Global Fixed Income Committee (GFIC)

Recommended Practices for Party Entitlement

OTC Markets: fixed income instruments

Editor(s): Etrading Software Limited
Date: Tuesday, 05 November 2013
Version: 1.1 Final
Paper designation: Analysis

DISCLAIMER

THE INFORMATION CONTAINED HEREIN AND THE FINANCIAL INFORMATION EXCHANGE PROTOCOL (COLLECTIVELY, THE "FIX PROTOCOL") ARE PROVIDED "AS IS" AND NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL MAKES ANY REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, AS TO THE FIX PROTOCOL (OR THE RESULTS TO BE OBTAINED BY THE USE THEREOF) OR ANY OTHER MATTER AND EACH SUCH PERSON AND ENTITY SPECIFICALLY DISCLAIMS ANY WARRANTY OF ORIGINALITY, ACCURACY, COMPLETENESS, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.SUCH PERSONS AND ENTITIES DO NOT WARRANT THAT THE FIX PROTOCOL WILL CONFORM TO ANY DESCRIPTION THEREOF OR BE FREE OF ERRORS.THE ENTIRE RISK OF ANY USE OF THE FIX PROTOCOL IS ASSUMED BY THE USER.

NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL SHALL HAVE ANY LIABILITY FOR DAMAGES OF ANY KIND ARISING IN ANY MANNER OUT OF OR IN CONNECTION WITH ANY USER'S USE OF (OR ANY INABILITY TO USE) THE FIX PROTOCOL, WHETHER DIRECT, INDIRECT, INCIDENTAL, SPECIAL ORCONSEQUENTIAL (INCLUDING, WITHOUT LIMITATION, LOSS OF DATA, LOSS OF USE, CLAIMS OF THIRD PARTIES OR LOST PROFITS OR REVENUES OR OTHER ECONOMIC LOSS), WHETHER IN TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY), CONTRACT OR OTHERWISE, WHETHER OR NOT ANY SUCH PERSON OR ENTITY HAS BEEN ADVISED OF, OR OTHERWISE MIGHT HAVE ANTICIPATED THE POSSIBILITY OF, SUCH DAMAGES.

DRAFT OR NOT RATIFIED PROPOSALS (REFER TO PROPOSAL STATUS AND/OR SUBMISSION STATUS ON COVER PAGE) ARE PROVIDED "AS-IS" TO INTERESTED PARTIES FOR DISCUSSION ONLY.PARTIES THAT CHOOSE TO IMPLEMENT THIS DRAFT PROPOSAL DO SO AT THEIR OWN RISK.IT IS A DRAFT DOCUMENT AND MAY BE UPDATED, REPLACED, OR MADE OBSOLETE BY OTHER DOCUMENTS AT ANY TIME.THE FPL GLOBAL TECHNICAL COMMITTEE WILL NOT ALLOW EARLY IMPLEMENTATION TO CONSTRAIN ITS ABILITY TO MAKE CHANGES TO THIS DOCUMENT PRIOR TO FINAL RELEASE.IT IS INAPPROPRIATE TO USE FPL WORKING DRAFTS AS REFERENCE MATERIAL OR TO CITE THEM AS OTHER THAN “WORKS IN PROGRESS”.THE FPL GLOBAL TECHNICAL COMMITTEE WILL ISSUE, UPON COMPLETION OF REVIEW AND RATIFICATION, AN OFFICIAL STATUS ("APPROVED") TO THE PROPOSAL AND A RELEASE NUMBER.

Table of Contents

Document History

Date	Change	Who	Version
21 February 2013	Initial Version	Yuval Cohen, Etrading Software	0.1
15 March 2013	Updates: Message Structure diagram + Workflows diagram Merged with Lisa T.'s comments	Yuval Cohen, Etrading Software	0.2
19 March 2013	Minor corrections		0.3
11 April 2013	Update following Lisa T. additional comments	Yuval Cohen, Etrading Software	0.4
22 April 2013	Changed: Added note of filter criterisa to WF5 Added: Notes to WF10 WF11 Completed WF11 Clean document post few reviewers comments	Yuval Cohen, Etrading Software	0.5
29 April 2013	Added overview diagram	Yuval Cohen, Etrading Software	0.6
31 July 2013	Improve the introduction section according to FPL recommendations	Yuval Cohen, Etrading Software	1.0
14 August 2013	Incorporate FPL new tags	Yuval Cohen, Etrading Software	1.1

Executive Summary

Background

FIX Protocol

FIX Protocol Ltd (FPL) is the non-profit, industry-driven standards body at the heart of global trading. FPL is independent and neutral, dedicated to addressing the real business and regulatory issues impacting multi-asset trading in global markets through standardisation, delivering operational efficiency, increased transparency, and reduced costs and risks for all market participants.

Central to FPL's work is the continuous development and promotion of the FIX Protocol messaging language, which has revolutionised the electronic trading environment and has successfully become the way the world trades.

The FIX Protocol specification is a comprehensive description of a set of messages and fields that can be exchanged between trading parties in order to trade electronically. Much work has gone into the specification to ensure that it covers a wide set of scenarios, including specific work to cover fixed income trading flows.

In the nearly 20 years of the FIX Protocol standard, the protocol has been synonymous with equities electronic trading and has become the de facto standard used by equities trading systems globally. FIX has evolved over those years to continually support the needs of the global user community across different user groups and several different asset classes.

Reasons for the Creation of the Recommended Practices Document

Currently, Fixed Income client enablement – the process by which sell-side firms enable clients to trade with them on ECNs – is a manual and labour-intensive process. While a lot of resources and focus have gone into optimising and standardising trading workflows, the area of client enablement has attracted very little interest from market participants so far. As a result, each ECN has been left to provide its own solution for client enablement resulting in dealers having to implement labour intensive, ad-hoc procedures to get this vital step right.

In January 2013, the Trading Enablement Standardisation Initiative (TESI) was launched by the global investment banking community to engage with execution venues and other stakeholders to promote the FIX Protocol for client and trader enablement on electronic trading platforms.

Thanks to extension packs 105, 128, 129 and 146 which were added to FIX 5.0 SP2, the FIX Protocol provides a rich set of messages (Party Reference Data) for market participants to exchange enablement information. These extension packs were originally designed by listed derivatives exchanges and brokers for communicating enablement information.

This Recommended Practices Document defines a standard set of guidelines based on Party Reference Data messages for exchanging enablement information for trading Fixed Income OTC instruments.

These guidelines, once implemented, are expected to bring the following benefits to the stakeholders:

- Faster client enablement
- Increased operational efficiency
- Reduction in STP failures
- Help meet more stringent operational regulatory requirements

As Dodd-Frank in the US and MiFID II in Europe push OTC derivatives trading onto Swap Execution Facilities (SEFs) and Organised Trading Facilities (OTFs) respectively, these Recommended Practices will help ensure that these trading venues can adopt the FIX Protocol for enablement in a consistent and standardised manner. This standardardised entitlement protocol increases market efficiency by allowing market participants to communicate and complete the entitlement process quickly and cost efficiently.

Work of the FPL Global Fixed Income Committee

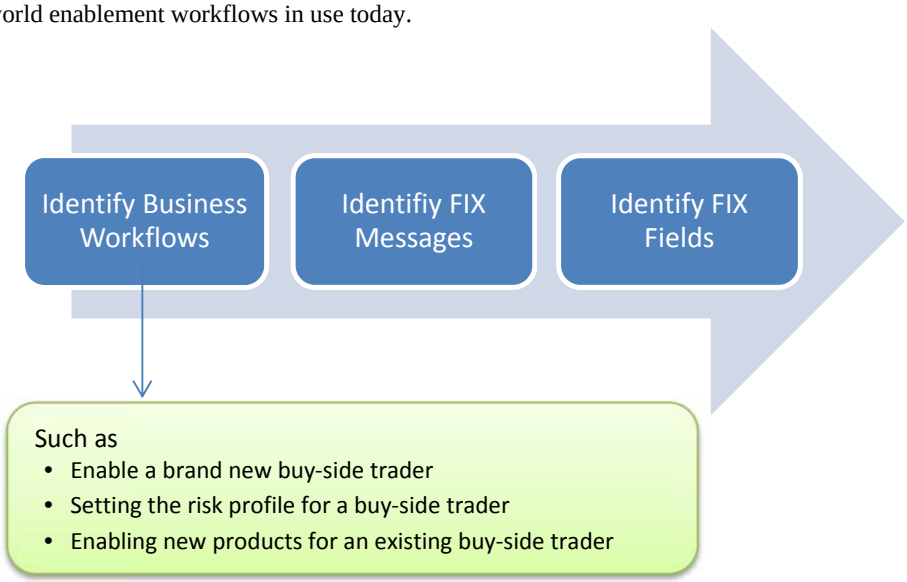
In February 2013, FPL launched an initiative to create the Fixed Income recommended practices for party entitlement for the OTC markets in response to a request by a group of 10 global Fixed Income dealers (Trading Enablement Standardisation Initiative for Fixed Income [TESI-FI]).

The initial focus was to standardise the OTC fixed income instrument entitlement process by creating an industry agreed set of recommended FIX practices. These documents were compiled by FPL GFIC between February and June of 2013.

Recommended Practices Guidelines

Principles behind the Recommended Practices

The primary focus of the recommended practices is to provide detailed guidelines for the implementation of party entitlement messages for Fixed Income OTC electronic markets. The committee members, comprising of the majority of the global Fixed Income dealers and Fixed Income trading venues, based their recommendations on real world enablement workflows in use today.



The starting point for each scenario was identifying the business workflow and then recommending the set of FIX messages and fields to use for that workflow.

The approach resulted in several gaps being identified in the FIX Protocol for fixed income trading.
The committee submitted relevant gap analysis documents to the FPL Global Technical Committee (GTC) for enhancements to the FIX Protocol and these were all approved and incorporated into the FIX 5.0 SP2 specification. This approach has ensured that this Recommended Practices document is a practical FIX implementation guide that can be used immediately by all the major Fixed Income execution venues.

Overview of Recommendations

The recommended practices for party entitlement for fixed income OTC markets contain the following sections:

- Objective
- Scope (In scope and out of scope)
- Market Conventions
- Diagrams: overview diagram, activity diagram and data model diagram
- Workflows
- Message details
- Entitlement attributes

1 Objective

The objective of this document is to define an agreed sub-set of permissible message implementations and, in doing so, provide a definitive statement as to how these messages should be implemented in practice to support Dealer to Clients (D2C) Fixed Income (FI) party entitlements workflows.

2 Scope

This section will set out what is in scope for the entitlement messaging requirements as well as highlighting what is not in scope (although this list will not be exhaustive).

2.1 In Scope

- D2C Execution Venues in the Fixed Income domain, including SEFs and OTFs
- Required workflows for managing buy-side user entitlements by sell-side firms trading through electronic D2C execution venues
- Ability to set entitlements for one or more buy-side users. Such entitlements pertain to either trading permissions or price viewing permissions, or both

2.2 Out of Scope

- Communication between execution venues and buy-side users
- Entitlement or on-boarding of clearing or settlement firms
- Direct communication between sell-side and buy-side (i.e. not routed through the execution venue)
- The preliminary process by which a sell-side firm may initially view and identify a new buy-side firm or specific buy-side user
- Entitlements on D2D Execution Venues

3 Target Audience

Parties interested in implementing the FIX Protocol for exchanging fixed income parties reference data between the sell-side and D2C Execution Venue.

4 Authors

Documents created by the FPL Global Fixed Income Committee with assistance from Etrading Software. To learn more about this committee please visit: www.fixprotocol.org/committees/gfic

5 Market Conventions

5.1 Fixed Income Market Structure

Today, in the fixed income markets, brokers execute a proportion of transactions with their buy-side clients through electronic venues. These electronic venues are known as Dealer to Client (D2C) venues.

In addition, to ensure that a dealer can service its clients' orders, it will execute transactions on Dealer to Dealer (D2D) venues between brokers to facilitate liquidity (sometimes referred to as inter-dealer activity).

At each electronic trading venue, the broker has to enable trading permissions for all their counterparties and their respective traders. This process of entitlement management is currently conducted manually through graphical user interfaces provided by each of the venues, each different from the other, as shown in Figure 1. The arrows indicate the information flow required to setup, monitor and maintain trading permissions in the market:

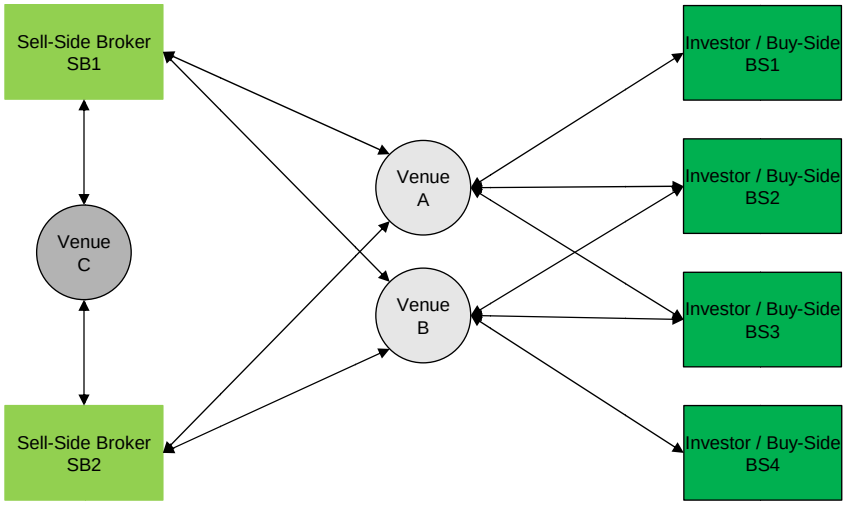


Figure 1: Entitlement Information Flow

In , venues A and B are Dealer to Client (D2C) and venue C is a Dealer to Dealer (D2D) venue.

5.2 Terminology

The terminology used commonly in the industry to describe the entities and attributes associated with trader enablement differs from that used in the FIX messaging standard. This section provides the mapping between the two by indicating the FIX tags and values and how they relate to the various entities in the data model.

5.2.1 Sell-Side Terminology

This section describes how the terminology relating to the sell-side maps onto the FIX messaging standard. Each subsection defines a single attribute and indicates the FIX tags and values necessary to fully specify this sell-side attribute

5.2.1.1 Sell-Side Identifier

Sell-side firms usually operate as multiple entities. Such entities may represent different branches (e.g. divided by geographical location), legal entities, departments (e.g. investment bank vs. retail bank), and so on. During the entitlement process, it is necessary to define which sell-side entity is entitled for the relationship. Usually, one or more people at the sell-side may be authorized to retrieve and/or modify entitlements for multiple entities.

5.2.1.2 Sales people

Each entitlement is usually associated with a primary sales person from the sell-side firm, and optionally a list of additional sales people. Each Sales person may have an ID at the Execution Venue as well as name and possibly contact number.

5.2.2 Buy-Side Terminology

This section describes how the terminology relating to the buy-side maps onto the FIX messaging standard. Each subsection defines a single attribute and indicates the FIX tags and values necessary to fully specify this buy-side attribute.

5.2.2.1 Buy-Side User ID

The buy-side user id is used to identify the buy-side user within party reference data messages. Buy-side user ID is set in the PartyDetailGrp or the RelatedPartyDetailGrp.
For simplicity the following sections assumes that buy-side attributes are populated in the PartyDetailGrp

5.2.2.2 Legal Entity Identifier

The legal entity identifier (LEI) is required by the CFTC regulations. In the FIX standard it is specified using the three tags PartyDetailID, PartyDetailIDSource and PartyDetailRole as follows:

Tag	Field Name	Value	Comment
1691	PartyDetailID	<LEI>	*Legal Entity Identifier*
1692	PartyDetailIDSource	N	Source of the identifier (N=LEI)
1693	PartyDetailRole	3	Identifies the type or role (13=Order Origination Firm)

5.2.2.3 Client Account

The client account is a trading account that is set by a sell-side firm in order to identify a buy-side account. Some buy-side entities may have multiple client accounts. In the FIX standard it is specified using the three tags PartyDetailID, PartyDetailIDSource and PartyDetailRole as follows:

Tag	Field Name	Value	Comment
1691	PartyDetailID	<Account ID>	*Account ID*
1692	PartyDetailIDSource	D	Source of the identifier (D=Proprietary)
1693	PartyDetailRole	24	Identifies the type or role (24=Customer Account)

5.2.2.4 Additional Buy-Side Attributes

Further attributes defining properties of the buy-side user may be defined using the FIX tags PartyDetailID, PartyDetailIDSource and PartyDetailRole repeating group. Using these tags it is possible to provide values for the buy-side user name, firm, department, contact telephone number, email address and postal address, as shown below:

Tag	Field Name	Value	Comment
1691	PartyDetailID	<value>	Sub-identifier for the party
1692	PartyDetailIDSource	9=Contact name 1=Firm 24=Department 7=Phone number 8=Email address 6=Postal address	Type of PartyDetailSubID value

5.2.3 Product/Sector Terminology

This section describes how the terminology relating to products and sectors maps onto the FIX messaging standard. Each subsection defines a single attribute and indicates the FIX tags and values necessary to fully specify this product/sector attribute.

5.2.3.1 Product

Most Fixed Income Execution Venues use the term Product to refer to MarketID(1301), e.g. most Execution Venues consider IRS, CDS and US-Treasury-Bonds as three different products.

5.2.3.2 Sector

Most Fixed Income Execution Venues use the term sector to refer to MarketSegmentID(1300). This tag is used in conjunction with MarketID(1301) in order to specify the sub classification.

5.2.3.3 Tier

The tier level is used to determine the prices that the sell-side firm offers to the buy-side users. The FIX tag MDStreamID(1500) is used to identify the price stream.

5.3 Entitlement convention

See Appendix I for the list of attributes used for entitlements:

5.3.1 Scope the Entitlement by trading feature

Some venues support the ability to scope entitlements by trading feature. For example, a buy-side user may be entitled to all/part/none of the following trading features:

- Buy-side user entitled to send list trades
- Buy-side user entitled for ‘click to trade’ workflows
- Buy-side user entitled to send specific order types (e.g. limit order)

- Buy-side user entitled to send single dealer quote requests
- Buy-side entitled to send multi-dealers quote requests
- Buy-side user entitled to view prices of (and/or trade) instruments that are cleared in a specific clearing house
- Buy-side user entitled to view prices of (and/or trade): multi-leg instruments

5.4 Typical scenarios

Below are some of the most commonly used scenarios and the FIX messages to implement them.

5.4.1 Sell-side entitles a new user at an existing buy-side

When a sell-side entitles another buy-side user at a buy-side firm for which some users have already been entitled; sell-side sends PartyEntitlementsDefinitionRequest(35=DA) to the Execution Venue and the venue is expected to reply with PartyEntitlementsDefinitionRequestAck(35=DB). This workflow is covered in WF7 - Sell-Side Modifies Multiple Entitlements – Venue Accepts

5.4.2 Sell-Side Modifies Sales Coverage

When sell-side wishes to modify the sales coverage for different entitlements, the sell-side should send a PartyEntitlementsDefinitionRequest(35=DA) having the new sales people as part of the PartysDetails FIX component.). This workflow is covered in WF7 - Sell-Side Modifies Multiple Entitlements – Venue Accepts

5.4.3 A Buy-Side User makes a new request for entitlement

This scenario is covered in WF10 - Buy-Side Requests a New Entitlement for one or more Users - Approved

5.4.4 Buy-side user leaves

When the Execution Venue is aware that a buy-side user left its firm, the Execution Venue sends a PartyDetailsListUpdateReport(35=CK) having ListUpdateAction(1324)=Delete(D). This workflow is covered in WF4 – Sell-Side Retrieves Snapshot + Updates of Buy-Side Entities/Users

5.4.5 Sell-Side Deletes an Entitlement

When the sell-side wishes to disable a buy-side user from trading, the sell-side should send a PartyEntitlementsDefinitionRequest(35=DA) message having EntitlementIndicator(1774)=N for that entitlement. This workflow is covered in WF7 - Sell-Side Modifies Multiple Entitlements – Venue Accepts

6 System Diagrams

6.1 Overview Diagram

The following diagram illustrates the FIX messages and the Workflows described in this chapter.

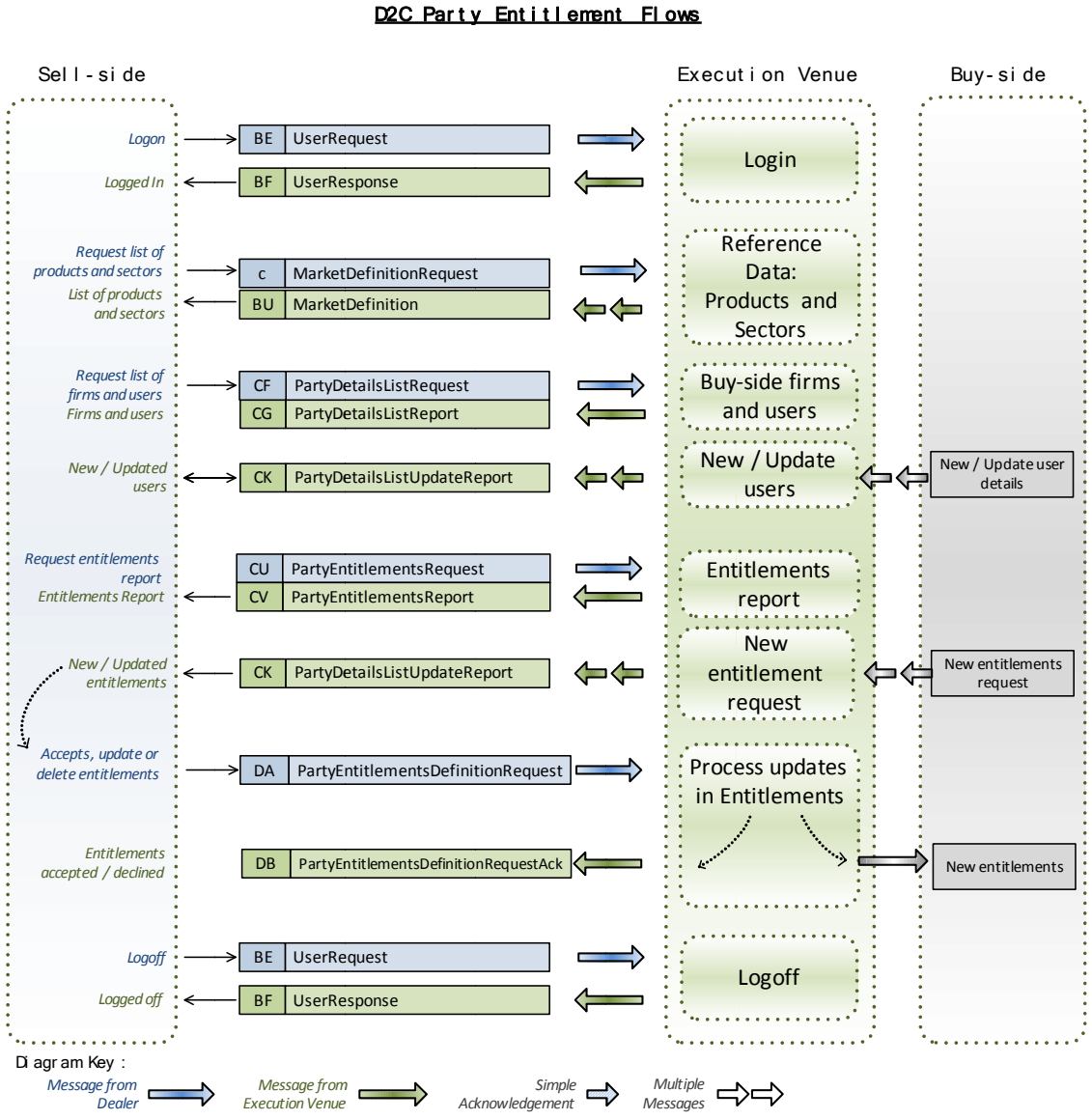


Figure 2: Overview Diagram

6.2 Activity Diagram

The activity diagram illustrates the FIX messages that will be used to manage Party Entitlements. The use of each message is explained in the context of the workflows later in this document in Chapter 5.

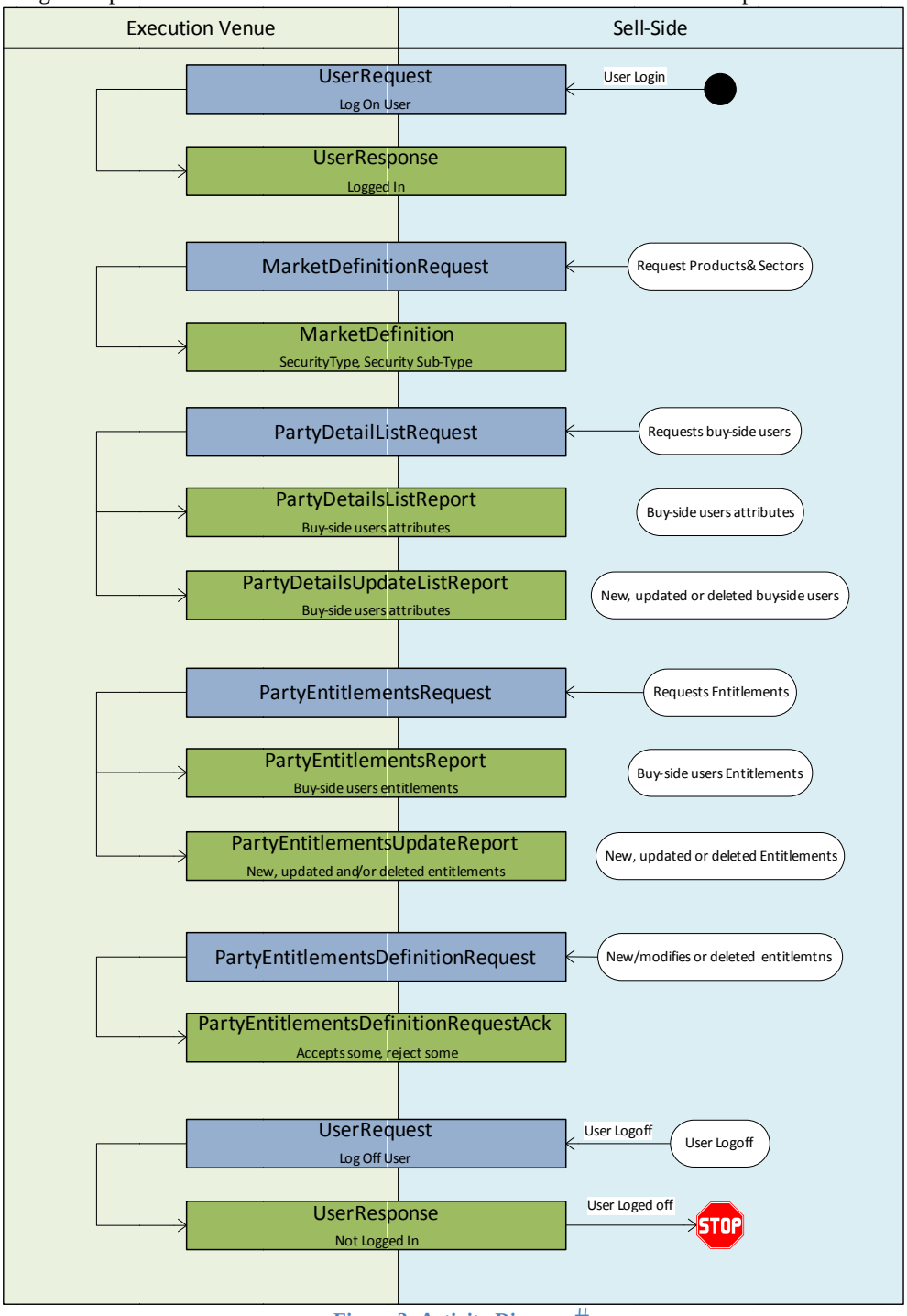


Figure 3: Activity Diagram#

6.3 Proposed Entitlements Data Model Diagram

Chapter 3, Market Conventions, described the various components used to describe Party Entitlements. The following diagram illustrates how they are organised within the data model comprising:

1. Sell-side Firm, with sell-side entity, its sales people and the private client account codes it uses to describe its clients
2. Execution venue and the instruments it trades described as products and sectors
3. Buy-side firm and the individual users that the sell-side trades with
4. Parties entitlement describing the relationship between the sell-side, buy-side and venue on which the transaction takes place

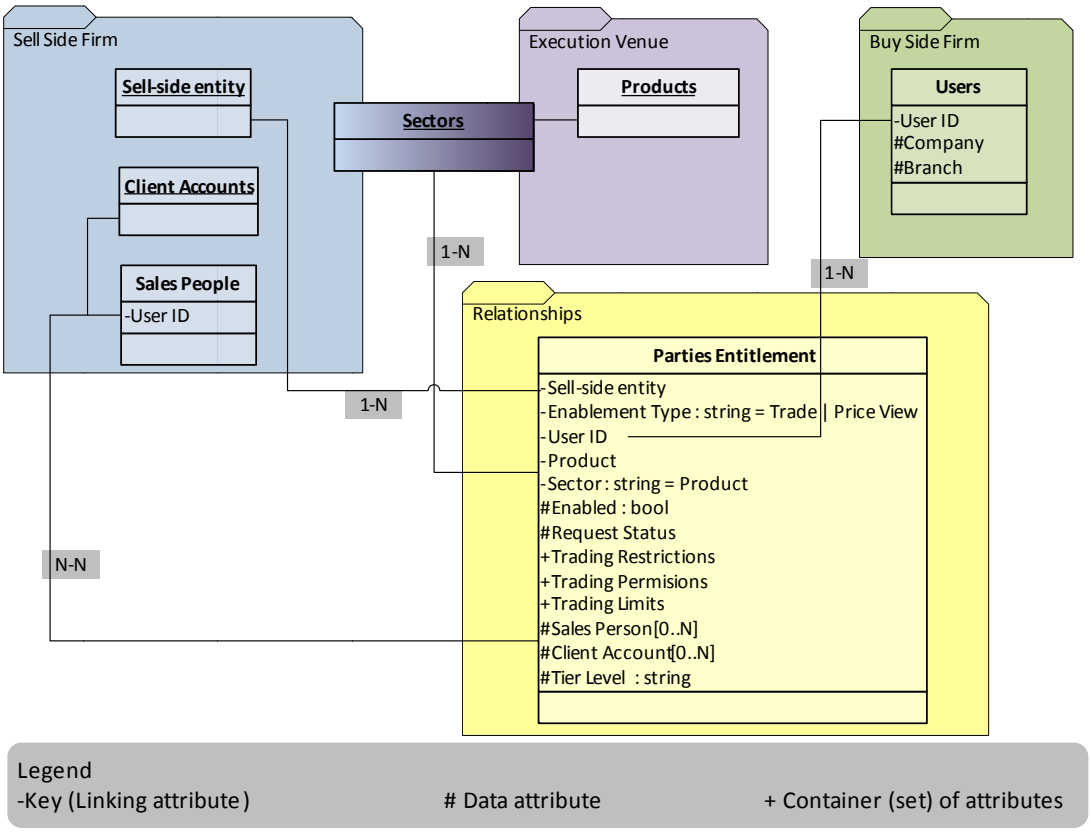


Figure 4: Entitlements Data Model Diagram#

7 Workflows

7.1 Message Flows Summary

This section details the various scenarios identified as requirements to be supported. Each scenario is described at a high level followed by the FIX messages used to support the scenario.

For a given implementation of these scenarios, Execution Venues need to take into consideration having the proper controls in place to safeguard the information and determining whether the requester is permissioned to see or retrieve the information. Further discussions on this topic are out of scope of this document.

Scenario
WF1-A - Sell-Side User Login
WF1-B - Sell-Side User Logoff
WF2 - Sell-Side Requests Products and Sectors
WF3 - Sell-Side Requests a list of Buy-Side Entities/Users
WF4 – Sell-Side Retrieves Snapshot + Updates of Buy-Side Entities/Users
WF5 - Sell-Side Requests Entitlement Report of Buy-Side Users - Snapshot
WF6 - Sell-Side Requests Entitlement Report of Buy-Side Users – Snapshot + Updates
WF7 - Sell-Side Modifies Multiple Entitlements – Venue Accepts
WF8 - Sell-Side Modifies Multiple Entitlements – Venue Accepts some Entitlements and Rejects other Entitlements
WF9 - Sell-Side Modifies Multiple Entitlements – Venue Rejects
WF10 - Buy-Side Requests a New Entitlement for one or more Users - Approved
WF11 - Buy-Side Requests Entitlement for one or more Users - Rejected

7.2 WF1-A - Sell-Side User Login

This scenario illustrates the case where an authorized sell-side user sends a login message to authenticate with the Execution Venue system and gains permission for further entitlement activities.

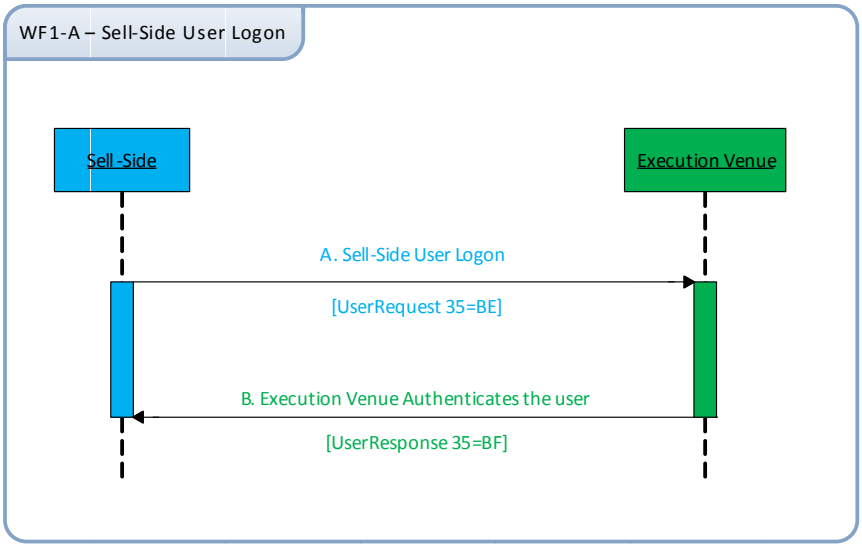


Figure 5:WF1-A - Sell-Side User Login

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Sell-Side User Logon	Sell-Side	→	**BE – UserRequest** UserReqID(923)=❶ UserReqType(924)=Log On User(1) Username(553)=<required> Password(554)=<optional>	Execution Venue
(B) Execution Venue Authenticates the user		←	**BF – UserResponse** UserReqID(923)=❶ Username(553)=<required> UserStatus(926)=Logged In(1)	

Table 1: WF1-A - Sell-Side User Login

Notes

- In subsequent messages the Username will be contained in the field: RequestingPartyID(1658), for example in:
  - PartyDetailsListRequest(35=CF)
  - PartyEntitlementsDefinitionRequest(35=DA)
- Workflows to change user password and request individual user status are implemented using very similar workflows and messages

7.3 WF1-B - Sell-Side User Logoff

This scenario illustrates the case where a sell-side user sends a logoff message to the Execution Venue system in order to terminate the session.

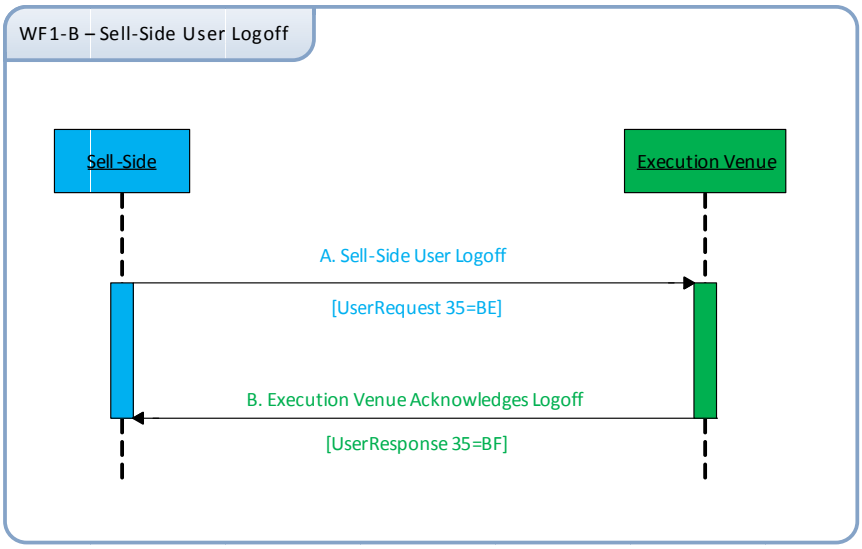


Figure 6:WF1-B - Sell-Side User Logoff

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Sell-Side User Logoff	Sell-Side	→	**BE – UserRequest** UserReqID(923)=❶ UserReqType(924)=Log Off User(2) Username(553)=<required> Password(554)=<optional>	Execution Venue
(B) Execution Venue Acknowledges Logoff		←	BF – UserResponse UserReqID(923)=❶ Username(553)=<required> UserStatus(926)=Not Logged In(2)	

Table 2: WF1-B - Sell-Side User Logoff

Notes

- During Logoff, the UserName is the same value of a user who has previously logged-in.
- After a user has logged-off, subsequent messages PartyDetailsListRequest(35=CF) and PartyEntitlementsDefinitionRequest(35=DA), having the same UserName in field RequestingPartyID(1658) should be rejected by the Execution Venue.

7.4 WF2 - Sell-Side Requests Products and Sectors

This scenario illustrates the case where a sell-side requests market segmentation information from the Execution Venue. The sell-side requests the list of products and sectors by sending a request message. The Execution Venue replies with a series of product / sector definition messages containing the list of products and sectors. The request may be filtered to receive sectors of a single product.

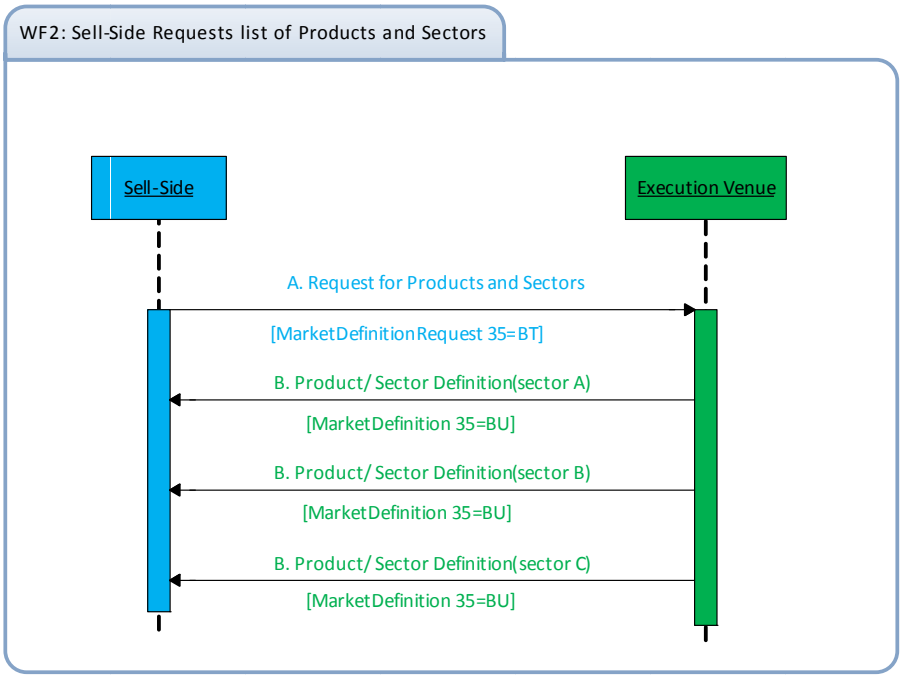


Figure 7: WF2 - Sell-Side Requests Products and Sectors

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Sell-Side Requests for Products and Sectors	Sell-Side	→	**BT – MarketDefinitionRequest** MarketReqID(1393)=❶ SubscriptionRequestType(263)=snapshot(0)	Execution Venue
(B) Product / Sector List	Sell-Side	← ← ←	**BU – MarketDefinition** MarketReportID(1394)=❷ MarketReqID(1393)=❶ MarketID(1301)=<required> MarketSegmentID(1300)	Execution Venue

Table 3: WF2 - Sell-Side Requests Products and Sectors

Notes

- Multiple Market Definition messages may be received – each one containing a different market and optional market segment

7.5 WF3 - Sell-Side Requests a Snapshot of Buy-Side Entities/Users

This scenario illustrates the case where the sell-side sends a request for buy-side user information to the Execution Venue. A request message may contain filtered attributes to retrieve data for a single buy-side company, a single branch within a buy-side company or a single product or sector. The Execution Venue replies by sending a PartyDetailListReport message containing the details of the requested buy-side users. The requesting message returned by the Execution Venue may be fragmented across multiple messages if the result set is large.

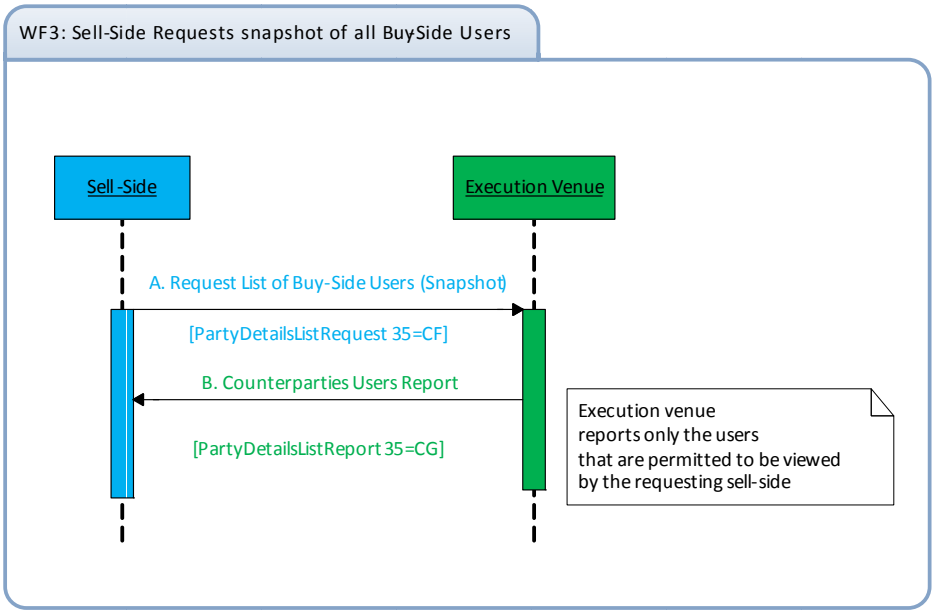


Figure 8: WF3 - Sell-Side Requests a list of Buy-Side Entities/Users

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Sell-Side Requests for Buy-side Users (Snapshot)	Sell-Side	→	**CF – PartyDetailsListRequest** PartyDetailsListRequestID(1505) =❶ SubscriptionRequestType(263)=Snapshot(0)	Execution Venue
(B) Execution Venue sends results set		←	**CG – PartyDetailsListReport** PartyDetailsListRequestID(1505) =❶ PartyDetailsListReportID (1510)= <required> RequestResult(1511)=Valid Request(0) NoPartyDetails(1671)=N (number of users in this message) -> PartyDetailID(1691) = <required> -> PartyDetailIDSource(1692) -> PartyDetailRole(1693) -> PartyDetailSubGrp	

Table 4: WF3 - Sell-Side Requests a list of Buy-Side Entities/Users

Notes

- **As a result of this work, FPL added the following enumerations:**
  - **PartyDetailRole(1693) = Sales Person(117)**
  - **PartyDetailRole(1693) = Sales Operator(118)**
- The result set may be fragmented across multiple PartyDetailsListReport messages by setting the TotNoParties(1512) and LastFragment(893) fields.

- Each buy-side user should have a PartyDetailGrp containing all the attributes that are associated with this user
- It is currently impossible to filter the request by products or sectors (i.e. request all users that are associated with a single security type)
- Requests may include attributes to filter the response by The requested party role (e.g. a request may be sent for all users of a single buy-side firm or a single branch within the buy-side firm)

7.6 WF4 – Sell-Side Requests Snapshot + Updates of Buy-Side Entities/Users

This scenario illustrates the case where the sell-side sends a request for buy-side user information with updates to the Execution Venue. A request message is sent, which may contain data filtering attributes to retrieve data for all users or a single buy-side company or a single branch within a buy-side company or a single product. The Execution Venue responds with a report message containing the details of the requested buy-side users. The result set returned by the Execution Venue may be fragmented across multiple messages. The Execution Venue continues to send updates when:

- A new buy-side user is added to their system
- There is a change to an existing buy-side user
- A buy-side user is deleted or removed from their system (i.e. "trader leaves")

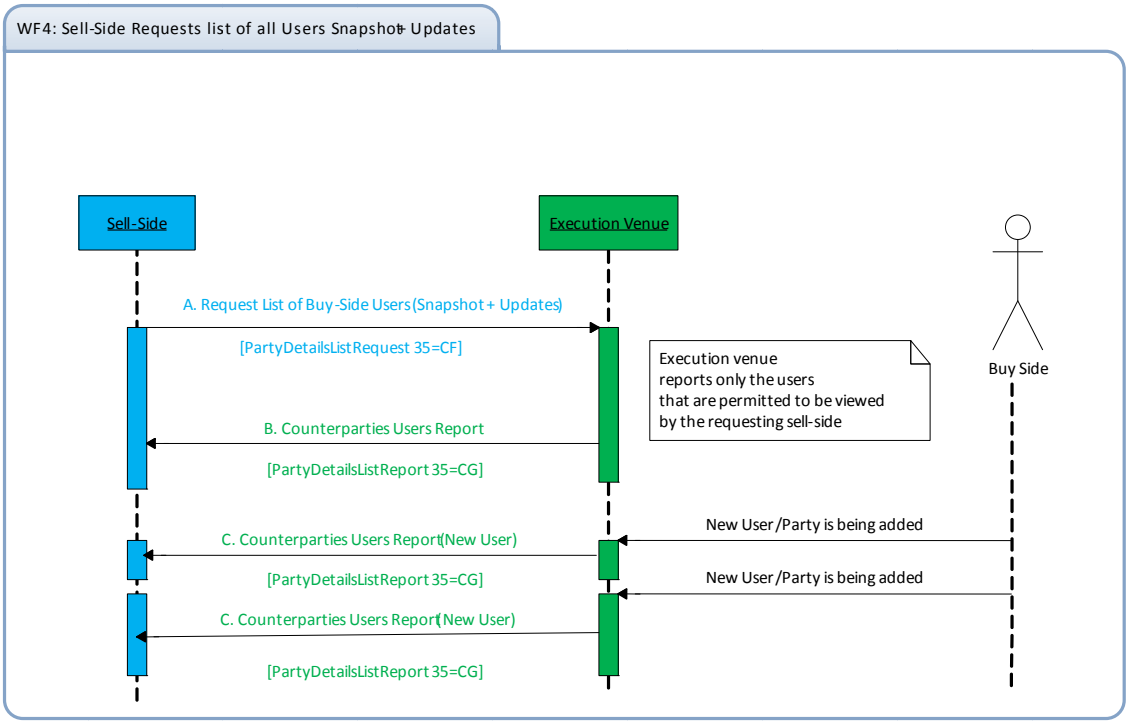


Figure 9: WF4 – Sell-Side Retrieves Snapshot + Updates of Buy-Side Entities/Users

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Sell-Side Requests for Buy-side Users	Sell-Side	→	**CF – PartyDetailsListRequest** PartyDetailsListRequestID(1505) =❶ SubscriptionRequestType(263)=Snapshot+Updates(1)	Execution Venue
(B) Counterparties User Report (Snapshot)		←	**CG – PartyDetailsListReport** PartyDetailsListRequestID(1505) =❶ PartyDetailsListReportID (1510)=❷ RequestResult(1511)=Valid Request(0) NoPartyDetails(1671)=N (number of users in this message) -> PartyDetailID(1691) = ❸ -> PartyDetailIDSource(1692) -> PartyDetailRole(1693) -> PartyDetailSubGrp	
(C) Counterparties New User Report		← ← ←	**CK – PartyDetailsListUpdateReport** PartyDetailsListRequestID(1505) =❶ PartyDetailsListReportID (1510)=❹ NoPartyUpdates (1676)=N (number of changes) -> ListUpdateAction = Add/Delete/Modify -> NoPartyDetails(1671)=N (number of users in this message) -> -> PartyDetailID(1691) = ❺ -> -> PartyDetailIDSource(1692) -> -> PartyDetailRole(1693) -> -> PartyDetailSubGrp	

Table 5: WF4 – Sell-Side Retrieves Snapshot + Updates of Buy-Side Entities/Users

Notes

- See scenario WF3 notes

7.7 WF5 - Sell-Side Requests Entitlement Report of Buy-Side Users - Snapshot

This scenario illustrates the case where the sell-side sends a request for the list of buy-side users and their entitlement status in the form of party entitlement request. The request message may include filtering criteria to retrieve entitlement data, for example, for all buy-side users, a single buy-side company/branch, a specified product or sector, or a specified entitlement status. The Execution Venue responds by returning a report message containing the result set that met the request criteria.

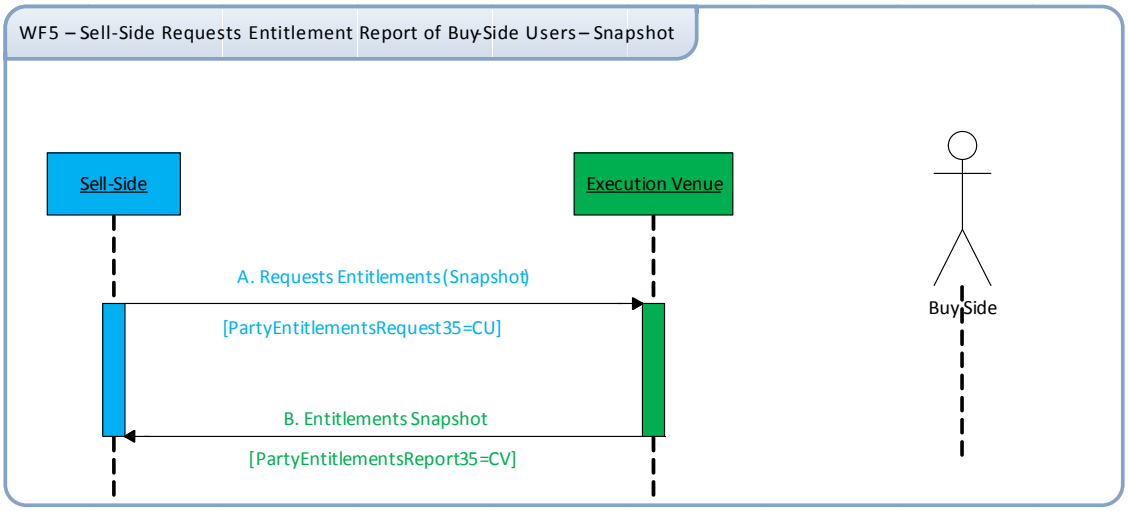


Figure 10: WF5 - Sell-Side Requests Entitlement Report of Buy-Side Users - Snapshot

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Entitlements report request	Sell-Side	→	**CU – PartyEntitlementsRequest** EntitlementRequestID(1770)=❶ SubscriptionRequestType(263)=Snapshot(0) PartyDetailStatus(1672) <optional> MarketSegmentScopeGrp <optional> NoEntitlementTypes(2345) <optional> -> EntitlementType(1775)	Execution Venue
(B) Entitlements Report (Snapshot)		←	**CV – PartyEntitlementsReport** EntitlementRequestID(1770)= ❶ EntitlementReportID(1771)= <required> RequestResult(1511)= Valid Request(0) NoPartyEntitlements(1772)=N -> NoPartyDetails(1671)=M -> PartyDetailGrp -> EntitlementGrp	

Table 6: WF5 - Sell-Side Requests Entitlement Report of Buy-Side Users - Snapshot

Notes

- The result set may be fragmented across multiple PartyEntitlementsReport messages by setting the TotNoParties(1512) and LastFragment(893) fields
- When a sell side sends a PartyDetailsListRequest or a PartyEntitlementsRequest he may wish to add a criteria to filter the results set by:

  - the value of PartyDetailStatus; e.g. set criteria to filter the results set to only active (or pending) parties
  - the market segment (i.e. product/sector); e.g. set criteria to filter the results set to only IRS
  - one or more values of EntitlementType; e.g. retrieve the entitlements to Trade and to Subscribe to market data; In such a case, the PartyEntitlementReport(35=CU) message contains NoEntitlementType(2345)=2 followed by EntitlementType(1775)=Trade(0) and EntitlementType(1775)= Subscribe to market data(5)
- See examples request messages in: Party Entitlements Request (35=CU)
- See an example report message in: Party Entitlement Report FIXML

7.8 WF6 - Sell-Side Requests Entitlement Report of Buy-Side Users – Snapshot + Updates

This scenario illustrates the case where the sell-side subscribes for the list of buy-side users and their entitlements status in the form of a party entitlement request. The Execution Venue responds by returning a PartyEntitlementReport containing the required parties, their current entitlements. The Venue follows with messages on any addition, deletion or modification of entitlements. Addition and deletion are usually a result of buy-side requests whilst the modifications which usually made by the sell-side (by sending PartyEntitlementsDefinitionRequests message) are sent back to the sell-side in PartyEntitlementsReport message (having ListUpdateAction(1324)=Modify(M)).;
It is possible to have new entitlements, modified entitlements and deleted entitlements in the same update message.

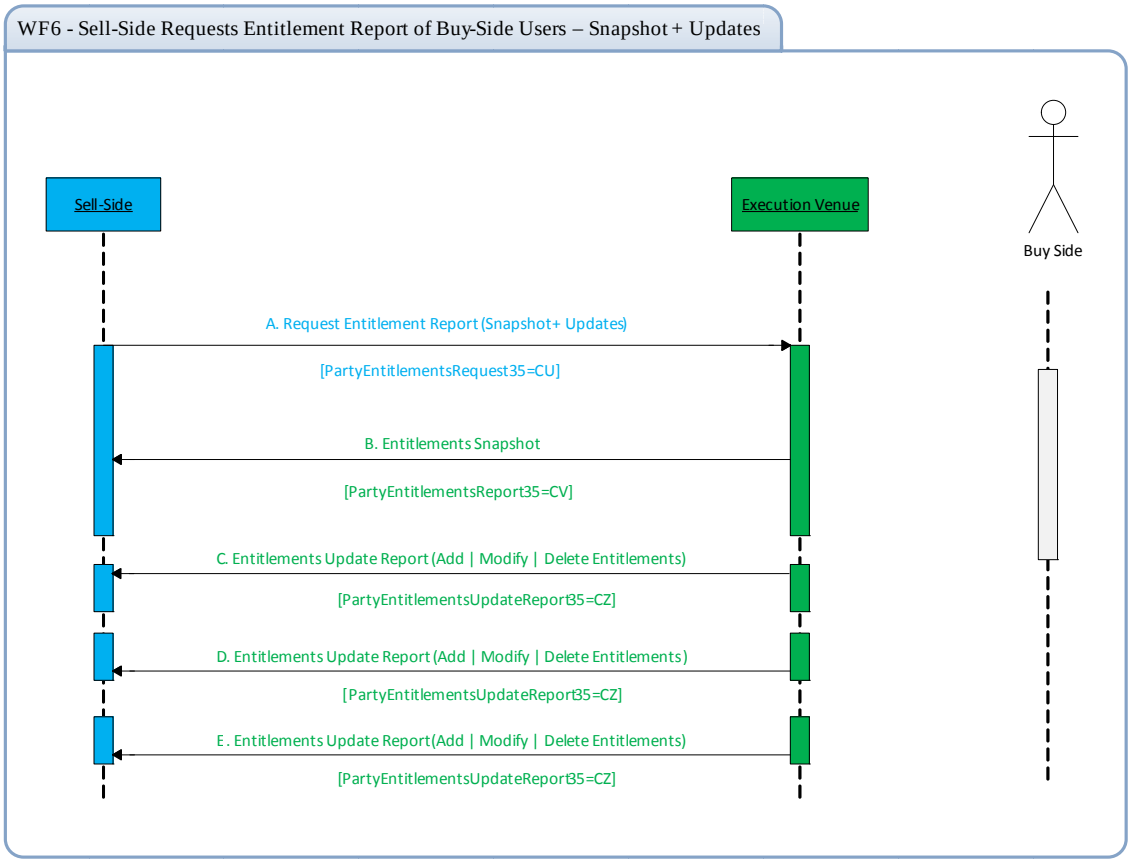


Figure 11: WF6 - Sell-Side Requests Entitlement Report of Buy-Side Users – Snapshot + Updates

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Entitlements report request	Sell-Side	→	**CU – PartyEntitlementsRequest** EntitlementRequestID(1770)=❶ SubscriptionRequestType(263)=Snapshot + Updates(1) PartyDetailStatus(1672) <optional> MarketSegmentScopeGrp <optional> NoEntitlementType(2345) <optional> -> EntitlementType(1775)	Execution Venue

Model FIX 5.0			
(B) Entitlements Report (Snapshot)		←	**CV – PartyEntitlementsReport** EntitlementRequestID(1770)= ❶ EntitlementReportID(1771)= ❷ RequestResult(1511)= Valid Request(0) NoPartyEntitlements(1772)=N -> NoPartyDetails(1671)=M -> PartyDetailGrp -> EntitlementGrp
(C) New Entitlement(s) (Update report)		← ← ←	**CZ – PartyEntitlementsUpdateReport** EntitlementRequestID(1770)=❶ EntitlementReportID(1771)= ❸ NoPartyEntitments(1772)=N -> ListUpdateAction(1324)=Add(A) -> PartyDetailGrp -> EntitlementGrp
(D) Modified Entitlement(s) (Update report)		← ← ←	**CZ – PartyEntitlementsUpdateReport** EntitlementRequestID(1770)=❶ EntitlementReportID(1771)= ❹ NoPartyEntitments(1772)=N -> ListUpdateAction(1324)=Modify(M) -> PartyDetailGrp -> EntitlementGrp
(E) Delete Entitlement(s) (Update report)		← ← ←	**CZ – PartyEntitlementsUpdateReport** EntitlementRequestID(1770)=❶ EntitlementReportID(1771)= ❺ NoPartyEntitments(1772)=N -> ListUpdateAction(1324)=Delete(D) -> PartyDetailGrp -> EntitlementGrp

Table 7: WF6 - Sell-Side Requests Entitlement Report of Buy-Side Users – Snapshot + Updates

Notes

- See scenario WF5 notes
- Each PartyEntitlementReport and PartyEntitlementUpdateReport message may contain more than a single entitlement
- A single PartyEntitlementUpdateReport message may contain updates to add some new entitlements, modify some entitlements and delete some entitlements
- PartyEntitlementReport and PartyEntitlementUpdateReport messages can be fragmented

7.9 WF7 - Sell-Side Modifies Multiple Entitlements – Venue Accepts

This scenario illustrates the case where the sell-side modifies multiple entitlements:

- Some entitlements are enabled
- Other entitlements are disabled

All modifications may be sent in one or multiple messages. The Execution Venue accepts all the modification.

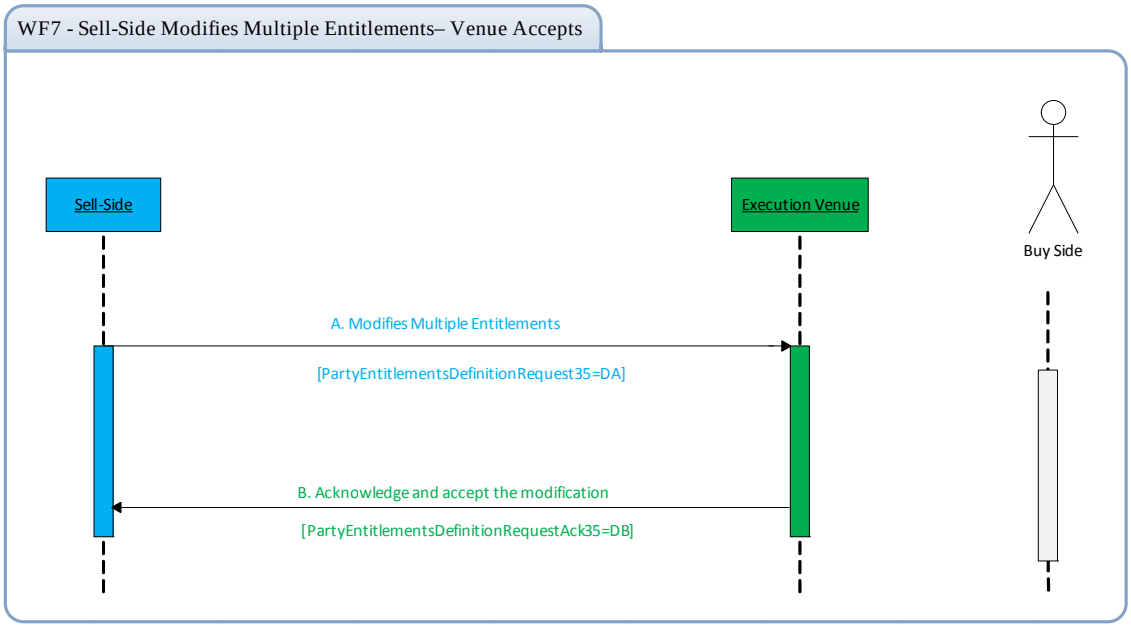


Figure 12: WF7 - Sell-Side Modifies Multiple Entitlements – Venue Accepts

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Sell-side Modifies multiple entitlements	Sell-Side	→	**DA – PartyEntitlementsDefinitionRequest** EntitlementRequestID(1770)=❶ RequestingPartyGrp NoPartyEntitlements(1772)=N -> ListUpdateAction(1324)=Modify(M) -> PartyDetailGrp -> NoEntitlements(1773)=M -> -> EntitlementIndicator(1774)=Y|N (i.e.: enabled | disabled) -> -> EntitlementID(1776)=<Identifying this entitlement>	Execution Venue
(B) Execution Venue acknowledges and accepts		←	**DB – PartyEntitlementsDefinitionRequestAck** EntitlementRequestID(1770)=❶ EntitlementRequestStatus(1882)=Accepted(0) EntitlementRequestResult(1881)=successful(0) (default)	

Table 8: WF7 - Sell-Side Modifies Multiple Entitlements – Venue Accepts

Notes

- Each PartyEntitlementDefinitionRequest may be used to modify multiple entitlements

- PartyEntitlementDefinitionRequest and PartyEntitlementDefinitionRequestAck messages can be fragmented i.e. split across multiple FIX messages

7.10 WF8 - Sell-Side Modifies Multiple Entitlements – Venue Accepts some Entitlements and Rejects other Entitlements

This scenario illustrates the case where the sell-side modifies multiple entitlements:

- Some entitlements are enabled
- Other entitlements are disabled

All modifications may be sent in one or multiple messages.
The Execution Venue

- Accepts some of the modifications
- Rejects the remainder of the modifications

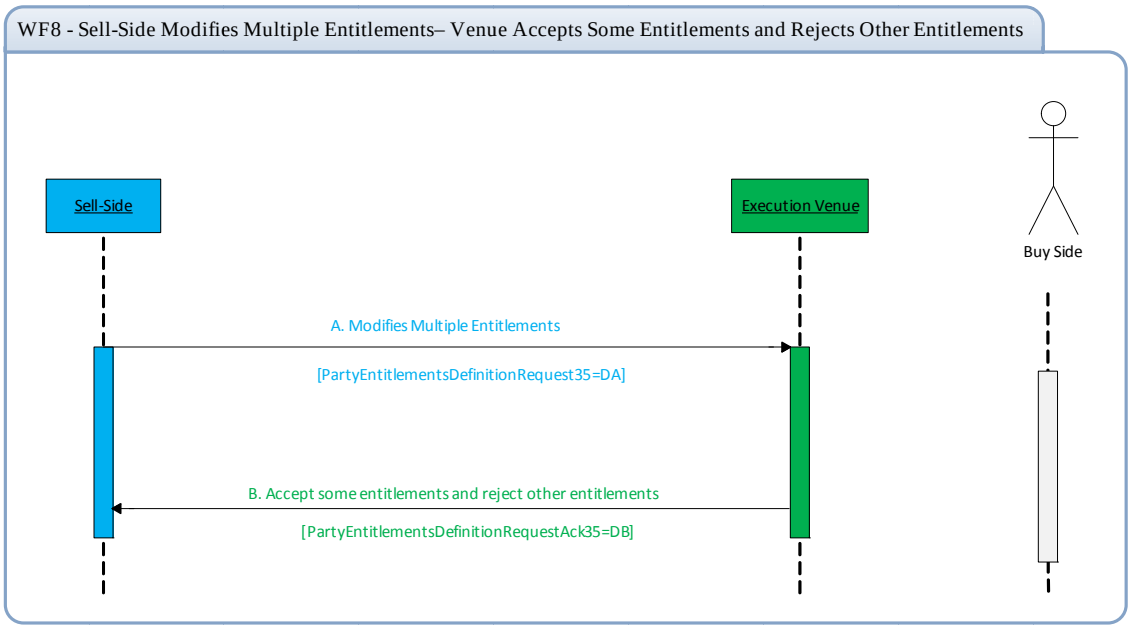


Figure 13: WF8 - Sell-Side Modifies Multiple Entitlements – Venue Accepts Some and Rejects Others

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2. In this example, the sell side modifies three entitlements for two different buy-side users. Two entitlements are accepted and the other entitlement is rejected:

Model FIX 5.0				
(A) Sell-side Modifies multiple entitlements	Sell-Side	→	**DA – PartyEntitlementsDefinitionRequest** EntitlementRequestID(1770)=❶ RequestingPartyGrp NoPartyEntitlements(1772)=3 -> ListUpdateAction(1324)=Modify(M) -> PartyDetailGrp -> -> NoPartyDetailID(1562) -> -> -> PartyDetailID(1563) = **User-1** -> NoEntitlements(1773)=1 -> -> EntitlementIndicator(1774)=**Y** (i.e. enabled) -> -> EntitlementType(1775)=**Subscribe to Market Data(5)** -> -> EntitlementID(1776)=❷ -> ListUpdateAction(1324)=Modify(M) -> PartyDetailGrp -> -> NoPartyDetailID(1562) -> -> -> PartyDetailID(1563) = **User-1** -> NoEntitlements(1773)=1 -> -> EntitlementIndicator(1774)=**Y** (i.e. enabled) -> -> EntitlementType(1775)=**Trade(0)** -> -> EntitlementID(1776)=❸ -> ListUpdateAction(1324)=Modify(M) -> PartyDetailGrp -> -> NoPartyDetailID(1562) -> -> -> PartyDetailID(1563) = **User-2** -> NoEntitlements(1773)=1 -> -> EntitlementIndicator(1774)=**Y** (i.e. enabled) -> -> EntitlementType(1775)= **Trade(0)** -> -> EntitlementID(1776)=❹	Execution Venue

Model FIX 5.0				
(B) Execution Venue accepts some entitlements and rejects the rest		←	**DB – PartyEntitlementsDefinitionRequestAck** EntitlementRequestID(1770)=❶ EntitlementRequestStatus(1882)=Accepted with changes(1) EntitlementRequestResult(1881)=successful(0) (default) RequestingPartyGrp NoPartyEntitlements(1772)=3 -> ListUpdateAction(1324)=Modify(M) -> EntitlementStatus(1883)= **Accepted (0)** -> EntitlementResult(1884)= **Successful (0)** -> PartyDetailGrp -> -> NoPartyDetailID(1562) -> -> -> PartyDetailID(1563) = **User-1** -> NoEntitlements(1773)=1 -> -> EntitlementIndicator(1774)=**Y** (i.e. enabled) -> -> EntitlementType(1775)= **Subscribe to Market Data(5)** -> -> EntitlementID(1776)=❷ -> ListUpdateAction(1324)=Modify(M) -> EntitlementStatus(1883)=**Rejected(2)** -> EntitlementResult(1884)=**<Required and non-zero>** -> PartyDetailGrp -> -> NoPartyDetailID(1562) -> -> -> PartyDetailID(1563) = **User-1** -> NoEntitlements(1773)=1 -> -> EntitlementIndicator(1774)=**Y** (i.e. enabled) -> -> EntitlementType(1775)=**Trade(0)** -> -> EntitlementID(1776)=❸ -> ListUpdateAction(1324)=Modify(M) -> EntitlementStatus(1883)=Accepted(0) -> EntitlementResult(1884)= Successful (0) -> PartyDetailGrp -> -> NoPartyDetailID(1562) -> -> -> PartyDetailID(1563) = **User-2** -> NoEntitlements(1773)=1 -> -> EntitlementIndicator(1774)=**Y** (i.e. enabled) -> -> EntitlementType(1775)= **Trade (0)** -> -> EntitlementID(1776)=❹	

Notes

- Each PartyEntitlementDefinitionRequest may be used to modify multiple entitlements
- PartyEntitlementDefinitionRequest and PartyEntitlementDefinitionRequestAck messages can be fragmented

7.11 WF9 - Sell-Side Modifies Multiple Entitlements – Venue Rejects

This scenario illustrates the case where the sell-side modifies multiple entitlements:

- Some entitlements are enabled
- Other entitlements are disabled

All modifications may be sent in one or multiple messages. The Execution Venue rejects all of these modifications.

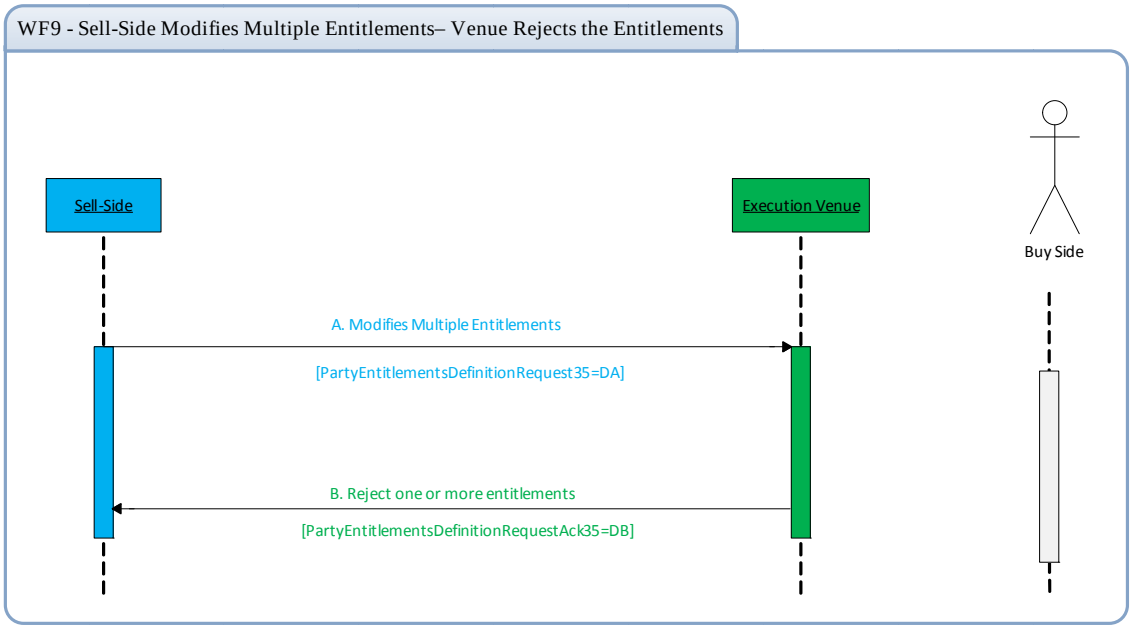


Figure 14: WF9 - Sell-Side Modifies Multiple Entitlements – Venue Rejects One or More Modifications

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Sell-side Modifies multiple entitlements	Sell-Side	→	**DA – PartyEntitlementsDefinitionRequest** EntitlementRequestID(1770)=❶ RequestingPartyGrp NoPartyEntitlements(1772)=N -> ListUpdateAction(1324)=Modify(M) -> PartyDetailGrp -> NoEntitlements(1773)=M -> -> EntitlementIndicator(1774)=Y|N (i.e.: enabled | disabled) -> -> EntitlementID(1776)=<Identifying this entitlement>	Execution Venue
(B) Execution Venue acknowledges and accepts		←	**DB – PartyEntitlementsDefinitionRequestAck** EntitlementRequestID(1770)=❶ EntitlementRequestStatus(1882)=Rejected(2) EntitlementRequestResult(1881)=<required>	

Table 9: WF9 - Sell-Side Modifies Multiple Entitlements – Venue Rejects One or More Modifications

Notes

- See scenario WF7

7.12 WF10 - Buy-Side Requests a New Entitlement for one or more Users - Approved

This scenario illustrates the case where the Execution Venue sends a request to the sell-side on behalf of one or more buy-side users for entitlement which is subsequently approved. A preliminary condition is that the sell-side subscribes to receive the requests or 'actively asks' (i.e. requests) to receive buy-side requests for entitlements.

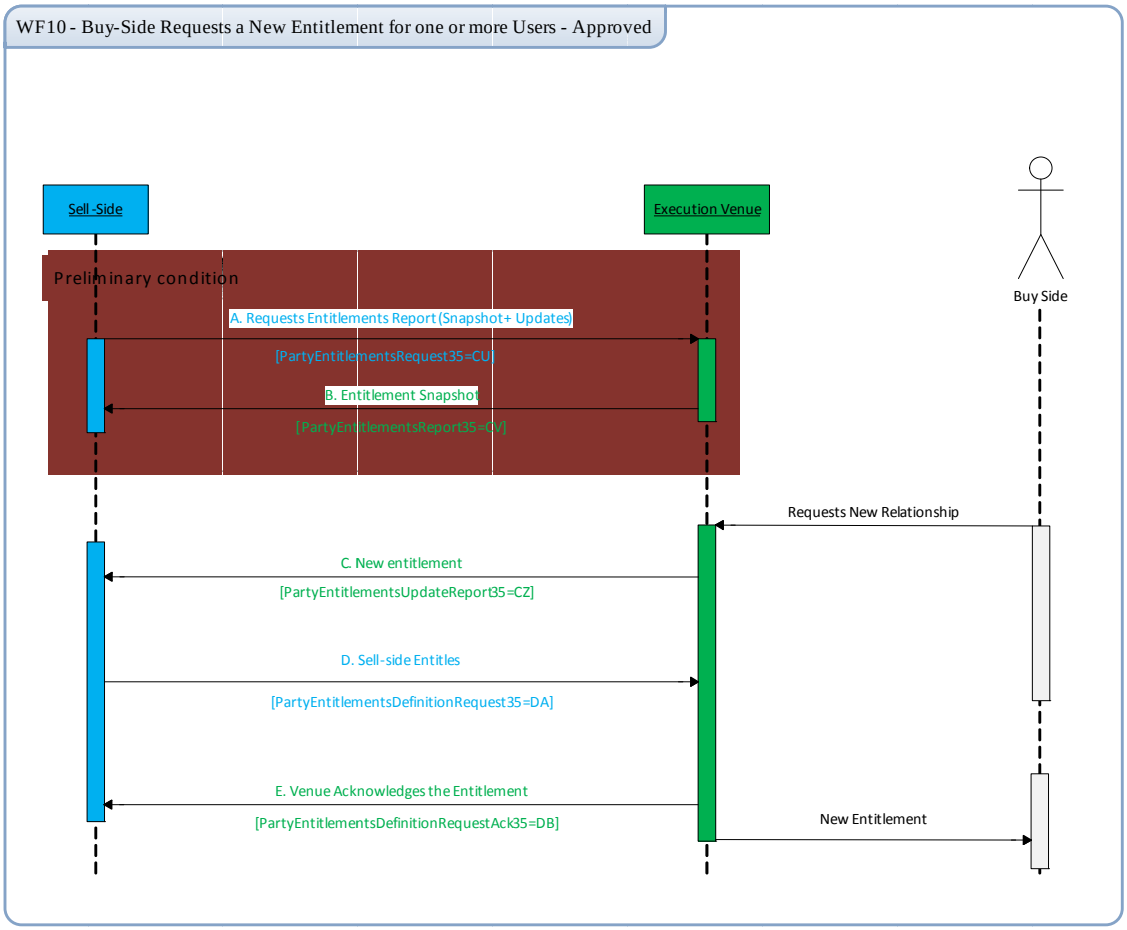


Figure 15: WF10 - Buy-Side Requests a New Entitlement for one or more Users - Approved

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Entitlement report request	Sell-Side	→	**CU – PartyEntitlementsRequest** EntitlementRequestID(1770)=❶ SubscriptionRequestType(263)=Snapshot + Updates(1) PartyDetailStatus(1672) <optional> MarketSegmentScopeGrp <optional> NoEntitlementType(2345) <optional> -> EntitlementType(1775)	Execution Venue
(B) Entitlement Report (Snapshot)		←	**CV – PartyEntitlementsReport** EntitlementRequestID(1770)=❶ EntitlementReportID(1771)= <required> See: WF6	
(C) New Entitlement(s) (Update report)		←	**CZ – PartyEntitlementsUpdateReport** EntitlementRequestID(1770)=❶ EntitlementReportID(1771)= <required> **EntitlementStatus(1883)=Pending(3)** NoPartyEntitments(1772)=1 -> ListUpdateAction(1324)=Add(A) -> PartyDetailsGrp -> NoEntitlements(1773)=M -> -> EntitlementID(1776)=❹	
(D) Sell-side Entitles		→	**DA – PartyEntitlementsDefinitionRequest** EntitlementRequestID(1770)=❺ RequestingPartyGrp NoPartyEntitlements(1772)=1 -> ListUpdateAction(1324)=Modify(M) -> PartyDetailGrp -> NoEntitlements(1773)=M -> -> EntitlementIndicator(1774)= **Y** (i.e. enabled) -> -> EntitlementID(1776)= ❹	
(E) Execution Venue acknowledges and accepts		←	**DB – PartyEntitlementsDefinitionRequestAck** EntitlementRequestID(1770)=❺ PartyDetailStatus(1672)=Active(0) EntitlementRequestStatus(1882)=Accepted(0) EntitlementRequestResult(1881)=successful(0) (default)	

Table 10: WF10 - Buy-Side Requests a New Entitlement for one or more Users - Approved

Notes

- See WF5 notes
- The sell-side owns the entitlement information to be set for a given counterparty, whether at the firm level or individual level
- The EntitlementStatus(1883)=Pending(3) is set by the ExecutionVenue; The PartyEntitlementsDefinitionRequest(35=DA) message would be requests for the execution venue to take an action that would result in the EntitlementStatus(1883) being set appropriately (Accepted(0) or Rejected(1)).
- During the period where the sell side sent the EntitlementDefinitionRequest (35=DA) and until the execution venue process and set the entitlement accordingly, the entitlement status may be reported as: EntitlementStatus(1883)=Requested(4).

7.13 WF11 - Buy-Side Requests Entitlement for one or more Users - Declined

This scenario illustrates the case where the Execution Venue sends a request to the sell-side on behalf of one or more buy-side users for entitlement which is subsequently declined. A preliminary condition is that the sell-side subscribes to receive the requests or 'actively asks' (i.e. requests) to receive buy-side requests for entitlements.

Figure 16: WF11 - Buy-Side Requests a New Entitlement for one or more Users – Declined

Model Flow

The following table illustrates the flows expected when communicating with an Execution Venue implementing FIX 5.0 SP2.

Model FIX 5.0				
(A) Entitlement report request	Sell-Side	→	**CU – PartyEntitlementsRequest** EntitlementRequestID(1770)=❶ SubscriptionRequestType(263)=Snapshot + Updates(1) PartyDetailStatus(1672) <optional> MarketSegmentScopeGrp <optional> NoEntitlementType(2345) <optional> -> EntitlementType(1775)	Execution Venue
(B) Entitlement Report (Snapshot)		←	**CV – PartyEntitlementsReport** EntitlementRequestID(1770)=❶ EntitlementReportID(1771)= <required> See: WF6	
(C) New Entitlement(s) (Update report)		←	**CZ – PartyEntitlementsUpdateReport** EntitlementRequestID(1770)=❶ EntitlementReportID(1771)= <required> **EntitlementStatus(1883)=Pending(3)** NoPartyEntitments(1772)=1 -> ListUpdateAction(1324)=Add(A) -> PartyDetailsGrp -> NoEntitlements(1773)=M -> -> EntitlementID(1776)=❹	
(D) Sell-side Declines the entitlement		→	**DA – PartyEntitlementsDefinitionRequest** EntitlementRequestID(1770)=❺ RequestingPartyGrp NoPartyEntitlements(1772)=1 -> ListUpdateAction(1324)=Modify(M) -> PartyDetailGrp -> NoEntitlements(1773)=M -> -> EntitlementIndicator(1774)= **N** (not entitled) -> -> EntitlementID(1776)= ❹	
(E) Execution Venue acknowledges		←	**DB – PartyEntitlementsDefinitionRequestAck** EntitlementRequestID(1770)=❺ PartyDetailStatus(1672)=Suspend(1) EntitlementRequestStatus(1882)=Accepted(0) EntitlementRequestResult(1881)=successful(0) (default)	

Table 11: WF11 - Buy-Side Requests a New Entitlement for one or more Users – Declined

Notes

- See WF10 notes

8 Message Detail

This section describes in detail all FIX application messages used in this volume. A summary of all the messages described in this volume is provided below.

8.1 User Request (35=BE)

UserRequest (BE)			Sell-Side -> Execution Venue	
This message is used to initiate a user action, logon, logout or password change. It can also be used to request a report on a user's status.				
Tag	**Field Name**	**Req'd**	**Description**	**Comment**
	StandardHeader	Y	MsgType = "BE"	
923	**UserRequestID**	Y	Unique identifier for a User Request.	
924	**UserRequestType**	Y	Indicates the action required by a User Request Message	
553	**Username**	Y	Userid or username.	
554	**Password**	N	Password or passphrase.	
925	**NewPassword**	N	New Password or passphrase	
1400	**EncryptedPasswordMethod**	N	Enumeration defining the encryption method used to encrypt password fields. At this time there are no encryption methods defined by FPL.	
1401	**EncryptedPasswordLen**	N	Length of the EncryptedPassword(1402) field	
1402	**EncryptedPassword**	N	Encrypted password - encrypted via the method specified in the field EncryptedPasswordMethod(1400)	
1403	**EncryptedNewPasswordLen**	N	Length of the EncryptedNewPassword(1404) field	
1404	**EncryptedNewPassword**	N	Encrypted new password - encrypted via the method specified in the field EncryptedPasswordMethod(1400)	
95	**RawDataLength**	N	Number of bytes in raw data field.	
96	**RawData**	N	Can be used to hand structures etc to other API's etc	
	StandardTrailer	Y	The standard FIX message trailer	

8.2 User Response (35=BF)

UserResponse (BF)			Execution Venue -> Sell-Side	
This message is used to respond to a user request message, it reports the status of the user after the completion of any action requested in the user request message.				
Tag	**Field Name**	**Req'd**	**Description**	**Comment**
	StandardHeader	Y	MsgType = "BF"	
923	**UserRequestID**	Y	Unique identifier for a User Request.	
553	**Username**	Y	Userid or username.	
926	**UserStatus**	N	Indicates the status of a user	
	ThrottleParamsGrp	N		
927	**UserStatusText**	N	Reason a request was not carried out	
	StandardTrailer	Y	The standard FIX message trailer	

8.3 Market Definition Request (35=BT)

MarketDefinitionRequest (BT)			Sell-Side -> Execution Venue	
The Market Definition Request message is used to request for market structure information from the Respondent that receives this request.				
Tag	**Field Name**	**Req'd**	**Description**	**Comment**
	StandardHeader	Y	MsgType = BT	
1393	**MarketReqID**	Y	Must be unique, or the ID of previous Market Segment Request to disable if SubscriptionRequestType = Disable previous Snapshot + Updates Request(2).	
263	**SubscriptionRequestType**	Y	Subscription Request Type	
1301	**MarketID**	N	Conditionally required if MarketSegmentID(1300) is specified on the request	
1300	**MarketSegmentID**	N	Identifies the market segment	
1325	**ParentMktSegmID**	N	Specifies that the Market Segment is a sub segment of the Market Segment defined in this field.	
	StandardTrailer	Y	The standard FIX message trailer	

8.4 Market Definition (35=BU)

MarketDefinition (BU)			Execution Venue -> Sell-Side	
The Market Definition message is used to respond to Market Definition Request. In a subscription, it will be used to provide the initial snapshot of the information requested. Subsequent updates are provided by the Market Definition Update Report.				
Tag	**Field Name**	**Req'd**	**Description**	**Comment**
	StandardHeader	Y	MsgType = BU	
	ApplicationSequenceControl	N	The ApplicationSequenceControl is used for application sequencing and recovery. Consisting of ApplSeqNum (1181), ApplID (1180), ApplLastSeqNum (1350), and ApplResendFlag (1352), FIX application messages that carries this component block will be able to use application level sequencing. ApplID, ApplSeqNum and ApplLastSeqNum fields identify the application id, application sequence number and the previous application sequence number (in case of intentional gaps) on each application message that carries this block.	
1394	**MarketReportID**	Y	Unique identifier for each Market Definition message	
1393	**MarketReqID**	N	Unique ID of a Market Definition Request message.	
1301	**MarketID**	Y	Identifies the Market	
1300	**MarketSegmentID**	N	Identifies the market segment	
1396	**MarketSegmentDesc**	N	Description or name of Market Segment	
1397	**EncodedMktSegmDescLen**	N	Must be set if EncodedMktSegmDesc field is	

			specified and must immediately precede it.	
1398	**EncodedMktSegmDesc**	N	Encoded (non-ASCII characters) representation of the MarketSegmDesc field in the encoded format specified via the MessageEncoding field.	
1325	**ParentMktSegmID**	N	Specifies that the Market Segment is a sub segment of the Market Segment defined in this field.	
15	**Currency**	N	The default trading currency	
	BaseTradingRules	N	Insert here the set of "BaseTradingRules" fields defined in "common components of application messages"	
	OrdTypeRules	N	Insert here the set of "OrdTypeRules" fields defined in "common components of application messages"	
	TimeInForceRules	N	Insert here the set of "TimeInForceRules" fields defined in "common components of application messages"	
	ExecInstRules	N	Insert here the set of "ExecInstRules" fields defined in "common components of application messages"	
60	**TransactTime**	N	Timestamp when the business transaction represented by the message occurred.	
58	**Text**	N	Comment, instructions, or other identifying information.	
354	**EncodedTextLen**	N	Must be set if EncodedText field is specified and must immediately precede it.	
355	**EncodedText**	N	Encoded (non-ASCII characters) representation of the Text field in the encoded format specified via the MessageEncoding field.	
	StandardTrailer	Y	The standard FIX message trailer	

8.5 Party Details Group component Diagram

The Party Details Group is contained (nested) in messages:

- PartyDetailsListReport (35=CG)
- PartyDetailsListUpdateReport (35=CK)
- PartyEntitlementsReport (35=CV)
- PartyEntitlementsUpdateReport (35=CZ)
- PartyEntitlementsDefinitionRequest (35=DA)
- PartyEntitlementsDefinitionRequestAck (35=DB)

This component may be constructed in two different ways.
The following diagram illustrates the two possibilities of structuring Party Details Group component:

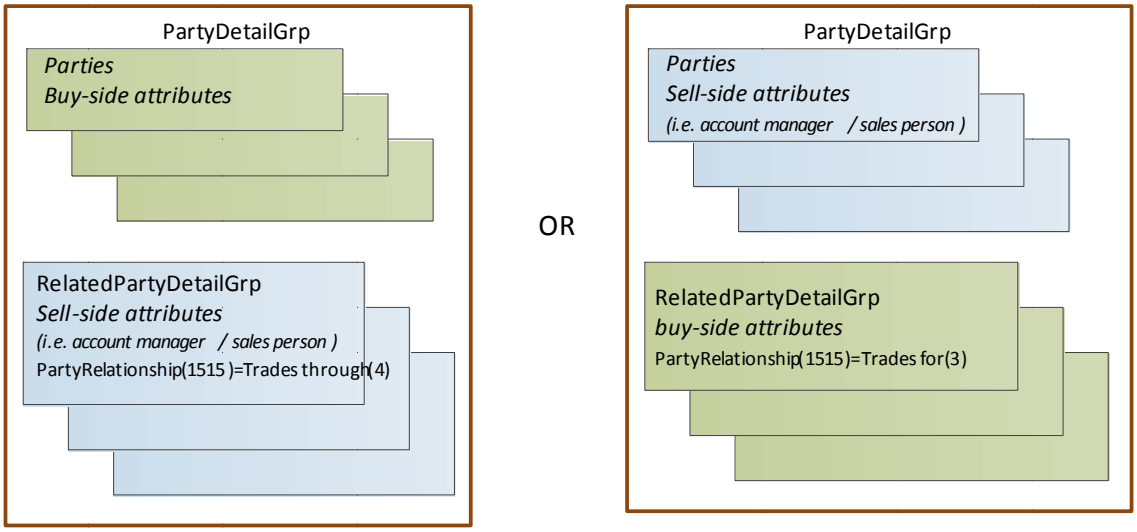


Figure 17: Party Details Group component Structure

8.6 Party Detail List Request (35=CF)

8.6.1 Party Details List Request Message Structure Diagram

The following diagram illustrates the structure of the Party Details List Request message:

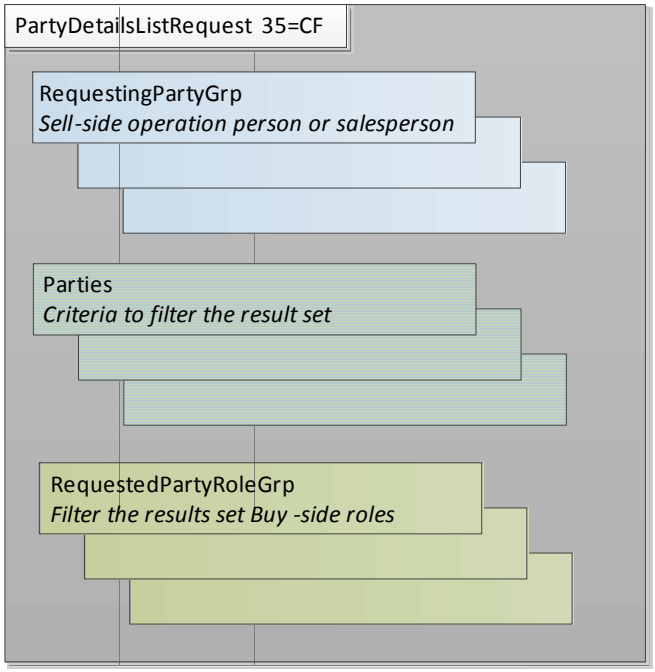


Figure 18: Party Details List Request 35=CF Message Structure

8.6.2 Party Detail List Request Message Definition

PartyDetailsListRequest (CF)			Sell-Side -> Execution Venue	
The PartyDetailsListRequest is used to request party detail information.				
Tag	**Field Name**	**Req'd**	**Description**	**Comment**
	StandardHeader	Y	MsgType = CF	
1505	**PartyDetailsListRequestID**	Y	Unique identifier for PartyDetailsListRequest.	
	RequestingPartyGrp	N	May be used to identify the party making the request and their role.	
1657	**NoRequestingPartyIDs**	N	Number of requesting party identifiers.	
->1658	**RequestingPartyID**	N	Required when NoRequestingPartyIDs > 0.	
->1659	**RequestingPartyIDSource**	N	Required when NoRequestingPartyIDs > 0.	
->1660	**RequestingPartyRole**	N	Required when NoRequestingPartyIDs > 0.	
->	**RequestingPartySubGrp**	N	Sub identifiers for the requesting party.	
	Parties	N	Scope of the query/request for specific party(-ies).	
453	**NoPartyIDs**	N	Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole	
->448	**PartyID**	N	Used to identify source of PartyID. Required if PartyIDSource is specified. Required if NoPartyIDs > 0.	
->447	**PartyIDSource**	N	Used to identify class source of PartyID value (e.g. BIC). Required if PartyID is specified. Required if NoPartyIDs > 0.	
->452	**PartyRole**	N	Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs > 0.	
->	**PtysSubGrp**	N	Repeating group of Party sub-identifiers.	
	RequestedPartyRoleGrp	N	Scope of the query/request for specific party role(s)	
1508	**NoRequestedPartyRoles**	N	Number of requested party roles.	
->1509	**RequestedPartyRole**	N	Identifies the type of party role requested. Required if NoRequestedPartyRoles > 0.	
	PartyRelationshipGrp	N	Scope of the query/reqeust for specific party relationship(s)	
1514	**NoPartyRelationships**	N	Number of party relationships.	
->1515	**PartyRelationship**	N	Identifies the type of party relationship requested. Required if NoPartyRelationships > 0.	
263	**SubscriptionRequestType**	N	Subscription Request Type	
58	**Text**	N	Free format text string (Note: this field does not have a specified maximum length)	
	StandardTrailer	Y	The standard FIX message trailer	

8.7 Party Detail List Report (35=CG)

8.7.1 Party Details List Report Message Structure Diagram

The following diagram illustrates the structure of Party Details List Report message:

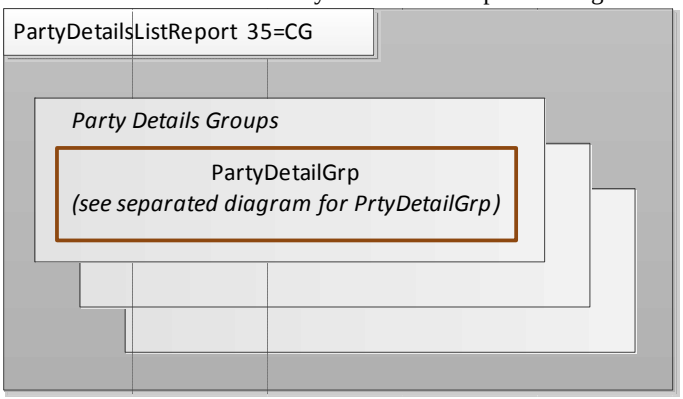


Figure 19: Party Details List Report 35=CG Message Structure

8.7.2 Party Detail List Report Message Definition

PartyDetailsListReport (CG)			Execution Venue -> Sell-Side	
The PartyDetailsListReport message is used to disseminate party details between counterparties. PartyDetailsListReport messages may be sent in response to a PartyDetailsListRequest message or sent unsolicited.				
Tag	**Field Name**	**Req'd**	**Description**	**Comment**
	StandardHeader	Y	MsgType = CG	
	ApplicationSequenceControl	N	The ApplicationSequenceControl is used for application sequencing and recovery. Consisting of ApplSeqNum (1181), ApplID (1180), ApplLastSeqNum (1350), and ApplResendFlag (1352), FIX application messages that carries this component block will be able to use application level sequencing. ApplID, ApplSeqNum and ApplLastSeqNum fields identify the application id, application sequence number and the previous application sequence number (in case of intentional gaps) on each application message that carries this block.	
1510	**PartyDetailsListReportID**	Y	Identifier for the PartyDetailsListReport and the PartyDetailsListUpdateReport.	
1505	**PartyDetailsListRequestID**	N	Conditionally required when responding to the PartyDetailsListRequest message.	
1511	**RequestResult**	N	Conditionally required when responding to the PartyDetailsListRequest message.	
1512	**TotNoParties**	N	Total number of PartyListGrp returned.	
893	**LastFragment**	N	Indicates whether this message is the last in a sequence of messages for those messages that support fragmentation, such as Allocation Instruction, Mass Quote, Security List, Derivative Security List	
	PartyDetailGrp	N	Contains details for a party, including related parties and alternative party identifiers.	
1671	**NoPartyDetails**	N	Number of party details.	
->1691	**PartyDetailID**	N	The identification of the party. Required when NoPartyDetails(1671) > 0.	
->1692	**PartyDetailIDSource**	N	Used to identify source of PartyID value (e.g. BIC). Required when NoPartyDetails(1671) > 0.	
->1693	**PartyDetailRole**	N	Identifies the type of PartyID (e.g. Executing Broker). Required when NoPartyDetails(1671) > 0.	
->1674	**PartyDetailRoleQualifier**	N	Qualifies the value of PartyRole(452)	
->	**PartyDetailSubGrp**	N	Additional party sub-identifiers	

->	**PartyDetailAltIDGrp**	N	Optionally used to specify alternate IDs to identify the party specified.	
->	**RelatedPartyDetailGrp**	N	May not be specified in PartyDetailsListUpdateReport(35=CK) if ListUpdateAction(1324) = D(Delete)	
->->1562	**NoRelatedPartyDetailID**	N	Number of related party detail identifiers.	
->->->1563	**RelatedPartyDetailID**	N	Required if NoRelatedPartyDetails > 0.	
->->->1564	**RelatedPartyDetailIDSource**	N	Required if NoRelatedPartyDetails > 0.	
->->->1565	**RelatedPartyDetailRole**	N	Required if NoRelatedPartyDetails > 0.	
->->->1675	**RelatedPartyDetailRoleQualifier**	N	Qualifies the value of RelatedPartyRole(1565)	
->->->	**RelatedPartyDetailSubGrp**	N	PartySubGrp for related parties.	
->->->	**RelatedPartyDetailAltIDGrp**	N	Alternative identifiers for parties related to the party specified in the PartyDetailGrp.	
->->->	**PartyRelationshipGrp**	N	Repeating group of party relationships.	
->->->->1514	**NoPartyRelationships**	N	Number of party relationships.	
->->->->->1515	**PartyRelationship**	N	Identifies the type of party relationship requested. Required if NoPartyRelationships > 0.	
->1672	**PartyDetailStatus**	N	Specifies the status of the party information, whether active or suspended (inactive).	
60	**TransactTime**	N	Timestamp when the business transaction represented by the message occurred.	
58	**Text**	N	Free format text string (Note: this field does not have a specified maximum length)	
354	**EncodedTextLen**	N	Byte length of encoded (non-ASCII characters) EncodedText (355) field.	
355	**EncodedText**	N	Encoded (non-ASCII characters) representation of the Text (58) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the Text field.	
1328	**RejectText**	N	Identifies the reason for rejection.	
	StandardTrailer	Y	The standard FIX message trailer	

8.8 Party Detail List Update Report (35=CG)

8.8.1 Party Details List Update Report Message Structure Diagram

The following diagram illustrates the structure of Party Details List Update Report message:

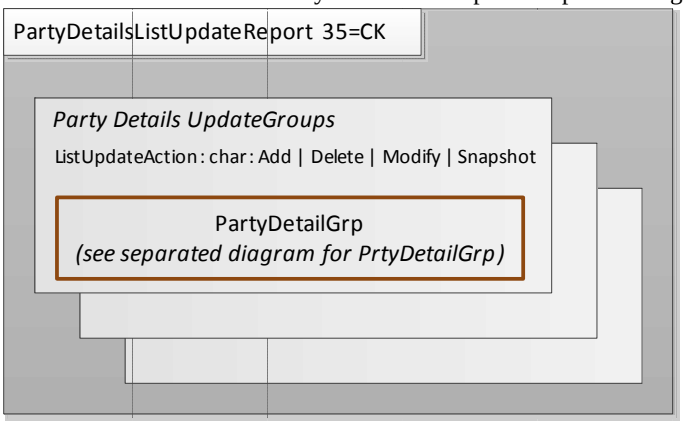


Figure 20: Party Details List Update Report 35=CK Message Structure

8.8.2 Party Detail List Update Report Message Definition

PartyDetailsListUpdateReport (CG)			Execution Venue -> Sell-Side	
The PartyDetailsListReport message is used to disseminate party details between counterparties. PartyDetailsListReport messages may be sent in response to a PartyDetailsListRequest message or sent unsolicited.				
Tag	**Field Name**	**Req'd**	**Description**	**Comment**
	StandardHeader	Y	MsgType = CG	
	ApplicationSequenceControl	N	The ApplicationSequenceControl is used for application sequencing and recovery. Consisting of ApplSeqNum (1181), ApplID (1180), ApplLastSeqNum (1350), and ApplResendFlag (1352), FIX application messages that carries this component block will be able to use application level sequencing. ApplID, ApplSeqNum and ApplLastSeqNum fields identify the application id, application sequence number and the previous application sequence number (in case of intentional gaps) on each application message that carries this block.	
1510	**PartyDetailsListReportID**	Y	Identifier for the PartyDetailsListReport and the PartyDetailsListUpdateReport.	
1505	**PartyDetailsListRequestID**	N	Conditionally required when responding to the PartyDetailsListRequest message.	
1511	**RequestResult**	N	Conditionally required when responding to the PartyDetailsListRequest message.	
1512	**TotNoParties**	N	Total number of PartyListGrp returned.	
893	**LastFragment**	N	Indicates whether this message is the last in a sequence of messages for those messages that support fragmentation, such as Allocation Instruction, Mass Quote, Security List, Derivative Security List	
	PartyDetailGrp	N	Contains details for a party, including related parties and alternative party identifiers.	
1671	**NoPartyDetails**	N	Number of party details.	
->1691	**PartyDetailID**	N	The identification of the party. Required when NoPartyDetails(1671) > 0.	
->1692	**PartyDetailIDSource**	N	Used to identify source of PartyID value (e.g. BIC). Required when NoPartyDetails(1671) > 0.	
->1693	**PartyDetailRole**	N	Identifies the type of PartyID (e.g. Executing Broker). Required when NoPartyDetails(1671) > 0.	
->1674	**PartyDetailRoleQualifier**	N	Qualifies the value of PartyRole(452)	
->	**PartyDetailSubGrp**	N	Additional party sub-identifiers	

->	**PartyDetailAltIDGrp**	N	Optionally used to specify alternate IDs to identify the party specified.	
->	**RelatedPartyDetailGrp**	N	May not be specified in PartyDetailsListUpdateReport(35=CK) if ListUpdateAction(1324) = D(Delete)	
->->1562	**NoRelatedPartyDetailID**	N	Number of related party detail identifiers.	
->->->1563	**RelatedPartyDetailID**	N	Required if NoRelatedPartyDetails > 0.	
->->->1564	**RelatedPartyDetailIDSource**	N	Required if NoRelatedPartyDetails > 0.	
->->->1565	**RelatedPartyDetailRole**	N	Required if NoRelatedPartyDetails > 0.	
->->->1675	**RelatedPartyDetailRoleQualifier**	N	Qualifies the value of RelatedPartyRole(1565)	
->->->	**RelatedPartyDetailSubGrp**	N	PartySubGrp for related parties.	
->->->	**RelatedPartyDetailAltIDGrp**	N	Alternative identifiers for parties related to the party specified in the PartyDetailGrp.	
->->->	**PartyRelationshipGrp**	N	Repeating group of party relationships.	
->->->->1514	**NoPartyRelationships**	N	Number of party relationships.	
->->->->->1515	**PartyRelationship**	N	Identifies the type of party relationship requested. Required if NoPartyRelationships > 0.	
->1672	**PartyDetailStatus**	N	Specifies the status of the party information, whether active or suspended (inactive).	
60	**TransactTime**	N	Timestamp when the business transaction represented by the message occurred.	
58	**Text**	N	Free format text string (Note: this field does not have a specified maximum length)	
354	**EncodedTextLen**	N	Byte length of encoded (non-ASCII characters) EncodedText (355) field.	
355	**EncodedText**	N	Encoded (non-ASCII characters) representation of the Text (58) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the Text field.	
1328	**RejectText**	N	Identifies the reason for rejection.	
	StandardTrailer	Y	The standard FIX message trailer	

8.9 Party Entitlements Request (35=CU)

8.9.1 Party Entitlements Request Message Structure Diagram

The following diagram illustrates the structure of the Party Entitlements Request message:

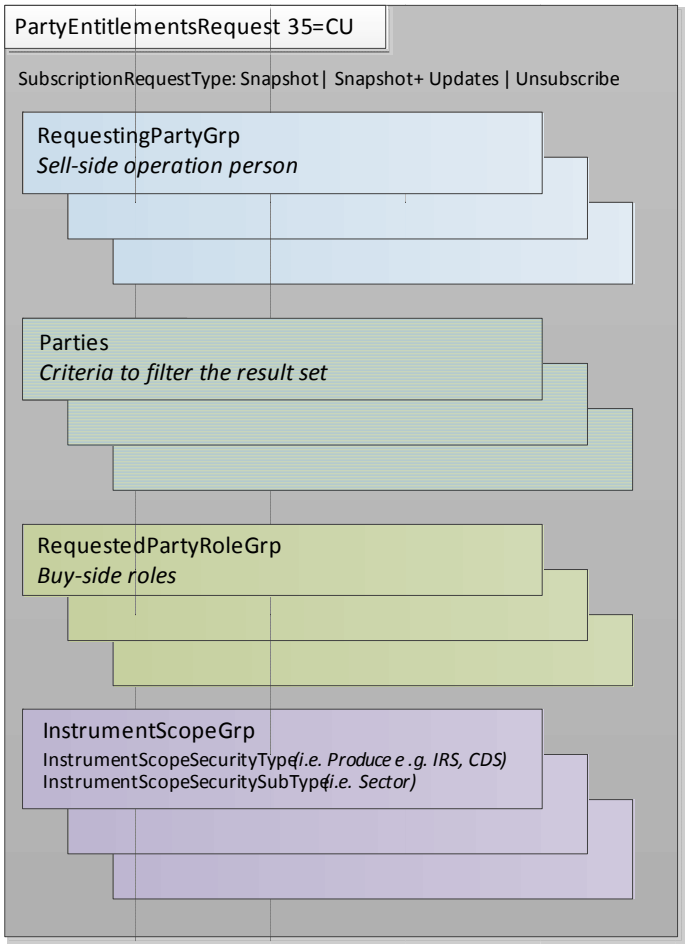


Figure 21: Party Entitlements Request 35=CU Message Structure

8.9.2 Party Entitlement Request Example-1

In this example the sell-side operation person, (i.e. Kate Middleton) requests all the entitlements that are associated with specific sales person (i.e. James Bradley).

8.9.2.1 Party Entitlement Request FIXML

```
<?xml version="1.0" encoding="UTF-8"?>
<FIXML xmlns="http://www.fixprotocol.org/FIXML-5-0-SP2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.fixprotocol.org/FIXML-5-0-SP2 ./Fixml/fixml-main-5-0-SP2.xsd">
   <PtyEntlmntReq  ReqID="1">
      <ReqPty ID="Bank-1" Src="D" R="1">
         <Sub ID="Kate Middleton" Typ="2"/>
      </ReqPty>
      <Pty ID="Bank-1" Src="D" R="1"> <!-- Results set contains only entitlements associated
with sales person: James Bradley -->
         <Sub ID="James Bradley" Typ="9" /> <!-- Sales person -->
         <Sub ID="James-Bradley-ID" Typ="2" /> <!-- Sales person ID -->
         </Pty>
   </PtyEntlmntReq>
</FIXML>
```

8.9.3 Party Entitlement Request Example-1

In this example the sell-side operation person, (i.e. Kate Middleton) requests all the entitlements for a specific buy side firm (i.e. Hedge-Fund).

8.9.3.1 Party Entitlement Request FIXML

```
<?xml version="1.0" encoding="UTF-8"?>
<FIXML xmlns="http://www.fixprotocol.org/FIXML-5-0-SP2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.fixprotocol.org/FIXML-5-0-SP2 ./Fixml/fixml-main-5-0-SP2.xsd">
   <PtyEntlmntReq  ReqID="1">
      <ReqPty ID="Bank-1" Src="D" R="1">
         <Sub ID="Kate Middleton" Typ="2"/>
      </ReqPty>
      <Pty  ID="User-1" Src="D" R="3"> <!-- Results set contains only entitlements of firm
'Hedge Fund' -->
               <Sub ID="Hedge Fund" Typ="1"/> <!-- firm -->
         </Pty>
   </PtyEntlmntReq>
</FIXML>
```

8.9.4 Party Entitlements Request Message Definition

PartyEntitlementsRequest (CU)			Sell-Side -> Execution Venue	
The PartyEntitlementsRequest message is used to request for entitlement information for one or more party(-ies), specific party role(s), or specific instruments(s).				
Tag	**Field Name**	**Req'd**	**Description**	**Comment**
	StandardHeader	Y	MsgType=CU	
1770	**EntitlementRequestID**	N	Unique identifier for PartyEntitlementsRequest(35=CU).	
263	**SubscriptionRequestType**	N	Subscription Request Type	
	RequestingPartyGrp	N	May be used to identify the party making the request and their role.	
1657	**NoRequestingPartyIDs**	N	Number of requesting party identifiers.	
->1658	**RequestingPartyID**	N	Required when NoRequestingPartyIDs > 0.	
->1659	**RequestingPartyIDSource**	N	Required when NoRequestingPartyIDs > 0.	
->1660	**RequestingPartyRole**	N	Required when NoRequestingPartyIDs > 0.	
->	**RequestingPartySubGrp**	N	Sub identifiers for the requesting party.	
	Parties	N	Scope of the query/request for specific party(-ies).	
453	**NoPartyIDs**	N	Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole	
->448	**PartyID**	N	Used to identify source of PartyID. Required if PartyIDSource is specified. Required if NoPartyIDs > 0.	
->447	**PartyIDSource**	N	Used to identify class source of PartyID value (e.g. BIC). Required if PartyID is specified. Required if NoPartyIDs > 0.	
->452	**PartyRole**	N	Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs > 0.	
->	**PtysSubGrp**	N	Repeating group of Party sub-identifiers.	
	RequestedPartyRoleGrp	N	Scope of the query/request for specific party roles. For example, "all information for PartyRole=24".	
1508	**NoRequestedPartyRoles**	N	Number of requested party roles.	
->1509	**RequestedPartyRole**	N	Identifies the type of party role requested. Required if NoRequestedPartyRoles > 0.	
1784	**EntitlementPlatform**	N	The area to which the entitlement is applicable. This can be a trading platform or an offering.	
	InstrumentScopeGrp	N	Scope of the query/request for	

			specific securities.	
1883	**EntitlementStatus**	N		
	EntitlementTypeGrp	N		
2345	NoEntitlementTypes	N	Nomber of Entitlement Types	
-> 1775	EntitlementType	N		
	MarketSegmentScopeGrp			
1310	NoMarket			
->1301	Market ID			
->1300	MarketSegmentID			
1656	**NoInstrumentScopes**	N	Number of instrument scopes.	
->1535	**InstrumentScopeOperator**	N	Required when NoInstrumentScopes > 0.	
->	**InstrumentScope**	N	Used to specify the instrument	
->->1536	**InstrumentScopeSymbol**	N	Used to limit instrument scope to specified symbol. See Symbol(55) field for description.	
->->1537	**InstrumentScopeSymbolSfx**	N	Used to limit instrument scope to specified symbol suffix. See SymbolSfx(65) field for description.	
->->1538	**InstrumentScopeSecurityID**	N	Used to limit instrument scope to specified security identifier. See SecurityID(48) field for description.	
->->1539	**InstrumentScopeSecurityIDSource**	N	Used to limit instrument scope to specified security identifier source. See SecurityIDSource(22) field for description.	
->->	**InstrumentScopeSecurityAltIDGrp**	N	Alternative SecurityIDs for an instrument specified in the InstrumentScope.	
->->1543	**InstrumentScopeProduct**	N	Used to limit instrument scope to specified instrument product category. See Product (460) field for description.	
->->1544	**InstrumentScopeProductComplex**	N	Used to limit instrument scope to specified product complex. See ProductComplex(1227) field for description.	
->->1545	**InstrumentScopeSecurityGroup**	N	Used to limit instrument scope to specified security group. See SecurityGroup(1151) field for description.	
->->1546	**InstrumentScopeCFICode**	N	Used to limit instrument scope to specified CFICode. See CFICode(461) field for description.	
->->1547	**InstrumentScopeSecurityType**	N	Used to limit instrument scope to specified security type. See SecurityType(167) field for description).	
->->154	**InstrumentScopeSecuritySubType**	N	Used to limit instrument scope to specified security sub-type. See	

8			SecuritySubType(762) field for description.	
->->1549	**InstrumentScopeMaturityMonthYear**	N	Used to limit instrument scope to specified maturity month and year. See MaturityMonthYear(200) field for description.	
->->1550	**InstrumentScopeMaturityTime**	N	Used to limit instrument scope to specified maturity time. See MaturityTime(1079) field for description.	
->->1551	**InstrumentScopeRestructuringType**	N	Used to limit instrument scope to specified restructuring type. See RestructuringType(1449) field for description.	
->->1552	**InstrumentScopeSeniority**	N	Used to limit instrument scope to specified seniority type. See Seniority(1450) field for description.	
->->1553	**InstrumentScopePutOrCall**	N	Used to limit instrument scope to puts or calls. See PutOrCall(201) field for description.	
->->1554	**InstrumentScopeFlexibleIndicator**	N	Used to limit instrument scope to securities that can be defined using flexible terms or not. See FlexibleIndicator(1244) field for description.	
->->1555	**InstrumentScopeCouponRate**	Y	Used to limit instrument scope to specified coupon rate. See CouponRate(223) field for description.	
->->1616	**InstrumentScopeSecurityExchange**	N	Used to limit instrument scope to specified security exchange. See SecurityExchange(207) field for description.	
->->1556	**InstrumentScopeSecurityDesc**	N	Used to limit instrument scope to specified security description. See SecurityDesc(107) field for description.	
->->1620	**InstrumentScopeEncodedSecurityDescLen**	N	Byte length of encoded (non-ASCII characters) InstrumentScopeEncodedSecurityDesc (1621) field	
->->1621	**InstrumentScopeEncodedSecurityDesc**	N	Encoded (non-ASCII characters) representation of the InstrumentScopeSecurityDesc (1556) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the InstrumentScopeSecurityDesc field.	
->->1557	**InstrumentScopeSettlType**	N	Can be used to specify FX tenors.	
58	**Text**	N	Free format text string (Note: this field does not have a specified	

			maximum length)	
	StandardTrailer	Y	The standard FIX message trailer	

8.10 Party Entitlements Report (35=CV)

8.10.1 Party Entitlements Report Message Structure Diagram

The following diagram illustrates the structure of the Party Entitlements Report message:

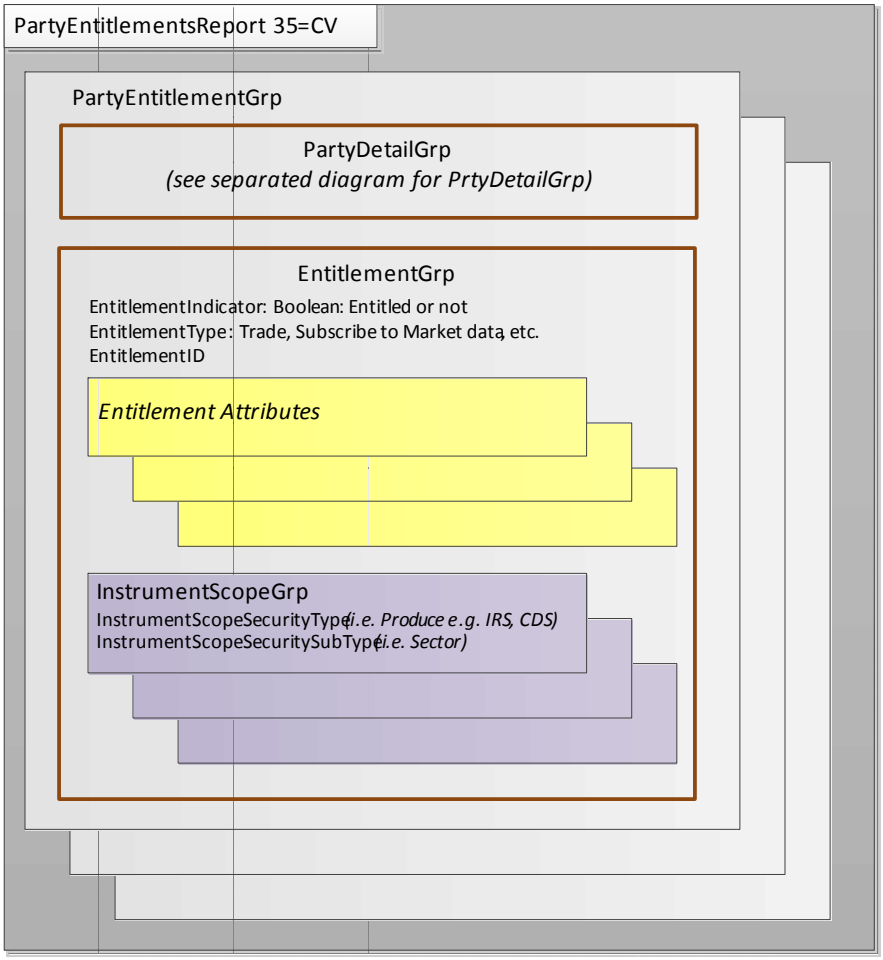


Figure 22: Party Entitlements Report 35=CV Message Structure

8.10.2 Party Entitlement Report Example

In this example the Execution Venue sends an entitlement report containing data for two users. This data may be summarised as:

1. An buy-side user (User-1) is entitled to trade IRS/SONIA with entity Entity-1
2. An buy-side user (User-2) is entitled to trade IRS/SONIA with entity Entity-1

The below table contains the business level attributes of the entitlement report:

Attribute	Entitled user	Enabled user
Sell-side Identifier	Bank-1	Entity-1
Client Account	Account-1	Account-2
Primary Sales Person User ID	James-Bradley-ID	James-Bradley-ID
Primary Sales Person Name	James Bradley	James Bradley
Product	IRS	IRS
Sector Name1	SONIA	SONIA
Buy-side User ID	User-1	User-2
Buy side user name	Jon A. Smith	Oliver Taylor
Firm	Hedge fund	Hedge fund
LEI	LZ123	LZ123
Enablement Type	Trade	Trade
Status	Entitled	Entitled
Last Modified	15-Feb-2013 11:47:12	15-Feb-2013 11:52:36
Modified By	Kate Middleton	William Morris
Tier	Level 1	Level 2
Allow RFQ trading	N	Y
Entitle to trades instruments that can be cleared by <Clearing House>		CME

Table 12: Example 1 - Execution Venue sends an entitlement report

8.10.2.1 Party Entitlement Report FIXML

The following is the FIXML message tags and values that are expected in this example:

```
<?xml version="1.0" encoding="UTF-8"?>
<FIXML xmlns="http://www.fixprotocol.org/FIXML-5-0-SP2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.fixprotocol.org/FIXML-5-0-SP2 ./Fixml/fixml-main-5-0-SP2.xsd">
   <PtyEntlmntRpt RptID="1" TotNoPtys="2">
      <PtyEntlmnt>
         <PtyDetl ID="User-1" Src="D" R="3"> <!-- buy-side user attributes -->
            <Sub ID="Jon A. Smith" Typ="9"/> <!-- Contact name-->
            <Sub ID="Hedge Fund" Typ="1"/> <!-- firm -->
            <Sub ID="jon.a.smith@HedgeFund.com" Typ="8" /> <!-- email -->
            <AltPty ID="LZ123" Src="N"/>                <!-- LEI  -->
            <ReltdPtyDetl ID="Bank-1" Src="D" R="1"> <!-- sell-side attributes -->
               <Sub ID="Account-1" Typ="19" /> <!-- Client account -->
               <Sub ID="James Bradley" Typ="9" /> <!-- Sales person -->
               <Sub ID="James-Bradley-ID" Typ="2" /> <!-- Sales person ID -->
               <Rltnshp Rltnshp="4" /> <!-- Trade through -->
            </ReltdPtyDetl>
         </PtyDetl>
         <Entlmnt Ind="Y" Typ="0" >
            <Attrib Typ="1500" Datatyp="5" Value="Level-1" /> <!-- FIX tag MDStreamID(1500)-->
            <Attrib Typ="4003" Datatyp="13" Value="N" /> <!-- Allow RFQ -->
            <Attrib Typ="4004" Datatyp="14" Value="CME" /> <!-- Entitled to trade instruments
that are cleared through CME-->
            <Attrib Typ="4004" Datatyp="14" Value="LCH" /> <!-- Entitled to trade instruments
that are cleared through LCH -->
            <InstrmtScope Oper="1" SecTyp="IRS" SecSubTyp="SONIA"/>
         </Entlmnt>
      </PtyEntlmnt>
      <PtyEntlmnt>
         <PtyDetl ID="User-2" Src="D" R="3"> <!-- buy-side user attributes -->
            <Sub ID="Oliver Taylor" Typ="9"/> <!-- Contact name-->
            <Sub ID="Hedge Fund" Typ="1"/> <!-- firm -->
            <Sub ID="oliver.taylor@HedgeFund.com" Typ="8" /> <!-- email -->
            <AltPty ID="LZ123" Src="N"/>                <!-- LEI  -->
            <ReltdPtyDetl ID="Bank-1" Src="D" R="1"> <!-- sell-side attributes -->
               <Sub ID="Account-2" Typ="19" /> <!-- Client account -->
               <Sub ID="James Bradley" Typ="9" /> <!-- Sales person -->
               <Sub ID="James-Bradley-ID" Typ="2" /> <!-- Sales person ID -->
               <Rltnshp Rltnshp="4" /> <!-- Trade through -->
            </ReltdPtyDetl>
         </PtyDetl>
         <Entlmnt Ind="Y" Typ="0" >
            <Attrib Typ="1500" Datatyp="5" Value="Level-2" /> <!-- FIX tag MDStreamID(1500)-->
            <Attrib Typ="4003" Datatyp="13" Value="Y" /> <!-- Allow RFQ -->
            <Attrib Typ="4004" Datatyp="14" Value="CME" /> <!--Entitled to trade instruments
that are cleared through CME -->
            <Attrib Typ="4004" Datatyp="14" Value="LCH" /> <!-- Entitled to trade instruments
that are cleared through LCH-->
            <InstrmtScope Oper="1" SecTyp="IRS" SecSubTyp="SONIA"/>
         </Entlmnt>
      </PtyEntlmnt>
   </PtyEntlmntRpt>
</FIXML>
```

8.10.3 Party Entitlements Report Message Definition

PartyEntitlementsReport (CV)			Execution Venue -> Sell-Side	
The PartyEntitlementsReport is used to report entitlements for one or more parties, party role(s), or specific instrument(s).				
Tag	**Field Name**	**Req'd**	**Description**	**Comment**
	StandardHeader	Y	MsgType=CV	
	ApplicationSequenceControl	N	The ApplicationSequenceControl is used for application sequencing and recovery. Consisting of ApplSeqNum (1181), ApplID (1180), ApplLastSeqNum (1350), and ApplResendFlag (1352), FIX application messages that carries this component block will be able to use application level sequencing. ApplID, ApplSeqNum and ApplLastSeqNum fields identify the application id, application sequence number and the previous application sequence number (in case of intentional gaps) on each application message that carries this block.	
1771	**EntitlementReportID**	Y	Identifier for the PartyEntitlementsReport(35=CV).	
1770	**EntitlementRequestID**	N	Conditionally required when responding to PartyEntitlementsRequest(35=CU).	
1511	**RequestResult**	N	Conditionally required when responding to Party Entitlements Request.	
1512	**TotNoParties**	N	Total number of PartyListGrp returned.	
893	**LastFragment**	N	Indicates whether this message is the last in a sequence of messages for those messages that support fragmentation, such as Allocation Instruction, Mass Quote, Security List, Derivative Security List	
	PartyEntitlementGrp	N	Conveys a list of parties (optionally including related parties) and the entitlements for each.	
1772	**NoPartyEntitlements**	N	Number of party entitlement values.	
->	**PartyDetailGrp**	N	Required if NoPartyEntitlements(1772) > 0.	
->->1671	**NoPartyDetails**	N	Number of party details.	
->->->1691	**PartyDetailID**	N	The identification of the party. Required when NoPartyDetails(1671) > 0.	

->->->1692	PartyDetailIDSource	N	Used to identify source of PartyID value (e.g. BIC). Required when NoPartyDetails(1671) > 0.	
->->->1693	PartyDetailRole	N	Identifies the type of PartyID (e.g. Executing Broker). Required when NoPartyDetails(1671) > 0.	
->->->1674	PartyDetailRoleQualifier	N	Qualifies the value of PartyRole(452)	
->->->	**PartyDetailSubGrp**	N	Additional party sub-identifiers	
->->->	**PartyDetailAltIDGrp**	N	Optionally used to specify alternate IDs to identify the party specified.	
->->->	**RelatedPartyDetailGrp**	N	May not be specified in PartyDetailsListUpdateReport(35=CK) if ListUpdateAction(1324) = D(Delete)	
->->->->1562	NoRelatedPartyDetailID	N	Number of related party detail identifiers.	
->->->->->1563	RelatedPartyDetailID	N	Required if NoRelatedPartyDetails > 0.	
->->->->->1564	RelatedPartyDetailIDSource	N	Required if NoRelatedPartyDetails > 0.	
->->->->->1565	RelatedPartyDetailRole	N	Required if NoRelatedPartyDetails > 0.	
->->->->->1675	RelatedPartyDetailRoleQualifier	N	Qualifies the value of RelatedPartyRole(1565)	
->->->->->	**RelatedPartyDetailSubGrp**	N	PartySubGrp for related parties.	
->->->->->	**RelatedPartyDetailAltIDGrp**	N	Alternative identifiers for parties related to the party specified in the PartyDetailGrp.	
->->->->->	**PartyRelationshipGrp**	N	Repeating group of party relationships.	
->->->->->1514	NoPartyRelationships	N	Number of party relationships.	
->->->->->->1511	PartyRelationship	N	Identifies the type of party relationship requested. Required if NoPartyRelationships > 0.	

5				
->->->1672	**PartyDetailStatus**	N	Specifies the status of the party information, whether active or suspended (inactive).	
->1883	**EntitlementStatus**	N		
->	**EntitlementGrp**	N	Required unless omitted to indicate the removal of entitlements for the party(-ies) specified in the PartyDetailGrp component.	
->->1773	**NoEntitlements**	N	Number of entitlement values.	
->->->1774	**EntitlementIndicator**	N	Required if NoEntitlements(1773) > 0.	
->->->1775	**EntitlementType**	N	Absence of this field indicates the meaning of the entitlement is implicit.	
->->->	**EntitlementAttribGrp**	N	conveys a list of one or more attributes related to an entitlement. An entitlement may contain an EntitlementType, which states what can be done at a gross level, e.g. that a party can make markets. It may be limited further within EntitlementGrp, e.g. that such market making is allowed only for a list of stocks. The EntitlementAttribGrp contains fine details clarifying or limiting the EntitlementType, e.g. that such market making must be conducted with a specific minimum quote size and a specific maximum spread.	
->->->->1777	**NoEntitlementAttrib**	N	Number of entitlement attributes.	
->->->->->1778	**EntitlementAttribType**	N	Required if NoEntitlementAttrib(1777) > 0.	
->->->->->->1779	**EntitlementAttribDatatype**	N	If specified, and this is an attribute published by FPL in the external code list, this must agree with the published datatype.	
->->->->->->1780	**EntitlementAttribValue**	N	Required if NoEntitlementAttrib(1777) > 0.	
->->->->->->178	**EntitlementAttribCurrency**	N	Currency for EntitlementAttribValue(1780). Can be used if these fields	

1			represent a price, price offset, or amount.	
->->->1776	**EntitlementID**	N	Unique identifier for a specific NoEntitlements(1773) repeating group instance.	
->->->	**InstrumentScopeGrp**	N	Repeating group of InstrumentScope Components. Used to specify the instruments to which a request applies.	
->->->->1656	**NoInstrumentScopes**	N	Number of instrument scopes.	
->->->->->1535	**InstrumentScopeOperator**	N	Required when NoInstrumentScopes > 0.	
->->->->->	**InstrumentScope**	N	Used to specify the instrument	
->->->->->->1536	**InstrumentScopeSymbol**	N	Used to limit instrument scope to specified symbol. See Symbol(55) field for description.	
->->->->->->1537	**InstrumentScopeSymbolSfx**	N	Used to limit instrument scope to specified symbol suffix. See SymbolSfx(65) field for description.	
->->->->->->1538	**InstrumentScopeSecurityID**	N	Used to limit instrument scope to specified security identifier. See SecurityID(48) field for description.	
->->->->->->1539	**InstrumentScopeSecurityIDSource**	N	Used to limit instrument scope to specified security identifier source. See SecurityIDSource(22) field for description.	
->->->->->->	**InstrumentScopeSecurityAltIDGrp**	N	Alternative SecurityIDs for an instrument specified in the InstrumentScope.	
->->->->->->1543	**InstrumentScopeProduct**	N	Used to limit instrument scope to specified instrument product category. See Product (460) field for description.	
->->->->->->1544	**InstrumentScopeProductComplex**	N	Used to limit instrument scope to specified product complex. See ProductComplex(1227) field for description.	
->->->->->->154	**InstrumentScopeSecurityGroup**	N	Used to limit instrument scope to specified security group. See SecurityGroup(1151) field for description.	

5				
->->->->->->1546	InstrumentScopeCFICode	N	Used to limit instrument scope to specified CFICode. See CFICode(461) field for description.	
->->->->->->1547	InstrumentScopeSecurityType	N	Used to limit instrument scope to specified security type. See SecurityType(167) field for description).	
->->->->->->1548	InstrumentScopeSecuritySubType	N	Used to limit instrument scope to specified security sub-type. See SecuritySubType(762) field for description.	
->->->->->->1549	InstrumentScopeMaturityMonthYear	N	Used to limit instrument scope to specified maturity month and year. See MaturityMonthYear(200) field for description.	
->->->->->->1550	InstrumentScopeMaturityTime	N	Used to limit instrument scope to specified maturity time. See MaturityTime(1079) field for description.	
->->->->->->1551	InstrumentScopeRestructuringType	N	Used to limit instrument scope to specified restructuring type. See RestructuringType(1449) field for description.	
->->->->->->1552	InstrumentScopeSeniority	N	Used to limit instrument scope to specified seniority type. See Seniority(1450) field for description.	
->->->->->->1553	InstrumentScopePutOrCall	N	Used to limit instrument scope to puts or calls. See PutOrCall(201) field for description.	
->->->->->->1554	InstrumentScopeFlexibleIndicator	N	Used to limit instrument scope to securities that can be defined using flexible terms or not. See FlexibleIndicator(1244) field for description.	
->->->->->->1555	InstrumentScopeCouponRate	Y	Used to limit instrument scope to specified coupon rate. See CouponRate(223) field for description.	
->->->->->->1616	InstrumentScopeSecurityExchange	N	Used to limit instrument scope to specified security exchange. See SecurityExchange(207) field for description.	
->->-	InstrumentScopeSecurityDesc	N	Used to limit instrument scope to	

>->->->1556			specified security description. See SecurityDesc(107) field for description.	
->->->->->1620	**InstrumentScopeEncodedSecurityDescLen**	N	Byte length of encoded (non-ASCII characters) InstrumentScopeEncodedSecurityDesc (1621) field	
->->->->->1621	**InstrumentScopeEncodedSecurityDesc**	N	Encoded (non-ASCII characters) representation of the InstrumentScopeSecurityDesc (1556) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the InstrumentScopeSecurityDesc field.	
->->->->->1557	**InstrumentScopeSettlType**	N	Can be used to specify FX tenors.	
->->->	**MarketSegmentScopeGrp**	N	conveys a list of markets and, optionally, their market segments. Note that the component MarketSegmentGrp exists, but is not useful for this purpose, as it conveys additional information not appropriate in this context.	
->->->1782	**EntitlementStartDate**	N	Indicates the starting date of the entitlement.	
->->->1783	**EntitlementEndDate**	N	Indicates the starting date of the entitlement.	
60	**TransactTime**	N	Timestamp when the business transaction represented by the message occurred.	
58	**Text**	N	Free format text string (Note: this field does not have a specified maximum length)	
1328	**RejectText**	N	Identifies the reason for rejection.	
	StandardTrailer	Y	The standard FIX message trailer	

8.11 Party Entitlements Update Report (35=CZ)

8.11.1 Party Entitlements Update Report Message Structure Diagram

The following diagram illustrates the structure of the Party Entitlements Update Report message:

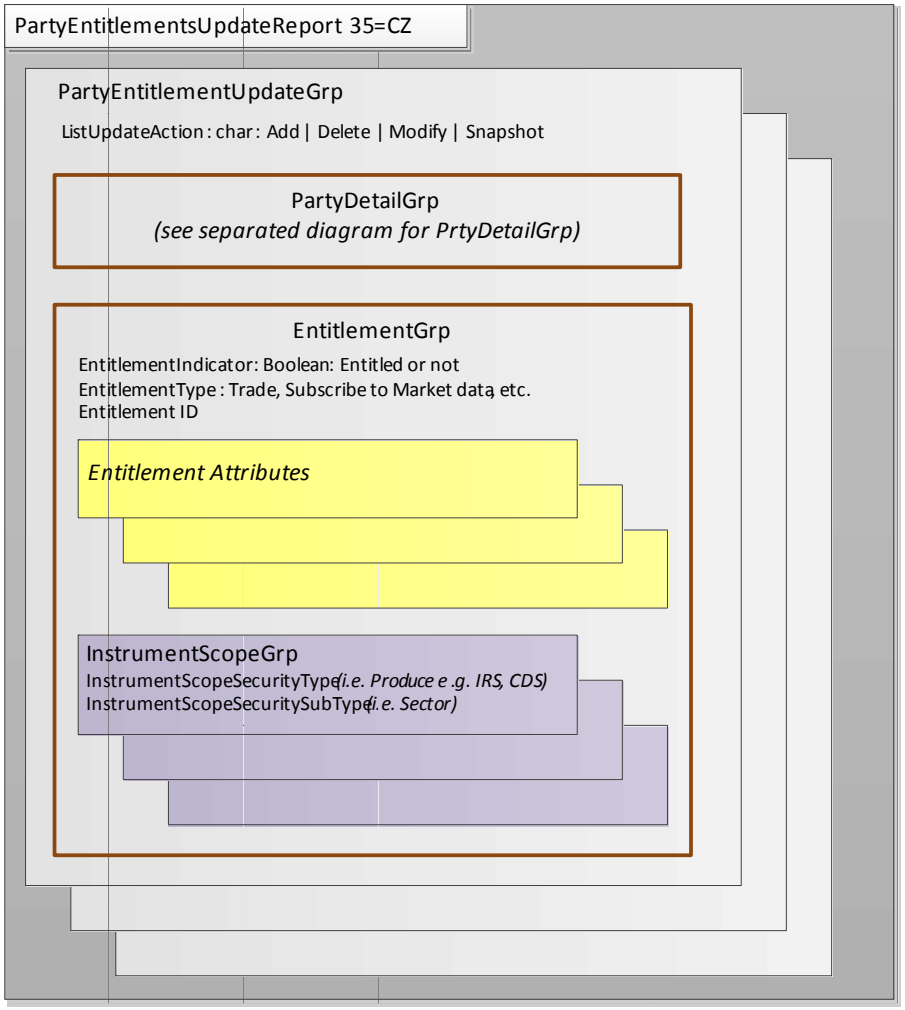


Figure 23: Party Entitlements Update Report 35=CZ Message Structure

8.11.2 Party Entitlements Update Report Message Definition

PartyEntitlementsUpdateReport (CZ)			Execution Venue -> Sell-Side	
The PartyEntitlementsUpdateReport(35=CZ) is used to convey incremental changes to party entitlements. It is similar to the PartyEntitlementsReport(35=CV). This message uses the PartyEntitlementsUpdateGrp component which includes the ability to specify an update action using ListUpdateAction(1324).				
Tag	**Field Name**	**Req'd**	**Description**	**Comment**
	StandardHeader	Y	MsgType=CZ	
	ApplicationSequenceControl	N	The ApplicationSequenceControl is used for application sequencing and recovery. Consisting of ApplSeqNum (1181), ApplID (1180), ApplLastSeqNum (1350), and ApplResendFlag (1352), FIX application messages that carries this component block will be able to use application level sequencing. ApplID, ApplSeqNum and ApplLastSeqNum fields identify the application id, application sequence number and the previous application sequence number (in case of intentional gaps) on each application message that carries this block.	
1771	**EntitlementReportID**	Y	Identifier for the PartyEntitlementsReport(35=CV).	
1770	**EntitlementRequestID**	N	Conditionally required when responding to PartyEntitlementsRequest(35=CU).	
1512	**TotNoParties**	N	Total number of PartyListGrp returned.	
893	**LastFragment**	N	Indicates whether this message is the last in a sequence of messages for those messages that support fragmentation, such as Allocation Instruction, Mass Quote, Security List, Derivative Security List	
	PartyEntitlementUpdateGrp	Y	Specifies the updated entitlements to be enforced for the given party(-ies) and related party(-ies).	
1772	**NoPartyEntitlements**	N	Number of party entitlement values.	
->1324	**ListUpdateAction**	N	Required if NoPartyEntitlements(1772).	
->	**PartyDetailGrp**	N	Optional when ListUpdateAction(1324) = M(Modify) or D(Delete) and EntitlementRefID(1885) is provided.	
->->1671	**NoPartyDetails**	N	Number of party details.	
->->-	**PartyDetailID**	N	The identification of the party.	

>1691			Required when NoPartyDetails(1671) > 0.	
->->->1692	**PartyDetailIDSource**	N	Used to identify source of PartyID value (e.g. BIC). Required when NoPartyDetails(1671) > 0.	
->->->1693	**PartyDetailRole**	N	Identifies the type of PartyID (e.g. Executing Broker). Required when NoPartyDetails(1671) > 0.	
->->->1674	**PartyDetailRoleQualifier**	N	Qualifies the value of PartyRole(452)	
->1883	**EntitlementStatus**	N		
->->->	**PartyDetailSubGrp**	N	Additional party sub-identifiers	
->->->	**PartyDetailAltIDGrp**	N	Optionally used to specify alternate IDs to identify the party specified.	
->->->	**RelatedPartyDetailGrp**	N	May not be specified in PartyDetailsListUpdateReport(35=CK) if ListUpdateAction(1324) = D(Delete)	
->->->->1562	**NoRelatedPartyDetailID**	N	Number of related party detail identifiers.	
->->->->->1563	**RelatedPartyDetailID**	N	Required if NoRelatedPartyDetails > 0.	
->->->->->1564	**RelatedPartyDetailIDSource**	N	Required if NoRelatedPartyDetails > 0.	
->->->->->1565	**RelatedPartyDetailRole**	N	Required if NoRelatedPartyDetails > 0.	
->->->->->1675	**RelatedPartyDetailRoleQualifier**	N	Qualifies the value of RelatedPartyRole(1565)	
->->->->->	**RelatedPartyDetailSubGrp**	N	PartySubGrp for related parties.	
->->->->->	**RelatedPartyDetailAltIDGrp**	N	Alternative identifiers for parties related to the party specified in the PartyDetailGrp.	
->->->->->	**PartyRelationshipGrp**	N	Repeating group of party relationships.	
->->->->->->151	**NoPartyRelationships**	N	Number of party relationships.	

4				
->->->->->1515	PartyRelationship	N	Identifies the type of party relationship requested. Required if NoPartyRelationships > 0.	
->->->1672	PartyDetailStatus	N	Specifies the status of the party information, whether active or suspended (inactive).	
->	**EntitlementGrp**	N	Optional when ListUpdateAction(1324) = M(Modify) or D(Delete) and EntitlementRefID(1885) is provided.	
->->1773	NoEntitlements	N	Number of entitlement values.	
->->->1774	EntitlementIndicator	N	Required if NoEntitlements(1773) > 0.	
->->->1775	EntitlementType	N	Absence of this field indicates the meaning of the entitlement is implicit.	
->->->	**EntitlementAttribGrp**	N	conveys a list of one or more attributes related to an entitlement. An entitlement may contain an EntitlementType, which states what can be done at a gross level, e.g. that a party can make markets. It may be limited further within EntitlementGrp, e.g. that such market making is allowed only for a list of stocks. The EntitlementAttribGrp contains fine details clarifying or limiting the EntitlementType, e.g. that such market making must be conducted with a specific minimum quote size and a specific maximum spread.	
->->->->1777	NoEntitlementAttrib	N	Number of entitlement attributes.	
->->->->->1778	EntitlementAttribType	N	Required if NoEntitlementAttrib(1777) > 0.	
->->->->->1779	EntitlementAttribDatatype	N	If specified, and this is an attribute published by FPL in the external code list, this must agree with the published datatype.	
->->->->->1780	EntitlementAttribValue	N	Required if NoEntitlementAttrib(1777) > 0.	
->->-	EntitlementAttribCurrency	N	Currency for	

>->->1781			EntitlementAttribValue(1780). Can be used if these fields represent a price, price offset, or amount.	
->->->1776	**EntitlementID**	N	Unique identifier for a specific NoEntitlements(1773) repeating group instance.	
->->->	**InstrumentScopeGrp**	N	Repeating group of InstrumentScope Components. Used to specify the instruments to which a request applies.	
->->->->1656	**NoInstrumentScopes**	N	Number of instrument scopes.	
->->->->->1535	**InstrumentScopeOperator**	N	Required when NoInstrumentScopes > 0.	
->->->->->	**InstrumentScope**	N	Used to specify the instrument	
->->->->->->1536	**InstrumentScopeSymbol**	N	Used to limit instrument scope to specified symbol. See Symbol(55) field for description.	
->->->->->->1537	**InstrumentScopeSymbolSfx**	N	Used to limit instrument scope to specified symbol suffix. See SymbolSfx(65) field for description.	
->->->->->->1538	**InstrumentScopeSecurityID**	N	Used to limit instrument scope to specified security identifier. See SecurityID(48) field for description.	
->->->->->->1539	**InstrumentScopeSecurityIDSource**	N	Used to limit instrument scope to specified security identifier source. See SecurityIDSource(22) field for description.	
->->->->->->	**InstrumentScopeSecurityAltIDGrp**	N	Alternative SecurityIDs for an instrument specified in the InstrumentScope.	
->->->->->->1543	**InstrumentScopeProduct**	N	Used to limit instrument scope to specified instrument product category. See Product (460) field for description.	
->->->->->->1544	**InstrumentScopeProductComplex**	N	Used to limit instrument scope to specified product complex. See ProductComplex(1227) field for description.	
->->->->->-	**InstrumentScopeSecurityGroup**	N	Used to limit instrument scope to specified security group. See	

>->1545			SecurityGroup(1151) field for description.	
->->->->->1546	**InstrumentScopeCFICode**	N	Used to limit instrument scope to specified CFICode. See CFICode(461) field for description.	
->->->->->1547	**InstrumentScopeSecurityType**	N	Used to limit instrument scope to specified security type. See SecurityType(167) field for description).	
->->->->->1548	**InstrumentScopeSecuritySubType**	N	Used to limit instrument scope to specified security sub-type. See SecuritySubType(762) field for description.	
->->->->->1549	**InstrumentScopeMaturityMonthYear**	N	Used to limit instrument scope to specified maturity month and year. See MaturityMonthYear(200) field for description.	
->->->->->1550	**InstrumentScopeMaturityTime**	N	Used to limit instrument scope to specified maturity time. See MaturityTime(1079) field for description.	
->->->->->1551	**InstrumentScopeRestructuringType**	N	Used to limit instrument scope to specified restructuring type. See RestructuringType(1449) field for description.	
->->->->->1552	**InstrumentScopeSeniority**	N	Used to limit instrument scope to specified seniority type. See Seniority(1450) field for description.	
->->->->->1553	**InstrumentScopePutOrCall**	N	Used to limit instrument scope to puts or calls. See PutOrCall(201) field for description.	
->->->->->1554	**InstrumentScopeFlexibleIndicator**	N	Used to limit instrument scope to securities that can be defined using flexible terms or not. See FlexibleIndicator(1244) field for description.	
->->->->->1555	**InstrumentScopeCouponRate**	Y	Used to limit instrument scope to specified coupon rate. See CouponRate(223) field for description.	
->->->->->161	**InstrumentScopeSecurityExchange**	N	Used to limit instrument scope to specified security exchange. See SecurityExchange(207) field for description.	

6				
->->->->->1556	**InstrumentScopeSecurityDesc**	N	Used to limit instrument scope to specified security description. See SecurityDesc(107) field for description.	
->->->->->1620	**InstrumentScopeEncodedSecurityDescLen**	N	Byte length of encoded (non-ASCII characters) InstrumentScopeEncodedSecurityDesc (1621) field	
->->->->->1621	**InstrumentScopeEncodedSecurityDesc**	N	Encoded (non-ASCII characters) representation of the InstrumentScopeSecurityDesc (1556) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the InstrumentScopeSecurityDesc field.	
->->->->->1557	**InstrumentScopeSettlType**	N	Can be used to specify FX tenors.	
->->->	**MarketSegmentScopeGrp**	N	conveys a list of markets and, optionally, their market segments. Note that the component MarketSegmentGrp exists, but is not useful for this purpose, as it conveys additional information not appropriate in this context.	
->->->1782	**EntitlementStartDate**	N	Indicates the starting date of the entitlement.	
->->->1783	**EntitlementEndDate**	N	Indicates the starting date of the entitlement.	
->1885	**EntitlementRefID**	N	Optional when PartyDetailGrp is provided or ListUpdateAction(1324) = A(Add).	
60	**TransactTime**	N	Timestamp when the business transaction represented by the message occurred.	
58	**Text**	N	Free format text string (Note: this field does not have a specified maximum length)	
	StandardTrailer	Y	The standard FIX message trailer	

8.12 Party Entitlements Definition Request (35=DA)

8.12.1 Party Entitlements Definition Request Message Structure Diagram

The following diagram illustrates the structure of the Party Entitlements Definition Request message:

Figure 24: Party Entitlements Definition Request 35=DA Message Structure

8.12.2 Party Entitlements Definition Request Example

In this example the sell-side sends an entitlement definition request containing data for a single user to the Execution Venue.
The below table contains the business level attributes of the entitlement definition:

Attribute	Values
Sell-side Identifier	Bank-1
Client Account	Account-1
Primary Sales Person User ID	James-Bradley-ID
Primary Sales Person Name	James Bradley

Product	IRS
Sector Name1	SONIA
Buy-side User ID	User-1
Buy side user name	Jon A. Smith
Firm	Hedge fund
LEI	LZ123
Enablement Type	Trade
Status	Enabled
Last Modified	12-Feb-2013 10:52:32
Modified By	Kate Middleton
Tier	Level 1
Allow RFQ trading	Y
Allow Clearing House CME	Y

Table 13: Example 2 – Sell-Side sends an entitlement request report

8.12.2.1 Party Entitlements Definition Request FIXML

The following is the FIXML message tags and values that are expected in this example:

```
<?xml version="1.0" encoding="UTF-8"?>
<FIXML xmlns="http://www.fixprotocol.org/FIXML-5-0-SP2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.fixprotocol.org/FIXML-5-0-SP2 ./Fixml/fixml-main-5-0-SP2.xsd">
   <PtyEntlmntDefReq  ReqID="1">
      <ReqPty ID="Bank-1" Src="D" R="1">
         <Sub ID="Kate Middleton" Typ="2"/>
      </ReqPty>
      <PtyEntlmntUpd  ListUpdActn="A" >
         <PtyDetl ID="User-1" Src="D" R="3">
            <Sub ID="Jon A. Smith" Typ="9"/> <!-- Contact name-->
            <Sub ID="Hedge Fund" Typ="1"/> <!-- firm -->
            <Sub ID="jon.a.smith@HedgeFund.com" Typ="8" /> <!-- email -->
            <AltPty ID="LZ123" Src="N"/>
            <ReltdPtyDetl ID="Bank-1" Src="D" R="1">
               <Sub ID="Account-1" Typ="19" /> <!-- Client account -->
               <Sub ID="James Bradley" Typ="9" /> <!-- Sales person -->
               <Sub ID="James-Bradley-ID" Typ="2" /> <!-- Sales person ID -->
               <Rltnshp Rltnshp="4" /> <!-- Trade through -->
            </ReltdPtyDetl>
         </PtyDetl>
         <Entlmnt Ind="Y" Typ="0" >
            <Attrib Typ="4000" Datatyp="13" Value="Y" /> <!-- Outstanding buy-side request -->
            <Attrib Typ="1500" Datatyp="5" Value="Level-1" /> <!-- FIX tag MDStreamID(1500)-->
            <Attrib Typ="4003" Datatyp="13" Value="Y" /> <!-- Allow RFQ -->
            <Attrib Typ="4004" Datatyp="13" Value="Y" /> <!-- Allow multi-dealer -->
            <Attrib Typ="4005" Datatyp="13" Value="N" /> <!-- Allow single-dealer -->
            <Attrib Typ="4006" Datatyp="13" Value="N" /> <!-- Allow block-bilateral -->
            <Attrib Typ="4007" Datatyp="13" Value="Y" /> <!-- Allow switch -->
            <Attrib Typ="4008" Datatyp="13" Value="N" /> <!-- Allow USD -->
            <Attrib Typ="4009" Datatyp="13" Value="Y" /> <!-- Entitled to trade instruments
that are cleared through CME -->
            <InstrmtScope CpnRt="1" SecTyp="IRS" SecSubTyp="SONIA"/>
         </Entlmnt>
      </PtyEntlmntUpd>
   </PtyEntlmntDefReq>
</FIXML>
```

8.12.3 Party Entitlements Definition Request Message Definition

PartyEntitlementsDefinitionRequest (DA)			Sell-Side -> Execution Venue	
The PartyEntitlementsDefinitionRequest(35=DA) is used for defining new entitlements, and modifying or deleting existing entitlements for the specified party(-ies).				
Tag	**Field Name**	**Req'd**	**Description**	**Comment**
1770	**EntitlementRequestID**	Y	Unique identifier for PartyEntitlementsRequest(35=CU).	
	RequestingPartyGrp	N	Can be used to identify the party making the request and their role.	
1657	**NoRequestingPartyIDs**	N	Number of requesting party identifiers.	
->1658	**RequestingPartyID**	N	Required when NoRequestingPartyIDs > 0.	
->1659	**RequestingPartyIDSource**	N	Required when NoRequestingPartyIDs > 0.	
->1660	**RequestingPartyRole**	N	Required when NoRequestingPartyIDs > 0.	
->	**RequestingPartySubGrp**	N	Sub identifiers for the requesting party.	
	PartyEntitlementUpdateGrp	Y	Specifies the entitlements to be defined, modified or deleted for the given party(-ies) and related party(-ies).	
1772	**NoPartyEntitlements**	N	Number of party entitlement values.	
->1324	**ListUpdateAction**	N	Required if NoPartyEntitlements(1772).	
->	**PartyDetailGrp**	N	Optional when ListUpdateAction(1324) = M(Modify) or D(Delete) and EntitlementRefID(1885) is provided.	
->->1671	**NoPartyDetails**	N	Number of party details.	
->->->1691	**PartyDetailID**	N	The identification of the party. Required when NoPartyDetails(1671) > 0.	
->->->1692	**PartyDetailIDSource**	N	Used to identify source of PartyID value (e.g. BIC). Required when NoPartyDetails(1671) > 0.	
->->->1693	**PartyDetailRole**	N	Identifies the type of PartyID (e.g. Executing Broker). Required when NoPartyDetails(1671) > 0.	
->->->1674	**PartyDetailRoleQualifier**	N	Qualifies the value of PartyRole(452)	
->->->	**PartyDetailSubGrp**	N	Additional party sub-identifiers	

->->->	**PartyDetailAltIDGrp**	N	Optionally used to specify alternate IDs to identify the party specified.	
->->->	**RelatedPartyDetailGrp**	N	May not be specified in PartyDetailsListUpdateReport(35=CK) if ListUpdateAction(1324) = D(Delete)	
->->->->1562	**NoRelatedPartyDetailID**	N	Number of related party detail identifiers.	
->->->->->1563	**RelatedPartyDetailID**	N	Required if NoRelatedPartyDetails > 0.	
->->->->->1564	**RelatedPartyDetailIDSource**	N	Required if NoRelatedPartyDetails > 0.	
->->->->->1565	**RelatedPartyDetailRole**	N	Required if NoRelatedPartyDetails > 0.	
->->->->->1675	**RelatedPartyDetailRoleQualifier**	N	Qualifies the value of RelatedPartyRole(1565)	
->->->->->	**RelatedPartyDetailSubGrp**	N	PartySubGrp for related parties.	
->->->->->	**RelatedPartyDetailAltIDGrp**	N	Alternative identifiers for parties related to the party specified in the PartyDetailGrp.	
->->->->->	**PartyRelationshipGrp**	N	Repeating group of party relationships.	
->->->->->->1514	**NoPartyRelationships**	N	Number of party relationships.	
->->->->->->->1515	**PartyRelationship**	N	Identifies the type of party relationship requested. Required if NoPartyRelationships > 0.	
->->->1672	**PartyDetailStatus**	N	Specifies the status of the party information, whether active or suspended (inactive).	
->	**EntitlementGrp**	N	Optional when ListUpdateAction(1324) = M(Modify) or D(Delete) and EntitlementRefID(1885) is provided.	
->->1773	**NoEntitlements**	N	Number of entitlement values.	

->->->1774	**EntitlementIndicator**	N	Required if NoEntitlements(1773) > 0.	
->->->1775	**EntitlementType**	N	Absence of this field indicates the meaning of the entitlement is implicit.	
->->->	**EntitlementAttribGrp**	N	conveys a list of one or more attributes related to an entitlement. An entitlement may contain an EntitlementType, which states what can be done at a gross level, e.g. that a party can make markets. It may be limited further within EntitlementGrp, e.g. that such market making is allowed only for a list of stocks. The EntitlementAttribGrp contains fine details clarifying or limiting the EntitlementType, e.g. that such market making must be conducted with a specific minimum quote size and a specific maximum spread.	
->->->->1777	**NoEntitlementAttrib**	N	Number of entitlement attributes.	
->->->->->1778	**EntitlementAttribType**	N	Required if NoEntitlementAttrib(1777) > 0.	
->->->->->1779	**EntitlementAttribDatatype**	N	If specified, and this is an attribute published by FPL in the external code list, this must agree with the published datatype.	
->->->->->1780	**EntitlementAttribValue**	N	Required if NoEntitlementAttrib(1777) > 0.	
->->->->->1781	**EntitlementAttribCurrency**	N	Currency for EntitlementAttribValue(1780). Can be used if these fields represent a price, price offset, or amount.	
->->->1776	**EntitlementID**	N	Unique identifier for a specific NoEntitlements(1773) repeating group instance.	
->->->	**InstrumentScopeGrp**	N	Repeating group of InstrumentScope Components. Used to specify the instruments to which a request applies.	
->->->->1656	**NoInstrumentScopes**	N	Number of instrument scopes.	
->->->->->	**InstrumentScopeOperator**	N	Required when NoInstrumentScopes > 0.	

>1535				
->->->->->	**InstrumentScope**	N	Used to specify the instrument	
->->->->->1536	**InstrumentScopeSymbol**	N	Used to limit instrument scope to specified symbol. See Symbol(55) field for description.	
->->->->->1537	**InstrumentScopeSymbolSfx**	N	Used to limit instrument scope to specified symbol suffix. See SymbolSfx(65) field for description.	
->->->->->1538	**InstrumentScopeSecurityID**	N	Used to limit instrument scope to specified security identifier. See SecurityID(48) field for description.	
->->->->->1539	**InstrumentScopeSecurityIDSource**	N	Used to limit instrument scope to specified security identifier source. See SecurityIDSource(22) field for description.	
->->->->->	**InstrumentScopeSecurityAltIDGrp**	N	Alternative SecurityIDs for an instrument specified in the InstrumentScope.	
->->->->->1543	**InstrumentScopeProduct**	N	Used to limit instrument scope to specified instrument product category. See Product (460) field for description.	
->->->->->1544	**InstrumentScopeProductComplex**	N	Used to limit instrument scope to specified product complex. See ProductComplex(1227) field for description.	
->->->->->1545	**InstrumentScopeSecurityGroup**	N	Used to limit instrument scope to specified security group. See SecurityGroup(1151) field for description.	
->->->->->1546	**InstrumentScopeCFICode**	N	Used to limit instrument scope to specified CFICode. See CFICode(461) field for description.	
->->->->->1547	**InstrumentScopeSecurityType**	N	Used to limit instrument scope to specified security type. See SecurityType(167) field for description).	
->->->->->154	**InstrumentScopeSecuritySubType**	N	Used to limit instrument scope to specified security sub-type. See SecuritySubType(762) field for description.	

8				
->->->->->1549	**InstrumentScopeMaturityMonthYear**	N	Used to limit instrument scope to specified maturity month and year. See MaturityMonthYear(200) field for description.	
->->->->->1550	**InstrumentScopeMaturityTime**	N	Used to limit instrument scope to specified maturity time. See MaturityTime(1079) field for description.	
->->->->->1551	**InstrumentScopeRestructuringType**	N	Used to limit instrument scope to specified restructuring type. See RestructuringType(1449) field for description.	
->->->->->1552	**InstrumentScopeSeniority**	N	Used to limit instrument scope to specified seniority type. See Seniority(1450) field for description.	
->->->->->1553	**InstrumentScopePutOrCall**	N	Used to limit instrument scope to puts or calls. See PutOrCall(201) field for description.	
->->->->->1554	**InstrumentScopeFlexibleIndicator**	N	Used to limit instrument scope to securities that can be defined using flexible terms or not. See FlexibleIndicator(1244) field for description.	
->->->->->1555	**InstrumentScopeCouponRate**	Y	Used to limit instrument scope to specified coupon rate. See CouponRate(223) field for description.	
->->->->->1616	**InstrumentScopeSecurityExchange**	N	Used to limit instrument scope to specified security exchange. See SecurityExchange(207) field for description.	
->->->->->1556	**InstrumentScopeSecurityDesc**	N	Used to limit instrument scope to specified security description. See SecurityDesc(107) field for description.	
->->->->->1620	**InstrumentScopeEncodedSecurityDescLen**	N	Byte length of encoded (non-ASCII characters) InstrumentScopeEncodedSecurityDesc (1621) field	
->->->->->1621	**InstrumentScopeEncodedSecurityDesc**	N	Encoded (non-ASCII characters) representation of the InstrumentScopeSecurityDesc (1556) field in the encoded format specified via the MessageEncoding (347) field. If	

			used, the ASCII (English) representation should also be specified in the InstrumentScopeSecurityDesc field.	
->->->->->1557	**InstrumentScopeSettlType**	N	Can be used to specify FX tenors.	
->->->	**MarketSegmentScopeGrp**	N	conveys a list of markets and, optionally, their market segments. Note that the component MarketSegmentGrp exists, but is not useful for this purpose, as it conveys additional information not appropriate in this context.	
->->->1782	**EntitlementStartDate**	N	Indicates the starting date of the entitlement.	
->->->1783	**EntitlementEndDate**	N	Indicates the starting date of the entitlement.	
->1885	**EntitlementRefID**	N	Optional when PartyDetailGrp is provided or ListUpdateAction(1324) = A(Add).	
58	**Text**	N	Free format text string (Note: this field does not have a specified maximum length)	
	StandardTrailer	Y	The standard FIX message trailer	

8.13 Party Entitlements Definition Request Acknowledge (35=DB)

8.13.1 Party Entitlements Definition Request Ack Message Structure Diagram

The following diagram illustrates the structure of the Party Entitlements Definition Request Ack message:

Figure 25: Party Entitlements Definition Request Ack 35=DB Message Structure

8.13.2 Party Entitlements Definition Request Acknowledgement Message Definition

PartyEntitlementsDefinitionRequestAck (DB)			Execution Venue -> Sell-Side	
The PartyEntitlementsDefinitionRequestAck(35=DB) is used as a response to the PartyEntitlemensDefinitionRequest(35=DA) to accept (with or without changes) or reject the definition of party entitlements.				
Tag	**Field Name**	**Req'd**	**Description**	**Comment**
1882	**EntitlementRequestStatus**	Y	Status of party entitlements definition request.	
1881	**EntitlementRequestResult**	N	Result of risk limit definition request.	

	RequestingPartyGrp	N	Identifies the party making the request.	
1657	**NoRequestingPartyIDs**	N	Number of requesting party identifiers.	
->1658	**RequestingPartyID**	N	Required when NoRequestingPartyIDs > 0.	
->1659	**RequestingPartyIDSource**	N	Required when NoRequestingPartyIDs > 0.	
->1660	**RequestingPartyRole**	N	Required when NoRequestingPartyIDs > 0.	
->	**RequestingPartySubGrp**	N	Sub identifiers for the requesting party.	
	PartyEntitlementAckGrp	N	The PartyEntitlementAckGrp component is used in the PartyEntitlementsDefinitionRequestAck(35=DB) message to provide the status of each action (add, modify or delete) requested by the PartyEntitlementsDefinitionRequest(35=DA) message.	
1772	**NoPartyEntitlements**	N	Number of party entitlement values.	
->1324	**ListUpdateAction**	N	Required if NoPartyEntitlements(1772).	
->1883	**EntitlementStatus**	N	Required if NoPartyEntitlements(1772).	
->1884	**EntitlementResult**	N	Result of entitlement definition for one party.	
->1328	**RejectText**	N	Identifies the reason for rejection.	
->	**PartyDetailGrp**	N	Optional when ListUpdateAction(1324) = M(Modify) or D(Delete) and EntitlementRefID(1885) is provided.	
->->1671	**NoPartyDetails**	N	Number of party details.	
->->->1691	**PartyDetailID**	N	The identification of the party. Required when NoPartyDetails(1671) > 0.	
->->->1692	**PartyDetailIDSource**	N	Used to identify source of PartyID value (e.g. BIC). Required when NoPartyDetails(1671) > 0.	
->->->1693	**PartyDetailRole**	N	Identifies the type of PartyID (e.g. Executing Broker). Required when NoPartyDetails(1671) > 0.	
->->->1674	**PartyDetailRoleQualifier**	N	Qualifies the value of PartyRole(452)	
->-	**PartyDetailSubGrp**	N	Additional party sub-identifiers	

>->				
->->>->	**PartyDetailAltIDGrp**	N	Optionally used to specify alternate IDs to identify the party specified.	
->->>->	**RelatedPartyDetailGrp**	N	May not be specified in PartyDetailsListUpdateReport(35=CK) if ListUpdateAction(1324) = D(Delete)	
->->>->->1562	**NoRelatedPartyDetailID**	N	Number of related party detail identifiers.	
->->>->->1563	**RelatedPartyDetailID**	N	Required if NoRelatedPartyDetails > 0.	
->->>->->1564	**RelatedPartyDetailIDSource**	N	Required if NoRelatedPartyDetails > 0.	
->->>->->1565	**RelatedPartyDetailRole**	N	Required if NoRelatedPartyDetails > 0.	
->->>->->1675	**RelatedPartyDetailRoleQualifier**	N	Qualifies the value of RelatedPartyRole(1565)	
->->>->->->	**RelatedPartyDetailSubGrp**	N	PartySubGrp for related parties.	
->->>->->	**RelatedPartyDetailAltIDGrp**	N	Alternative identifiers for parties related to the party specified in the PartyDetailGrp.	
->->>->->	**PartyRelationshipGrp**	N	Repeating group of party relationships.	
->->>->->->1514	**NoPartyRelationships**	N	Number of party relationships.	
->->>->->->->1515	**PartyRelationship**	N	Identifies the type of party relationship requested. Required if NoPartyRelationships > 0.	
->->>->1672	**PartyDetailStatus**	N	Specifies the status of the party information, whether active or suspended (inactive).	
->	**EntitlementGrp**	N	Optional when ListUpdateAction(1324) = M(Modify) or D(Delete) and	

			EntitlementRefID(1885) is provided.	
->->1773	**NoEntitlements**	N	Number of entitlement values.	
->->->1774	**EntitlementIndicator**	N	Required if NoEntitlements(1773) > 0.	
->->->1775	**EntitlementType**	N	Absence of this field indicates the meaning of the entitlement is implicit.	
->->->	**EntitlementAttribGrp**	N	conveys a list of one or more attributes related to an entitlement. An entitlement may contain an EntitlementType, which states what can be done at a gross level, e.g. that a party can make markets. It may be limited further within EntitlementGrp, e.g. that such market making is allowed only for a list of stocks. The EntitlementAttribGrp contains fine details clarifying or limiting the EntitlementType, e.g. that such market making must be conducted with a specific minimum quote size and a specific maximum spread.	
->->->->1777	**NoEntitlementAttrib**	N	Number of entitlement attributes.	
->->->->->1778	**EntitlementAttribType**	N	Required if NoEntitlementAttrib(1777) > 0.	
->->->->->1779	**EntitlementAttribDatatype**	N	If specified, and this is an attribute published by FPL in the external code list, this must agree with the published datatype.	
->->->->->1780	**EntitlementAttribValue**	N	Required if NoEntitlementAttrib(1777) > 0.	
->->->->->1781	**EntitlementAttribCurrency**	N	Currency for EntitlementAttribValue(1780). Can be used if these fields represent a price, price offset, or amount.	
->->->1776	**EntitlementID**	N	Unique identifier for a specific NoEntitlements(1773) repeating group instance.	
->->->	**InstrumentScopeGrp**	N	Repeating group of InstrumentScope Components. Used to specify the	

			instruments to which a request applies.	
->->->->1656	NoInstrumentScopes	N	Number of instrument scopes.	
->->->->->	InstrumentScopeOperator	N	Required when NoInstrumentScopes > 0.	
->->->->1535	**InstrumentScope**	N	Used to specify the instrument	
->->->->->	InstrumentScopeSymbol	N	Used to limit instrument scope to specified symbol. See Symbol(55) field for description.	
->->->->->1536	InstrumentScopeSymbolSfx	N	Used to limit instrument scope to specified symbol suffix. See SymbolSfx(65) field for description.	
->->->->->1537	InstrumentScopeSecurityID	N	Used to limit instrument scope to specified security identifier. See SecurityID(48) field for description.	
->->->->->1538	InstrumentScopeSecurityIDSource	N	Used to limit instrument scope to specified security identifier source. See SecurityIDSource(22) field for description.	
->->->->->1539	**InstrumentScopeSecurityAltIDGrp**	N	Alternative SecurityIDs for an instrument specified in the InstrumentScope.	
->->->->->1543	InstrumentScopeProduct	N	Used to limit instrument scope to specified instrument product category. See Product (460) field for description.	
->->->->->1544	InstrumentScopeProductComplex	N	Used to limit instrument scope to specified product complex. See ProductComplex(1227) field for description.	
->->->->->1545	InstrumentScopeSecurityGroup	N	Used to limit instrument scope to specified security group. See SecurityGroup(1151) field for description.	
->->->->->1546	InstrumentScopeCFICode	N	Used to limit instrument scope to specified CFICode. See CFICode(461) field for description.	

->->->->->1547	InstrumentScopeSecurityType	N	Used to limit instrument scope to specified security type. See SecurityType(167) field for description).	
->->->->->1548	InstrumentScopeSecuritySubType	N	Used to limit instrument scope to specified security sub-type. See SecuritySubType(762) field for description.	
->->->->->1549	InstrumentScopeMaturityMonthYear	N	Used to limit instrument scope to specified maturity month and year. See MaturityMonthYear(200) field for description.	
->->->->->1550	InstrumentScopeMaturityTime	N	Used to limit instrument scope to specified maturity time. See MaturityTime(1079) field for description.	
->->->->->1551	InstrumentScopeRestructuringType	N	Used to limit instrument scope to specified restructuring type. See RestructuringType(1449) field for description.	
->->->->->1552	InstrumentScopeSeniority	N	Used to limit instrument scope to specified seniority type. See Seniority(1450) field for description.	
->->->->->1553	InstrumentScopePutOrCall	N	Used to limit instrument scope to puts or calls. See PutOrCall(201) field for description.	
->->->->->1554	InstrumentScopeFlexibleIndicator	N	Used to limit instrument scope to securities that can be defined using flexible terms or not. See FlexibleIndicator(1244) field for description.	
->->->->->1555	InstrumentScopeCouponRate	Y	Used to limit instrument scope to specified coupon rate. See CouponRate(223) field for description.	
->->->->->1616	InstrumentScopeSecurityExchange	N	Used to limit instrument scope to specified security exchange. See SecurityExchange(207) field for description.	
->->->->->1556	InstrumentScopeSecurityDesc	N	Used to limit instrument scope to specified security description. See SecurityDesc(107) field for description.	
->->->	InstrumentScopeEncodedSecurityDescLen	N	Byte length of encoded (non-ASCII characters)	

>->->1620			InstrumentScopeEncodedSecurityDesc (1621) field	
->->->->->1621	**InstrumentScopeEncodedSecurityDesc**	N	Encoded (non-ASCII characters) representation of the InstrumentScopeSecurityDesc (1556) field in the encoded format specified via the MessageEncoding (347) field. If used, the ASCII (English) representation should also be specified in the InstrumentScopeSecurityDesc field.	
->->->->->1557	**InstrumentScopeSettlType**	N	Can be used to specify FX tenors.	
->->->	**MarketSegmentScopeGrp**	N	conveys a list of markets and, optionally, their market segments. Note that the component MarketSegmentGrp exists, but is not useful for this purpose, as it conveys additional information not appropriate in this context.	
->->->1782	**EntitlementStartDate**	N	Indicates the starting date of the entitlement.	
->->->1783	**EntitlementEndDate**	N	Indicates the starting date of the entitlement.	
->1885	**EntitlementRefID**	N	Optional when PartyDetailGrp is provided or ListUpdateAction(1324) = A(Add).	
58	**Text**	N	Free format text string (Note: this field does not have a specified maximum length)	
	StandardTrailer	Y	The standard FIX message trailer	

Appendix I – Entitlement Attributes

The following table summarises the entitlement attributes that are required for the scope of this document:

Attribute Name	Datatype	Description
Tier	String	Price tier level at which the user is able to trade
RFQ	Boolean	Allow sending request for quote (RFQ): Y | N
Multi Dealer RFQ	Boolean	Allow sending multi-dealer RFQ: Y | N
Single Dealer RFQ	Boolean	Allow sending single-dealer RFQ: Y | N
Block bilateral trading	Boolean	Allow block bilateral trading: Y | N
List Trading	Boolean	Allow List Trading: Y | N
Always Last Look	Boolean	Always require a last look: Y | N
Limit Order	Boolean	Allow click to trade: Y | N
Target Price	Boolean	Allow click to Target Price: Y | N
Non-Inventory	Boolean	Allow Non-Inventory Trading: Y | N
Customizes Swap	Boolean	Allow Sending RFQ for customized swap: Y | N
Firm Price	Boolean	Forwarding of the Firm price to the buy-side: Y | N
Flexible Settlement Date	Boolean	Allow update of the settlement date: Y | N
Block Post Trade Allocation	Boolean	Block post trade allocation: Y | N
Single Trade Limit	Qty	Limit quantity for a single trade (RFQ or order)
Single Trade Limit Currency	Currency	Currency in which Single Trade Limit is specified
Clearing House <XXX>	Boolean	Specify whether clearing can occur through selected clearing houses e.g. Clearing House CME = Y Clearing House LCH=N Clearing House LCH(FCM)=Y Clearing House LCH(US) = N
Direct Clearing Member	Boolean	Indicates if the user is a direct clearing member or not
Trade At Close List	Boolean	Allow list trading at market close
Allow Unseasoned Trading	Boolean	Allow user to trade unseasoned securities (Override constraint preventing US market participants to trade European Bonds within 45 days of issuance)
Allow Leave Order Trading	Boolean	Allow leave order trading
Allow Direct Orders	Boolean	Allow user to place orders directly at an external platform
Allow Cross Trading	Boolean	Allow switch trading of different types of bonds (for example corporate v/s government bonds)
Allow Cross Currency	Boolean	Allow switch trading of bonds with different currency denominations
Allow Zero-Sized Trading	Boolean	Allow user to respond to quotes with zero bid or offer sizes
Include In Email List	Boolean	Flag to include user in email notifications
Allow Structured Supra	Boolean	Allow trading of structured Supranational securities
Allow Rule 144A	Boolean	Allow user to trade in accordance with Rule 144A (SEC defined rule for trading of securities with Qualified Institutional Buyers)

Appendix II – Document Figures

Appendix III – Document Tables