.

Within ICELink, the Dealer initially alleges the transaction. It then flows through to the IA, FCM, and finally the CCP to affirm or reject. If at any point the transaction is rejected, it does not move on to the next participant.

A few important details to note regarding the workflow:

- If the Dealer is not self-clearing, the transaction will flow to the Dealer's FCM first for affirmation before moving on to the IA's FCM.
- ICELink allows the Dealer to perform a recall on a transaction if it has not yet been submitted to the CCP for Clearing. This essentially cancels the transaction, and any new transaction created in ICELink from that cancelled transaction will have a reference back to the original transaction id.
- If the IA's FCM rejects back to the IA, ICELink will cancel the original transaction, create a new transaction with the same economic details, and affirm the transaction on behalf of the Dealer. In this case the new transaction will have a reference back to the original transaction id.
- If the IA decides to allocate its side of the transaction across multiple FCMs, ICELink will set the original transaction to a status of 'MATCHED', create a new transaction per child allocation, and affirm the new transactions on behalf of the Dealer. In this case the new child transactions will have a reference back to the 'block' parent transaction. This flow is called 'Partial Clearing' of allocations since each child transaction is affirmed separately by the FCM/CCP.
- If the IA decides to allocate its side of the transaction within a single FCM, how ICELink behaves will depend on the Client setup. If the Client is configured for Partial Clearing, the behavior will be the same as outlined in the point above. Otherwise, the default behavior is to use 'Block Clearing' of allocations. In such a case, ICELink will simply add the new child deals to the existing transaction, and the FCM/CCP will have to affirm (or reject) them as one step at the block level.
- If netting/compression is enabled. Then either at the end of day/week Clearing house send the net or compressed offset values of all the booked deals between the parties. These offsets are received only for cleared deals via BulkTransactionListener.batchBrokeredTerminationChanged(). For this action, trade(s) in Calypso will be terminated and new trade(s) will be created for each offset value.

These messages are handled by the Calypso ICELink interface and will be described in subsequent sections. [See Section 6.6 for details on Allocations.]



## 6.2 Calypso Trade Creation Overview (Dealer / IA)

For a simple clearing workflow, an initial Bilateral trade will be created in Calypso between the two parties on the trade. Once the trade is affirmed for Clearing by the CCP, the Bilateral trade will be Novated to the CCP as counterparty. The result is that the original Bilateral trade is terminated in Calypso and a new trade is created for the cleared version.

In addition, before the trade is cleared, it can be recalled or rejected by various parties in the trade. All these statuses will be reflected in trade keywords (as described in Section 6.8).

The trade external reference will be formulated as:

ICELink\_<PO Shortname>\_<ICELink Deal Id>

# 6.3 Calypso Trade Creation Overview (FCM)

When running the ICELink interface in FCM mode, the only trade created will be the FCM vs. the CCP, and mirror trade will be created by the Clearing module by looking at the clearing-account setup, by looking at icelink participant-id and matching it with external reference of Clearing Account. The same participant-id is also stored on CCPReference trade keyword. The trade can either be affirmed or rejected by the FCM. Once the trade is affirmed by the FCM, and subsequently cleared by the CCP, the trade keywords will be updated to reflect that the trade is cleared.

In addition, before the trade is affirmed by the FCM, it can be recalled by the Dealer who initiated the trade. Again, this will be reflected in trade keywords.

The trade external reference will be formulated as: <CP.PO\_ExternalRefMapping>\_<ICELink Deal Id>

where <CP.PO\_ExternalRefMapping> is taken from the CalypsoMapping window, as shown below.

🗾 Calypso Mapping Window			-		×
	^ \$				
		Name:	ICELink/ExternalRefe	rence	
- Sees	Interf	ace Value:	cyp_hf1_1.icetrust		
Frequency	Calv	nso Value:	ICE		
Restructuring	Cury				
SalesPerson	Revers	se Default:			

If mapping is missing, then external reference will be formulated as only: <ICELink Deal Id>.



## 6.4 Calypso Trade Workflow Setup

To support all the transitions required by the ICELink workflow, the trade workflow setup for processing ICELink trades must support the following transitions:

- NEW (for creation of new trades)
- AMEND (trade approvals will come as keyword updates; see Section 6.8 for more details)
- CANCEL (if a trade is rejected or recalled by any party on the trade) (note that even if all 3 parties affirm the trade, the CCP may still reject it)
- TERMINATE (when the bilateral is novated after clearing, for Dealer / IA)

# Note: Please refer to the Calypso Data Uploader documentation if you wish to map these actions to custom names.

## 6.5 Novation Requirements

This section is only applicable when the ICELink interface is run in Dealer or IA mode. The trade is never novated when the ICELink interface is run in FCM mode.

### 6.5.1 Message Rule

By default, the Calypso Novation API will keep the same external reference on both the original trade and the newly cleared trade. This will cause an issue if updates are received from ICELink for the trade. To avoid this, you \*must\* add the UpdateTermination trade rule to the TERMINATE transition in your trade workflow for ICELink trades, so that the bilateral's external reference gets updated.

### 6.5.2 ClearingKeywords Domain

In addition, the Calypso Novation API will keep all the same keywords on both the original and cleared trades. In some cases, clients will want to remove certain clearing-specific keywords from the terminated bilateral trade. This can be controlled using the ClearingKeywords domain. Any keywords added to this domain will be removed from the bilateral trade when it is Novated.





Note: This domain is added and pre-populated for you by the DataUploader + ICELink schema installation scripts. Please review it as part of the ICELink interface setup.

### 6.5.3 Fees

By default, the Calypso Novation API will keep the trade fees on the original bilateral trade. If you wish to have the fees moved from the bilateral to the cleared trade, please add the fee name to the propagateFees domain. [Please ensure that you restart the Calypso DataServer and EventServer after this change.]

💋 Domain Values Window (User: calypso_user)									
Search:	propagate	Find	🗖 Value 🐧						
	ropagateFees UPFRONT_FEE ropagateTradeKeyword								

If you have mapped the default fee name (eg. UPFRONT\_FEE) using the Calypso Mapping Window, please ensure that you specify the new fee name in the propagateFees domain.

() Note: This will only affect fees with a date after the novation date. Also, the fees will remain on the original trade, but the Transfers will be cancelled by the Transfer Engine.

## 6.6 Allocation Requirements

This section describes how allocations are handled for each user mode supported by the ICELink interface, and how Block & Partial Clearing are handled for each. [See Section 6.1 for an explanation of Block & Partial Clearing.]

Note that in ICELink, only an IA can perform allocations, but the Dealer & FCM must handle allocation messages appropriately when the IA allocates.

## 6.6.1 Allocation Support (Dealer)

In the case of **Block Clearing**, the Dealer will only receive a single bilateral trade representing the full block amount, and that trade will be cleared by the CCP for the Dealer. [One bilateral USI will be assigned to the bilateral trade and another USI will be assigned to the cleared trade.]

Therefore, for Block Clearing no changes are required to support allocations; the incoming messages will look the same as non-allocated messages, and no additional workflow changes are necessary.

However, in the case of Partial Clearing – where a transaction is created for each child allocation – each child transaction will be received by the Dealer from ICELink and will be cleared separately by the CCP. [One bilateral USI will be assigned to the bilateral trade, but each cleared child trade will be assigned a separate USI.]

Therefore, for Partial Clearing the Dealer must setup additional workflow transitions to handle allocations – allocating the original bilateral trade and tracking the status changes of the child trades through to clearing. The required setup is the same as for IA mode, so please review the next section for setup details.



## 6.6.2 Allocation Support (IA)

In ICELink, when an IA allocates a transaction, ICELink will send message representations for the child allocations – either as deals within the same transaction message (Block Clearing) or as separate transactions (Partial Clearing). Note: In both the cases, one bilateral USI will be assigned to the bilateral trade, but each cleared child trade will be assigned a separate USI.

In either case, the ICELink interface can model this in Calypso using Calypso's allocation capabilities.

It is important to note that the ICELink interface is making use of the 'Externally Allocated' features of the Calypso allocation API. This allows the interface to mimic as-if the trade had been allocated within Calypso, but at the same time, it allows the ICELink interface to control how the trades are allocated and the exact numeric values of the notional and fee splits.

#### Workflow Setup [Dealer and IA]

To support all the transitions and statuses required by the allocation workflow - for both Block & Partial Clearing - please refer to the Calypso Allocation documentation for the recommended trade workflow setup. In general, you may follow the recommended workflow setup with the following exceptions:

- Do **not** add the 'NotAllocationChild' trade rule to the AMEND transition for allocated **child** trades. [This would prevent updates on the child trades which are required for updating the status keywords.]
- Do **not** add the 'PROPAGATE' transition for the allocated **child** trades, and do **not** setup the 'PropagateBlockTradeChangesAction' domain. [This would propagate keyword updates on the block trade to the child trades.]

For informational purposes, the allocation-related transitions applied by the ICELink interface are documented below. [Note that the transitions listed previously in Section 6.4 for non-allocated trades apply to allocated child trades as well and are not listed again.]

- ALLOCATE (this action is required to allocate the initial **block** trade, as well as update the allocated **block** trade as each child trade is saved in Calypso and linked to the block trade)
- AMEND (this is required for partial clearing, to update the allocated **block** trade once it is allocated)
- CANCEL (this is required to cancel the allocated **block** trade and the allocated **child** trades)
- VOID (this is required to cancel the allocated **child** trades in certain cases, to ensure that Transfers are handled correctly)
- NETTING/COMPRESSION: Client is expected to have standard TERMINATE action from VERIFIED/CLEARED
- Note: The VOID transition must move the trade to the CANCELED status (or whichever status you have mapped in Calypso to correspond to trade cancellation). If this is not done, the child trades' Transfers will not be cancelled correctly.











#### Workflow Setup [FCM]

For FCM, we have a dedicated workflow. This FCM workflow is bundled up and delivered as a part of Data Uploader module as 'TRADE\_WORKFLOW\_FCM.wf'. If the FCM workflow is not present or has not been imported, then default (generic) of Type ALL will be picked for saving a trade.



After importing 'TRADE\_WORKFLOW\_FCM.wf', user will have to make following changes onto the workflows to use it with ICELink platform.

- 1. Add a new transition: PENDING\_LIMIT\_REVERT ACCEPT LIMIT\_APPROVED
- 2. Add a new transition: REJECTED AMEND REJECTED

3. Rename action REJECT present between status 'PENDING\_LIMIT\_REVERT' and 'REVERT\_FAILED', to FORCE\_REJECT

4. Add a new transition: PENDING\_LIMIT\_REVERT - REJECT - REJECTED.



#### FCM Message Flows Scenario

#### From ICELink GUI

• FCM receives a trade.

We receive the ice-link message and create a corresponding trade in Calypso with status as 'PENDING\_LIMIT'

• FCM accepts the trade from ICELink GUI.

When we receive the ice-link message, we apply action 'ACCEPT' on a given trade and move it to status 'LIMIT\_APPROVED'

- CCP Message
  - If CCP Approves the trade

We receive the ice-link message and create a corresponding trade in Calypso with status as 'PENDING\_LIMIT'

- If CCP Reject the trade

We apply action REFUSE onto the trade and move it to status 'PENDING\_LIMI\_REVERT'

After that,

If FCM accept again; then apply action from PENDING\_LIMIT\_REVERT - ACCEPT - LIMIT\_APPROVED If FCM reject too; then apply action from PENDING\_LIMIT\_REVERT - REJECT - REJECTED

#### From Calypso GUI

• FCM receives a trade.

We receive the ice-link message and create a corresponding trade in Calypso with status as 'PENDING\_LIMIT'

• FCM accepts the trade from Calypso workflow.

We send 'AFFIRM' message to ICE.

If successfully send, we apply action ACCEPT and move the calypso trade to 'LIMIT\_APPROVED'.

- ICELink send back acknowledgement to Calypso for affirmed message.
   We apply action 'UPDATE' onto the Calypso Trade.
- FCM accepts the trade from Calypso workflow.

We send 'AFFIRM' message to ICE.

If successfully send, we apply action ACCEPT and move the calypso trade to 'LIMIT\_APPROVED'.





#### Message Rule

By default, if all child trades are canceled in Calypso due to rejection/recall in ICELink, the parent trade will not be marked cancelled. To address this, you **\*must\*** add the ICELinkCancelBlock trade rule to the VOID transition in your trade workflow for ICELink trades, so that the parent trade will get canceled if all child trades are canceled.

#### Additional Notes

- (1) Note: As part of using the 'Externally Allocated' features of the Calypso allocation API, each child trade is saved within Calypso one at a time, although the ICELink message contains all the child trades in a single message. Therefore, you may wish to setup the PARTIAL\_ALLOC status & required transitions in your trade workflow, as per the Calypso Allocation documentation. However, if this workflow is only being used for ICELink trades, you may choose instead to remove the PARTIAL\_ALLOC status & related transitions, and simply remove the CheckFullAllocation trade rule from the ALLOCATE & AMEND transitions.
- () Note: The TERMINATE and NOVATION transitions are not required for ICELink allocated block trades.

### 6.6.3 Allocation Support (FCM)

In the case of Block Clearing, if the FCM is clearing a trade on behalf of a Dealer, the same rules apply as for the Dealer and therefore no additional workflow setup is required; only a single trade will be received from ICELink and the incoming messages will look the same as non-allocated messages.

In the case of Partial Clearing for a Dealer, or when clearing for a client in the case of both Block & Partial Clearing, the FCM **will** receive the child deal details.

In those cases, from a business perspective, each child trade must be individually limit checked and the block trade has no business meaning. Therefore, within Calypso we **only** create the child trades and not the allocated block trade. [Certain allocation related keywords are still set for informational purposes.]

As such, in those cases as well, no changes are required to support allocations, since each child trade will be handled as if it were a separate non-allocated message.

- (1) Note: In the case of Block Clearing for the Client, the child trades must be affirmed / rejected as one at the block level, all or nothing. If the ICELink interface's bidirectional functionality is not being used, this single affirm / reject action must be done from within the ICELink Web GUI. Since this is done manually, you will need to first review the credit check results of all linked child trades in Calypso to determine whether to affirm or reject the block transaction from the ICELink Web GUI. In addition, you will need to ensure that the credit consent workflow can handle a reject message coming for a trade that passed the initial credit check.
- (1) If the ICELink interface's bidirectional functionality is being used, please review the next section to ensure the trade workflow is correctly setup to handle Block Clearing for the Client.



### 6.6.4 Netting/Compression

In case if netting/compression is enabled. Then either at the end of day/week Clearing house send the net or compressed offset values of all the cleared deals between the parties. These offsets are received only for cleared deals via batch Brokered Termination Changed API. For this action, trade(s) in Calypso will be terminated and new trade(s) will be created for each offset value.

As a part of Netting and Compression, the ICELink interface supports:

- Netting
- Portfolio Transfer

This is implemented in 2-way process. Once the trade is netted or transferred, ICELink platform will send the netting-id to Calypso. Using netting-id, ICELink Engine will request ICELink platform to send collections of all termination trades and remnant trades which are part of that netting cycle. Once we received all the deals, we process those messages in Calypso. For termination process, standard termination transition/action will be applied.

#### Netting/Compression

A user books 2 trades with the same counterparty with same underlying product, rate and spread.

- CDSIndex [Deal-Id 123]: Buy 5M (CDX-EMS19V3-5Y)
- CDSIndex [Deal-Id 124]: Sell 12M (CDX-EMS19V3-5Y)

Then in case of netting event followings trades will be created in Calypso.

ICELink Deal-ID	Action	Direction	Amount	Product Description
Deal-Id 123	New	Buy	5M	CDSIndex (CDX-EMS19V3-5Y)
Deal-Id 124	New	Sell	12M	CDSIndex (CDX-EMS19V3-5Y)
Deal-Id 123	Termination	Buy	5M	CDSIndex (CDX-EMS19V3-5Y)
Deal-Id 124	Termination	Sell	12M	CDSIndex (CDX-EMS19V3-5Y)
Deal-Id 125	New	Sell	7M	CDSIndex (CDX-EMS19V3-5Y)

#### Portfolio Transfer - Between two clients

In the case of Portfolio Transfer, FCM can transfer the cleared trade from one client sub-account to another client sub-account.

At the time Portfolio Transfer users can decide the amount and upfront fee to be transferred.

E.g. FCM decide to transfer the cleared trade from HSBC\_LON to HSBC\_USA, and books 2 trades with same counterparty with same underlying product, rate and spread.

• CDSIndex - Buy - 5M (Index1) HSBC\_LON (New)



• CDSIndex - Buy - 5M (Index1) HSBC\_USA (TransferTo)

Then in case of portfolio transfer event, following trades will be created in Calypso.

ICELink Deal-ID	Action	Client	Direction	Amount	Product Description
Deal-Id 512	NEW	HSBC_LON	Buy	5M	CDSIndex(CDX-EMS19V3-5Y)
Deal-Id 513	NEW	HSBC_LON	Sell	5M	CDSIndex(CDX-EMS19V3-5Y)
Deal-Id 514	NEW	HSBC_USA	Buy	5M	CDSIndex(CDX-EMS19V3-5Y)

### Portfolio Transfer – Between two FCM

In the case of Portfolio Transfer, a client can transfer the cleared trade from one FCM to another FCM. At the time Portfolio Transfer client can decide the amount and upfront fee to be transferred.

E.g. Client decides to transfer the cleared trade from CYP\_PBOUT\_FCM to CYP\_PBIN\_FCM.

For CYP\_PBOUT\_FCM, two trades will be created.

- CDSIndex Buy 5M (Index1) CYP\_PBOUT\_FCM (New)
- CDSIndex Sell 5M (Index1) CYP\_PBOUT\_FCM (Offset)

ICELink ID	Action	Client	Direction	Amount	Product Description
Deal-Id 512	NEW	CYP_PBOUT_FCM	Buy	5M	CDSIndex(CDX-EMS19V3-5Y)
Deal-Id 513	NEW	CYP_PBOUT_FCM	Sell	5M	CDSIndex(CDX-EMS19V3-5Y)

For CYP\_PBOUT\_FCM, one trade will be created.

• CDSIndex - Buy - 5M (Index1) CYP\_PBIN\_FCM (New)

ICELink ID	Action	Client	Direction	Amount	Product Description
Deal-Id 514	NEW	CYP_PBIN_FCM	Buy	5M	CDSIndex(CDX-EMS19V3-5Y)

Note: If Netting is enabled for the client, then deal-id 512 and 513 will be terminated as a part of <u>Netting/Compression</u> process. For more information, please refer section Netting and Compression.



## 6.7 Bidirectional Requirements

This section describes how to setup bidirectional functionality for each user mode supported by the ICELink interface.

## 6.7.1 Bidirectional Overview

Bidirectional functionality allows a user to perform actions from within Calypso and have the appropriate message sent to ICELink using the ICELink messaging API.

The bidirectional functionality is implemented as 'Platform' workflow rules. Simply add the appropriate workflow rule to a Calypso action and applying that action on a trade will prompt the ICELink engine to create the external message and send it to ICELink.

Once the message has been acknowledged by ICELink, the following will occur:

- The corresponding PLATFORMMSG workflow message will go to COMPLETED.
- The trade keyword 'PlatformSubmitStatus' will be updated with '<Action> Successful'
- ICELink will send the appropriate trade message to update the trade in Calypso (i.e. as-if the action had been performed from the ICELink web interface).

For example, if a user wants the ability to Affirm trades from within Calypso, simply create an action (e.g. Affirm) and add the PlatformAffirm Trade Rule to that action. Typically, the action will go back on itself (e.g. Pending  $\rightarrow$  Affirm  $\rightarrow$  Pending), since ICELink will send Calypso a separate trade message **AFTER** the Affirm action is processed successfully, and **that** separate message will update the trade details/status in Calypso accordingly.

If any errors occur sending the message to ICELink, whether due to missing details or invalid action applied, the PLATFORMMSG workflow message will have errors connected to it which can be reviewed to determine the error, address the specific issue, and resend the workflow message.

## 6.7.2 Bidirectional Support (All)

Note that in general, all mapping setup instructions which applied to incoming messages (eg. Calypso Mapping, Legal Entity attributes, Product Mapping) will be used in the opposite direction when sending outgoing messages to ICELink. However, in certain cases, additional keywords will need to be set on the Calypso Trade before submitting it to ICELink to ensure that the ICELink engine will correctly pick up the trade message and be able to create a complete outgoing message to be sent to ICELink.

Once the trade is booked in Calypso and all keywords have been set – and assuming the appropriate Trade Rule has been added to the appropriate workflow action – simply applying the action on the trade will send the message to ICELink.

You may wish to use trade templates or custom workflow rules to populate at least the keywords which will stay the same across all trades booked for ICELink. Please review the sections below for specific requirements for each user mode.

# Note: All Calypso Legal Entities on the trade or contained in keywords must be setup as per the instructions in Section 3 of this document in order for outgoing messages to work correctly.



## 6.7.3 Bidirectional Support (Dealer)

The Dealer is able to perform the following actions using the bidirectional functionality:

- Allege [PlatformAllege Trade Rule]
- Recall [PlatformPull Trade Rule]

In both these scenarios, additional Trade Keywords should be set for the trade to be processed correctly by the ICELink engine and enable it to send the message to ICELink.

#### Dealer Allege

The following Trade Keywords must be set on all trades which will be Alleged to ICELink from within Calypso:

- TradeSource: This should always be set to 'ICELink' for all trades being submitted to ICELink.
- PlatformAPIUser: This should be set to the ICELink login id being used by the ICELink engine. This is important for cases where multiple ICELink engines are being run.
- CCP: This should be set to the Calypso Legal Entity short name for the CCP.
- CCPClearingBroker: In the case of a Dealer which is not self-clearing, this should be set to the Calypso Legal Entity short name for the FCM.

#### Dealer Recall

As a trade must first be Alleged before it can be Recalled, all required trade keywords will already have been set previously on the trade. In the case of Recall, there is an optional keyword that may be set:

• PlatformRejectReason: You may wish to set a reason text in this field, which will be passed along in the ICELink message and received by other parties on the trade. If not set, it will be defaulted to 'Rejected'.

### 6.7.4 Bidirectional Support (IA)

The IA is able to perform the following actions using the bidirectional functionality:

- Affirm [PlatformAffirm Trade Rule]
- Reject [PlatformReject Trade Rule]

As the trade will have already been created for the IA in Calypso by the ICELink interface once the Dealer alleges it, most of the required trade keywords will already have been set for you on the trade, but there are a few additional keywords to set. In addition, when affirming an allocated trade, additional booking steps and setup are required.

#### IA Affirm

The following keywords should be set:

• CCPClearingBroker: This should be set to the Calypso Legal Entity short name for the FCM.



• ICELinkTransactionRefld: This keyword is optional. It allows you to set some custom reference on that trade which will be saved in ICELink.

#### IA Reject

The following optional keyword may be set:

• PlatformRejectReason: You may wish to set a reason text in this field, which will be passed along in the ICELink message and received by other parties on the trade. If not set, it will be defaulted to 'Rejected'.

#### IA Affirm with Allocations

From the ICELink Web GUI, the IA may allocate the trade alleged by the Dealer before affirming. This action is supported from within Calypso as well in bidirectional mode.

For this to work correctly, you **must** add the following workflow rules to the appropriate transitions:

- 'ICELinkUpdateExternallyAllocated' trade rule to the VERIFIED ALLOCATE ALLOCATED transition.
- 'PlatformAffirm' trade rule to the ALLOCATED AFFIRM ALLOCATED transition (as mentioned previously, the action name can be of your choosing, this is just an example)
- 'UpdateAllocationChild' trade rule to the AFFIRM action (or whichever action you have added the PlatformAffirm trade rule to)

Then, to perform the allocate & affirm, simply allocate the trade from within Calypso using the standard Calypso screens. Once allocated, you must set the CCPClearingBroker keyword on each child trade. [You may also wish to set the CCPClearingBroker keyword on the parent trade before allocating, which will copy the value to all children once you allocate.] Then affirm the allocated parent trade as you would a normal trade.

- () Note: Both Block & Partial Clearing of allocations are supported in this mode. Simply set the CCPClearingBroker keyword on each child trade to the different FCMs you wish to use for each trade.
- (1) Note: The 'UpdateAllocationChild' trade rule will change the External Reference & remove certain control keywords on the child trades, to ensure future incoming ICELink messages for the child trades will be processed correctly. This is controlled by the 'AllocationKeywords' domain, which is added and prepopulated for you by the DataUploader + ICELink schema installation scripts. You may wish to add additional keywords to this domain, but you must not remove any of the default keywords added to the domain as this will break the bidirectional functionality.

### 6.7.5 Bidirectional Support (FCM)

The FCM is able to perform the following actions using the bidirectional functionality:

- Affirm [PlatformAffirm Trade Rule]
- Reject [PlatformReject Trade Rule]



#### FCM Affirm

There are no additional keywords to set.

#### FCM Reject

The following optional keyword may be set:

• PlatformRejectReason: You may wish to set a reason text in this field, which will be passed along in the ICELink message and received by other parties on the trade. If not set, it will be defaulted to 'Rejected'.

#### Workflow Setup

Reference workflow setup for bi-directional FCM.



#### NOTE : This reference workflow is also shipped with ICELink rel-zip with name as 'ICELinkFCM\_Reference\_Trade.wf'



#### Block Clearing of Allocations

As described in the 'Allocation Support (FCM)' section of this document, in the case of Block Clearing of allocations, the child trades must be affirmed / rejected as one at the block level, all or nothing. This presents a difficulty in the typical FCM workflow where each child trade is run through a separate limit check, with a separate pass/fail result.

To address this, the ICELink interface handles keeping track of the status of all child trades, and once all have reached a final state of pass/fail, it will send a single response back to ICELink. If at least one has failed, it will send a single Reject message for the entire block of trades. If all trades are passed, it will send a single Affirm message for the entire block.

For the engine to detect this, it requires that specific domains be setup to specify what the pass/fail statuses are in the trade workflow. These domains are:

- Clearing.Trade.AcceptStatus
- Clearing.Trade.RejectStatus

In addition, these two statuses should have the following actions defined on them:

- AMEND (action should go back onto the same status, so that the PlatformSubmitStatus keyword can be updated when the message is sent to ICELink)
- CANCEL (the Accept status requires a cancel action as well, as you may still need to reject a trade due to another child trade failing to pass the limit check)

To further clarify how the workflow would look, we have included a sample FCM workflow under the resources directory, FCM\_Trade.wf

## 6.8 Workflow Keywords

Throughout a trade's lifecycle, trade status keywords will be updated on the trade to reflect its approval state. These keywords can be used together with Static Data Filters to move the trade through any custom workflow / status you create in Calypso.

The pertinent keywords are described below:

• ICELinkTPApprovalStatus: This keyword reflects the status of the trade from the TP (Trade Party) point of view.

For example, if you are running ICELink as a self-clearing Dealer and Allege a trade through the ICELink Web GUI, the trade will flow into Calypso with a value of 'Done' in this keyword. If you are clearing through an FCM, the value will be 'PendingFCM'.

If a Dealer alleges a trade to you, the initial value would be 'Pending', and once you chose a FCM in ICELink it would change to 'PendingFCM'. The valid values for this keyword are: Pending, PendingFCM, Done, Rejected, RejectedFCM, and Recalled.

If you are running ICELink as an FCM, this keyword will reflect the approval status of the trade from the perspective of whichever side you are representing on the trade.





- ICELinkCPApprovalStatus: This keyword works similarly to the one above, except that it reflects the status of the trade from the counterparty's point of view.
- **CCPStatus**: Once both keywords above have a value of 'Done', which means that all relevant parties, including FCMs, have approved, this keyword will get populated. The initial value will be 'Sending' to reflect that it has been sent to the CCP for clearing. Valid values after that are either Cleared or Rejected.



# ICELink Engine Setup

The ICELink Engine is responsible for getting messages from the ICELink platform and handing it off to the appropriate workflows. When run in FCM mode, it is also responsible for sending Affirm/Reject messages back to ICELink.

Please review the standard Calypso documentation for Engine setup to read about several useful engine parameters (such as thread count) and how to set them.

# 7.1 Configure the Engine

All of the database-based Engine configuration is completed as part of applying the schema, including the engine name, event subscription, event filter, event policy, as well as assigning a unique id to the Engine.

Server <del>+</del>	Metrics <del>+</del>	Profiler <del>+</del>	Manage <del>v</del>	Monitoring <del>,</del>	Logs	4	
					Enviro	onment: ICELINKTRUNK	User: calypso_user
En aina (	~~						
Engine (	.onnguratior	1					
Engine Name:	0	E	ingine ID:	Max Queue Size: 😮	Max Bate	ch Size: 😗 🛛 Numl	ber of Threads: 😗
CELinkEngi	ne	ľ	107				
ingine Class:				Event Pool Policy: 😗		Pricing Environment	0
com.calypso	.tk.engine.ICELink	Engine		ICELinkEngine	,	•	
Display Name:	0	Appli	cation Type:	Save settle position c	hanges: 🧐		
ICELINK Eng	ine	Engi	neServer	Coeffermation attailant			
Description:				PROJECTED_DAYS	.62		
				REVERSAL_CRE			
Persisted Eve	nt Configuration:			STARTUP			
PSEventAcc	- countBilling	•	00	TIMEOUT RESTART			
PSEventICE	Link			LISE BOOK PRICING E	ENV		
PSEventPlat	tormPublish			VALUATION TIMES			
			-	VALUATION TIMEZON	IES		
vent Filters:							
AllTransfers	KnownEventFilter	•	00	VERDICINECK EIDET			
CELinkEngi	neEventFilter		-				
				XFER_NEVER_BV			
			-	AFER_NEXT_EVENT			
ingine Manag	er Configuration:	Start	: on Startup:	XFER_PAST_GENERATI	ION		
engineserve	r 🔻			XFER_USE_AUTOMATI	C_ACCOUNT		
				XFER_USE_MONEYDIFI	F		
				XFER_USE_POS_AGGR	EGATION_ONL	LY	
				XFER_USE_REVERSE			
				config		icelink.properties	-

April 2024



## 7.2 Setup the IceLink config file

To run the ICELink Engine out-of-the-box you will need a properties file with the name "icelink.properties" with the appropriate connection settings. [You may change the prefix, but it must end with ".properties"]

A sample file is included under <calypso home>/resources with the name ".sample" as the suffix. Please change the Mode and ICELINK connection properties to the correct values for your setup.

Valid values for the Mode property are: Dealer, IA, FCM

Property 'messageFileDir' can be used to change the directory location for generating and saving of debug xml files produced for ICELink\_DEBG\_XML log category [for more information please refer section:7.4.2]. By default, debug xml files are generated at <UserHome>/Calypso/ICELink directory.

Note: The engine name is controlled by the property file name. If you choose to rename the default property file, you will need to update the engine name, event subscription, event filter, and event policy entries within Calypso.

## 7.3 Setup the gatewayservice config file

In addition to viewing exception messages in the Task Station, you may also configure ICELink to write all Success or Reject messages to the file system. This is configured within the gatewayservice.properties file distributed with Data Uploader.

A sample file is included under <calypso home>/resources with the name ".sample" as the suffix. To configure messages to be written to the file system, set the following in the config file:

ICELinkPublishers=File,RejectedFile

At the least, you must have an empty entry (ie. ICELinkPublishers=) to avoid any spurious errors in the log.

You can customize the FilePublishToLocation property to control where these files are written.

## 7.4 Launching the ICELink Engine

The ICELink engine is launched as part of the Engine Server.

Based on your volume, you may need to increase the Xmx and MaxPermSize values. Also, for -config you may leave off the ".properties" part of the name and it will be assumed for you.

With the ICELink Engine operating, you can then allege/affirm trades through the ICELink Web GUI. The ICELink Engine will process the trade messages and create corresponding trades in Calypso.

The Task Station will display any errors that may occur.



#### Adding Logging Categories

To see logging messages for the Data Uploader module and ICELink interface you need to set the following log categories:

- UPLOADER: Set this to see logging for the Data Uploader translation from the internal Calypso xml format to the actual trade object.
- ICELink: Set this to see logging for the ICELink translation from the external format to the internal Calypso xml format.
- ICELINK\_DEBUG\_XML: Set this category to create xml files of the incoming messages from the ICELink platform as well as the interim Calypso xml format created by the translation under \$USER\_HOME/Calypso/<TradeSource>
- ICELINK\_DEBUG\_API: Set this category to see additional logging statements for the incoming ICELink platform messages. These logging statements are helpful to see the entire message coming in from the ICELink platform before they are interpreted by the ICELink Engine.

## 7.5 Troubleshooting

The ICELink Connection API automatically attempts to reconnect if a connection with the ICELink platform is lost. On reconnect, it will first process any queued messages, and then be available to process new messages.

In case messages are not being received by the engine, please check the log files and Task Station for connection or processing errors.

## 7.6 Running Multiple Engines

In some scenarios, Calypso customers might have a requirement to connect as multiple ICELink users at a single time. To support this, the ICELink Engine supports running multiple engines at the same time, with different names and configuration for each.

The default installation assumes a single engine being run with the engine name ICELinkEngine, and the SQL schema files install the entries required for that. If you wish to run multiple engines, you will have to setup similar entries for your engine.

For your convenience, an additional schema file (ICELinkMultiEngineSchemaData) has been included in the installation with the appropriate entries to setup your new engine. Just replace the string 'ICELinkClientEngine' in the schema file (in 5 places) with your new engine name.

In addition to running the schema file, you will need to setup the Engine ID. See section 7.1 above for more details.

Lastly, you will need to name your properties file accordingly. For engine name 'ICELinkClientEngine' you would name the file 'icelinkClient.properties' and set the mode and connection details appropriately for that engine. Then follow the steps in Section 7.4.3 above to start your engine specifying the new properties file in the config argument.



# Performance and Stability Improvement

In order to improve performance in ICELink, the framework is changed to execute.

To improve the performance for message processing we now have the provision to perform the upload process in the Engine side via the API instead of DataServer via the workflow rules. Workflow rules are executed in the data server thus keeping DataServer busy every time data is uploaded. Client-side execution has the advantage of using the API and cache. And it can thus perform better and DataServer is free for performing other tasks.

The new enhancement provides the equivalent features that were available via the workflow based approach such as:

- Persistence of external messages as BO messages.
- Ability to Re-process failed messages.
- Maintain the order in which the messages are received.
- Acknowledgement generation.

To enable the clients to use the new approach, we have introduced two new properties in "icelinkcsv.properties". Please note that if these are not set, we will default to the current way of processing in DataServer via the workflow rules which is the BOMessage mode.

The following are the properties:

- uploadMode
- persistMessages

#Valid combination for uploadMode and persistMessage

π					
#					
#				PERSIST MESSAGES	
#					
#			None	Failure	All
#					
#	M	Local	No BOMessage	BOMessage will be created	BOMessage will be
#	0		will be created	only in case of failure	always created
#	D				
#	E	BOMessage	Not Applicable	Not Applicable	BOMessage will be
#	S				always created
#					

#

# For better performance it is recommended to use uploadMode as Local with persistMessages as Failure.

- uploadMode: Possible values are 'BOMessage' and 'Local', default value is 'BOMessage'.
  - BOMessage: It is self-explanatory; it is the workflow-based model that every interface currently uses.



- Local: It is for using the API however when using the Local mode, we need another property "persistMessages".
- **persistMessages**: Applicable only when "uploadMode" property is set to 'Local'. Possible values are 'All', 'None' and 'Failure', by default it is set to 'None'.
  - All: External messages are always persisted as BO messages.
  - None: External messages are \_NOT\_ persisted as BO messages. If the message fails in translation or validation, the message needs to be resent or handled via custom code.
  - Failure: External Messages are persisted only in case of failure in translation or validation. This is the \_recommended\_ configuration for the 'Local' mode as this will not save any BO messages in case it is all processed fine and only create messages in case of any translation failures which will enables the failed messages to be reprocessed as before and also improve performance by not saving the success BO messages.

Please note that in the 'Local' mode for message reprocessing "**UpdateManagerEngine**" needs to be running. Failed messages will be stuck in "PENDING" status and pending Messages are re-processed via the UpdateManagerEngine. The message workflows are changed to generate an event 'PSEventUploadReprocess', every time a failed message is re-processed. Update Manager Engine would then receive these events and processes them again to generate acknowledgement if needed.

The following Events are required by the Update Manager Engine.

• PSEventUploadReprocess

Server <del>+</del>	Metrics +	Profiler <del>v</del>	Manage <del>v</del>	Monitoring <del>,</del>	Logs		4			PSO®
						Enviro	onment: TRUNK	User:	calypso_user	[Logout
Engine	Configuration	n								
Engine Name	: 😢	E	Engine ID:	Max Queue Size: 🛛	Max	× Batch	Size: 😮	Number	of Threads: 😗	
UpdateMana	agerEngine		107							
Engine Class:				Event Pool Policy: 😗			Pricing Enviror	ment: 🕜		
com.calyps	o.tk.engine.Update	ManagerEngine				•	default	•		
Display Name	: 😯	Appli	ication Type:	Save settle position c	hanges:	0				
Update Man	ager Engine	Engi	ineServer	•						
Description:				Configuration attribut	tes					n
				Attribute Name			Attribute V	alue	Ê	
Persisted Eve	ent Configuration:			BALANCE_MODE						
PSEventAc	countBilling	•	00	CLASS_NAME						
PSEventRe	publish			DISPLAY_NAME						
PSEventUp	oadReprocess			DateType						
			-	EVENT_ORDER						
Event Filters:				EXCLUDE_PRODUCTT	YPE					
AllTransfers	KnownEventFilter	•	00	EXCLUDE STATUS						
UpdateManagerEngineEventFilter		-	HANDLE FUTURE LIO	CASH FL	LOWS					
				IGNORE ACTION						
			-	INSTANCE NAME						
Engine Mana	ger Configuration:	Star	t on Startup:	INV MAY POSITION						
engineserve	er 🔻	Image: A start of the start		LIQUIDATION TIMEO	IT					
					M					
				MCC DATE KENWORD						
				MCC_DATE_RETWORL	,					
				MCC_FEED_NAME					-	



# Keywords

In addition to the keywords listed above in the "Trade Workflow" section, the Calypso Standard Clearing keywords – as well as ICELink specific keywords – will be populated for incoming trades accordingly. A full list is provided in the table below.

# 9.1 General Keywords

Keyword Name	Description	Comments
TradeSource	Always set to 'ICELink'	Used for engine logic behind the scenes; do not change.
PlatformAPIUser	ICELink login id for engine	Used for engine logic behind the scenes; do not change.
USIPrefix	ICELink RegReporting value	
USIValue	ICELink RegReporting value	
ReportingParty	ICELink RegReporting value	
ССР	LE short name	
CCPClearingBroker	The clearing broker (when available in the trade)	
OriginalCounterparty	Bilateral counterparty	Set on the cleared trade, to show the original counterparty before novation
ICELinkTPApprovalStatus		Described above in Workflow section
ICELinkCPApprovalStatus		Described above in Workflow section
Platform	Always set to 'ICELink'	
PlatformStatus	ICELink Deal State	
PlatformTransactionId	ICELink Transaction Id	
PlatformTradeld	ICELink Deal Id	
CCPStatus	Sending, Cleared, Rejected	
CCPClearedDate	Date when trade was cleared by CCP	
CCPMessageTimestamp	GMT timestamp of last message to/from CCP	



Keyword Name	Description	Comments
CCPTradeID	CCP assigned deal id	
PriorUSIPrefix	ICELink RegReporting value	Set on a cleared trade, with the USI Prefix of the original bilateral trade
PriorUSIValue	ICELink RegReporting value	Set on a cleared trade, with the USI Value of the original bilateral trade
BlockUSIPrefix	ICELink RegReporting value	Set on an allocation leg trade, with the USI Prefix of the allocation block trade
BlockUSIValue	ICELink RegReporting value	Set on an allocation leg trade, with the USI Value of the allocation block trade
PlatformBlockTradeld	ICELink Deal Id of the Block deal	Set on an allocation leg trade, with the ICELink Deal Id of the allocation block trade
PlatformRejectCode	Reject code set by user in ICELink Web GUI	
PlatformRejectReason	Reject reason set by user in ICELink Web GUI	
CCPAccount	Is this trade in a CLIENT or HOUSE account at the CCP	In Dealer/IA modes, this keyword isn't set until we can determine whether or not they are using a FCM for clearing
CCPOriginCode	Did this trade originate due to CLIENT or HOUSE activity	In Dealer/IA modes, this keyword isn't set until we can determine whether or not one side of the trade is using a FCM for clearing
CCPAccountReference	The clearing house account name. This is used to map the client/house account at the clearing house to the Calypso account	For FCM mode, this will be set to the Client's ICELink entity short code. For Dealer/IA modes this will be set to their ICELink entity short code
IS_CLIENT	Is this the FCM->Client trade or FCM->CCP trade	Only applicable for FCM mode, Cleared trades.
		Always set to false by the interface. The client linked trade created by the Clearing module will have this field set to true
ICELinkTransactionRefld	Id entered into ICELink web interface as an External Reference to other systems	This is not a mandatory ICELink field
PlatformOriginalTransactionId	If this transaction resulted from a workflow action on a previous transaction, this would contain that previous transaction id	When the FCM rejects the client, ICELink creates a new transaction and affirms it on behalf of the dealer.



Keyword Name	Description	Comments		
		Also, when a client allocates across multiple FCMs, ICELink creates one new transaction per FCM allocation		
AllocationClearingType	Specifies whether this trade is part of a Block Clearing or Partial Clearing allocation workflow	Set to "Partial" if partial, "Block" if block, and empty if this is not part of an allocation workflow at all		
PlatformSubmitStatus	Specifies the status of any outgoing action that was submitted to ICELink (i.e. bidirectional request)	Possible values: • <action> Pending • <action> Successful • <action> Failed</action></action></action>		
PlatformAllegedTime	Specifies the ICELink Deal alleged time in GMT	Date in format: yyyy-MM-DD hh:mm:ss a z Where a = AM/PM z = Timezone		
PlatformLastModifiedTime	Specifies the ICELink Deal last modified time in GMT	Date in format: yyyy-MM-DD hh:mm:ss a z Where a = AM/PM z = Timezone		
CCPNettingId	A common netting Id assigned by the CCP. It will be common for all trades that are part of the netting. Since all trades (CDS, CDSIndex) which are part of netting of netting process will have the same netting-id, so to make it unique we add underlying-id at the end of netting id.	E.g: ICELink Netting-Id: TCC_20170630_hcf_pbout_fcm Netting id populated onto keyword: TCC_20170630_hcf_pbout_fcm_2312 -where 2312 is underlying product id		
CorrelationID	ICELink correlation id which is a part of netting task			
ClearingConfirmedCorrelationID	ICELink clearing confirmed correlation id which is a part of netting task			
CCPMessageTimestamp	Time stamp of last clearing message	Date in format: yyyy-MM-DD hh:mm:ss a z Where a = AM/PM z = Timezone		



Keyword Name	Description	Comments
CCPClearedDate	The date the trade is registered with the clearing house.	Date in format: yyyy-MM-DD hh:mm:ss a z Where a = AM/PM z = Timezone
IncomingMessageID	Incoming message id associated to netting process	
ClearingConfirmedIncomingMessageID	Clearing confirmed incoming message id associated to netting process	
CCPOriginatingEvent	This will be stored on the NEW trade with the value – NetPositionTrade/ TransferTrade.	
CCPTerminatingEvent	This will be stored on the NEW trade with the value – NetPositionTrade/ TransferTrade.	
WareHouse_TRI	Unique WareHouseTRI Id provided by ICELink	

# 9.2 MIFID Keywords

Keyword Name	Description	Comments
OrderBuyer	As entered in ICELink	If Calypso LE found, its short code will be added, else participant code will be added
OrderSeller	As entered in ICELink	If Calypso LE found, its short code will be added, else participant code will be added
ExecutionDateTime	Trading date and time microseconds	Date in format: yyyy-MM-dd hh:mm:ss:SSS a z
ReportingMIFIDTransactionIdentifier	Generated or user submitted per transaction	
ReportingMIFIDShortSaleIndicator	User submitted per transaction	
ReportingMIFIDOTCPostTradeIndicator	Selectable flags submitted per transaction	



Keyword Name	Description	Comments
ReportingMIFIDWaiverIndicator	Selectable flags submitted per transaction	
ReportingInvestmentDecisionMakerName	User submitted per transaction, can be defaulted based upon user logged in	In case if Investment decision maker is 'National{LEI}', then we look for Calypso LE. If found its short code will be added, else participant code will be added
ReportingInvestmentDecisionMakerLocation	One or more locations as selected in ICELink	
ExecutionVenueMIC	MICs taken from static, selectable on a transaction	
InstrumentISIN	Derived from transaction details if possible, can be overridden on transaction	
InstrumentCFI	We capture the value passed from ICELink	
InstrumentFullName	We capture the value passed from ICELink.	
ReportingMIFIDCounterparty	Calypso Legal entity for the corresponding party selected in ICELink as the Reporting Party	If Calypso LE found, its short code will be added, else participant code will be added
ReportingTradingCapacity	A default value for trading capacity, can be overridden on transaction	
ReportingTraderName	Trade Name	If Calypso LE found, its short code will be added, else participant code will be added
ReportingTraderLocation	One or more locations as selected in ICELink	
TypeOfOrder	As entered in ICELink	
OrderRateOfExchange	As entered in ICELink	
OrderTotalCommissionAndExpenses	As entered in ICELink	
OrderClientSettlementResponsibilities	As entered in ICELink	
OrderTransmission	Boolean. True if selected in ICELink	



## 9.3 ICELink Custom Fields

Custom fields present in the ICELink GUI can be imported as Calypso trade keywords.

To accomplish this, the custom field name and its corresponding trade keyword should be added in the Calypso Mapping Window, in the 'Keywords' category under the 'ICELink' interface section. [See screenshot below.]

If a custom field is received from ICELink but is missing from the mapping window, it will be ignored and won't be imported as a Calypso trade keyword.

# (i) Note: If you are mapping the custom field to a new trade keyword, please remember to add the keyword name to the 'tradeKeyword' domain so that it appears in the trade keyword window.

	Name: Interface Value: Calypso Value: Reverse Default: << Add >> Remove Configure Interfaces Configure Types	ICELink/Keywords Free1 ICELinkFree1
Load		Close