



FIX High Performance Working Group

Application Layer Extensions

Part 1

February 11, 2013

Revision 0.6

Proposal Status: ~~Public comment~~Approved

For Global Technical Committee Governance Internal Use Only

Submission Date	January 31, 2013	Control Number	EP188
Submission Status	Public comment <u>Approved</u>	Ratified Date	Jan. 31, 2014
Primary Contact Person	Hanno Klein	Release Identifier	5.0 SP3

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 OR CONSEQUENTIAL (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 SPECIFICATION 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") OF/FOR THE PROPOSAL AND A RELEASE NUMBER.

No proprietary or ownership interest of any kind is granted with respect to the FIX Protocol (or any rights therein).

Copyright 2003-2014 FIX Protocol Limited, all rights reserved.

Table of Contents

Document History.....	6
1 Introduction	8
1.1 Background	8
1.2 Working group charter	8
1.3 Working group organization	8
1.4 This document	8
2 Business Workflow.....	9
2.1 General Requirements	9
2.2 Order State Transitions	9
2.3 Immediate Order Execution Upon Entry or Modification.....	14
2.4 Multiple Order Executions Upon Entry or Modification	15
2.5 Multiple Order Executions with Leg Executions	15
2.6 Non-Resting Orders	15
2.7 Order Request Identifier.....	16
2.8 Mass Order	17
3 Issues and Discussion Points	19
3.1 Usage of LastPx/LastQty and FillsGrp	19
3.2 Usage of FillsGrp versus OrderEventGrp	19
3.3 Data Type of NoFillsIndex.....	20
3.4 Distinction between Triggering and Activating an Order	20
3.5 Extension of ExecRestatementReason vs ExecTypeReason.....	21
3.6 Optionality of ClOrdID when using OrderRequestID.....	21
4 Proposed Message Flow	21
4.1 Immediate Order Execution Upon Entry.....	22
4.2 Immediate Order Execution Upon Modification.....	23
4.3 Non-Resting Orders	24
4.4 Order Request Identifier.....	25
4.5 MassOrder With Additional Events for all Entries	26
4.6 MassOrder With Additional Events for some Entries	27
4.7 MassOrder Without Additional Events	28
4.8 MassOrder Rejected.....	30
4.9 MassOrder With No Acknowledgements.....	31
4.10 MassOrder With Acknowledgement of Each Order.....	32
4.11 MassOrder With Summary Acknowledgement.....	33
5 FIX Message Tables	34
5.1 FIX Message ExecutionReport	34
5.2 FIX Message MassOrder.....	46
5.3 FIX Message MassOrderAck	48
5.4 FIX Message NewOrderSingle	50
5.5 FIX Message NewOrderMultileg.....	51
5.6 FIX Message NewOrderCross	52
5.7 FIX Message OrderCancelReplaceRequest	52
5.8 FIX Message MultilegOrderCancelReplace.....	53
5.9 FIX Message CrossOrderCancelReplaceRequest	54
5.10 FIX Message OrderCancelRequest	55
5.11 FIX Message CrossOrderCancelRequest	55
5.12 FIX Message OrderCancelReject.....	56
6 FIX Component Blocks	57
6.1 Component InstrmtLegExecGrp	57
6.2 Component OrderEntryGrp.....	58
6.3 Component OrderEntryAckGrp	60
7 Category Changes.....	61
Appendix A - Data Dictionary.....	62
Appendix B - Glossary Entries	69
Appendix C - Abbreviations	69
Appendix D - Usage Examples.....	69

Immediate Order Execution Upon Entry or Modification	69
Multiple Order Execution Upon Entry or Modification	70
Non-Resting Orders	71
Mass Order Transactions with and without Additional Events	72
Mass Order Transactions with Different Levels of Acknowledgement.....	74

Table of Figures

Figure 1: Order state transitions for high performance FIX	10
Figure 2: Immediate order execution upon entry.....	22
Figure 3: Immediate order execution upon modification.....	23
Figure 4: Non-resting orders.....	24
Figure 5: Order request identifier	25
Figure 6: MassOrder with additional events for all entries.....	26
Figure 7: MassOrder with additional events for some entries	27
Figure 8: MassOrder without additional events but with rejections	28
Figure 9: MassOrder without additional events and without rejections	29
Figure 10: MassOrder rejected	30
Figure 11: MassOrder with no acknowledgements	31
Figure 12: MassOrder with acknowledgement of each order	32
Figure 13: MassOrder with summary acknowledgement	33

Document History

Revision	Date	Author	Revision Comments
0.1	Aug 22, 2012	Hanno Klein Deutsche Börse Group	Initial version. Includes topics from Volume 7 of the FIX 5.0 SP2 Errata Specification (<i>User Group: Exchanges and Markets</i>).
0.2	Oct 10, 2012	Hanno Klein Deutsche Börse Group Aditya Kapur CME Group	Added "Part 1" to title to allow partitioning of overall effort. Added multiple order executions that have leg executions Added optimized state transitions Added order request identifier Added mass order
0.3	Dec 12, 2012	Hanno Klein Deutsche Börse Group Aditya Kapur CME Group	Replaced mass order with extended version Extended and corrected order state transitions Added issue to clarify distinction between triggering and activation of an order Added issue regarding ExecRestatementReason vs new field ExecTypeReason Added MassOrderRequestID to ExecutionReport
0.4	Jan 21, 2013	Hanno Klein Deutsche Börse Group Aditya Kapur CME Group	Added status and message ID fields to mass order acknowledgement Added message flows and usage examples for mass order messages Added additional fields and the Parties component block to the root level of the mass order messages Added ApplicationSequenceControl component block to the MassOrderAck message Added state transition for orders being cancelled whilst being suspended Completed issue resolutions
0.5	Jan 24, 2013	Hanno Klein Deutsche Börse Group	Created clean version for HPWG review Jan 23 Re-inserted page header fields Applied changes after HPWG review: <ul style="list-style-type: none"> • Added summary of gaps to Chapter 1 • Added message flow for new order request identifier • Added CxlQty to ExecutionReport (only part of MassOrderAck before) • Updated order state transition diagram • Corrected errors (consistency, formatting) and added clarifications and table of figures • Added issue for optional ClOrdID

Revision	Date	Author	Revision Comments
0.6	Feb 11, 2013	Hanno Klein Deutsche Börse Group	Applied changes after GTC review on Jan 31 and Feb 7: <ul style="list-style-type: none"> • Added conditional requirement of OrdType in OrderEntryGrp • Open issue 3.6 (optional ClOrdID) to remain open for public comment
	March 3, 2013	L. Taikitsadaporn	Clean up edits prior to public comment posting. Added additional snippets of message tables for NewOrderSingle, NewOrderMultileg, NewOrderCorss, OrderCancelReplaceRequest, MultilegOrderCancelReplace, CrossOrderCancelReplaceReuest, OrderCancelRequest, CrossOrderCancelRequest, OrderCancelReject to show addition of new field OrderRequestID(#d 2422).
ASBUILT	May 1, 2014	L. Taikitsadaporn	ASBUILT created
	July 7, 2014	Brian Danylkiw	ThrottleInst(1685) is included within the component ThrottleResponse(ID=1069) and therefore not permitted in the top level of the MassOrderAck(35=DK) message, therefore it has been deleted from the top level of the message.

1 Introduction

1.1 Background

The High Performance working group (HPWG) was originally created by the FIX Protocol Limited (FPL) Global Steering Committee in October 2010 but was not able to ~~acquire~~ acquire much momentum until April 2012 when there was an increased commitment from a number of key marketplaces.

1.2 Working group charter

The charter of the working group is to identify opportunities to enrich high performance financial messaging and propose specific enhancements to FIX to address these identified opportunities, including application level, session level (recovery), and encoding.

The goal is to determine the best approach for a fit-for-purpose protocol largely based upon the FIX standard that is “as simple as possible, but no simpler.” It will aim to reuse existing technologies and approaches that have affinity with existing implementations and a high degree of certainty with respect to intellectual property.

Proof of concepts and empirical testing shall be employed where needed. Differing from previous initiatives, the working group will make a request for recommendations for the use of existing technologies.

The FIX High Performance Working Group has identified a number of areas that would benefit from standardisation:

1. Application layer aspects such as message flows, semantic verbosity etc.
2. Session layer aspects such as sequencing, recovery etc.
3. Encoding layer aspects such as wire representation of application and session layer data

1.3 Working group organization

The working group is organized in five subgroups:

- The application layer subgroup
- The session layer subgroup
- The “Simple Binary” encoding subgroup
- The Google Protocol Buffers encoding subgroup and
- The ASN.1 encoding subgroup.

1.4 This document

This document provides guidelines towards the usage of the FIX application layer in high performance environments. As far as the analysis uncovers gaps in the current FIX standard related to such a usage, proposals will be made to close them.

The following extensions are proposed as part of the Part 1 Gap Analysis:

- New field ExecTypeReason(~~TBD2431~~) in ExecutionReport(~~35=8~~) to provide details for ExecType(150)
- New valid values for [the](#) field Triggered(1823) to distinguish trigger types (stop, OCO, OTO, OOU)
- Remove conditional requirement of LastPx(31) and LastQty(32) in the ExecutionReport(~~35=8~~) message if [the](#) components [FillsGrp](#) or [OrderEventGrp](#) are present
- New field ~~NoFillsIndex~~FillRefID(~~TBD2421~~) to reference a fill within [the](#) [FillsGrp](#) [component](#) from a leg execution within [the](#) [InstrmtLegExecGrp](#) [component](#)
- New field OrderRequestID(~~TBD2422~~) for order maintenance and ExecutionReport(~~35=8~~) messages to link requests and responses independent of the identification of an order via ClOrdID(11)
- New messages MassOrder(~~35=tbdDJ~~) and MassOrderAck(~~35=tbdDK~~) with new components OrderEntryGrp and OrderEntryAckGrp and new fields MassOrderRequestID(~~TBD2423~~), MassOrderReportID(~~TBD2424~~), OrderResponseLevel(~~TBD2427~~),

MassOrderRequestStatus([TBD2425](#)), MassOrderRequestResult([TBD2426](#)), OrderEntryAction([TBD2429](#)), OrderEntryID([TBD2430](#)) and TotNoOrderEntries([TBD2432](#)) to support the mass entry of order transactions requiring a limited number of order attributes

2 Business Workflow

This chapter starts off with the high level requirements applicable to the application layer in high performance environments. This is followed by individual sections covering a specific area of the FIX standard. Each section contains a business requirement as well as the recommended best practice and any gaps that have been identified.

2.1 General Requirements

This section defines high-level requirements that are not specific to a single use case.

2.1.1 Minimal Message Volume

Each business process is expressed by a series of messages being exchanged between counterparties. The number of messages has a significant impact on the overall performance of the business process. The message level verbosity should be as low as possible without losing information.

2.1.2 Avoidance of Field Echoing

Messages can be requests to conduct a specific transaction which are typically countered by a message representing the response to the request. The length of a message has a significant impact on the time it takes to process it. Some of this is driven by the encoding of the message which is not the subject of this document. A crucial part is the number of fields that are returned in the response to a request. The submitter of a request may only need to receive a very few of the fields that were part of his request. The field level verbosity of responses should be as low as possible without loss of information on the side of the request submitter. The provision of technically redundant fields should be optional.

2.2 Order State Transitions

REQUIREMENTS:

Each FIX ExecutionReport([35=8](#)) message contains two fields which are used to communicate both the current state of the order as understood by its submitter (OrdStatus([39](#))) and the purpose of the message (ExecType([150](#))). A third field (ExecTypeReason([tbd2431](#))) provides additional information for a subset of use cases, especially for unsolicited restatements or cancellations of orders. In FIX 4.x all order state transitions are explicit and require a separate message for each event or state change, e.g. a new order always needs to be confirmed-acknowledged first with OrdStatus = 0 (New) and ExecType = 0 (New), even if it is immediately executed and the ExecutionReport([35=8](#)) message represents an intermediate status that was superseded by another status within the same matching engine transaction at the execution venue.

A fundamental requirement is to reduce the number of messages from both sides representing a single business transaction. This requires the definition of values for OrdStatus([39](#)) and ExecType([150](#)) in the context of a given use case which involves one or more individual events.

As of FIX 5.0 SP1 it is possible to reduce the number of ExecutionReport([35=8](#)) messages for some use cases. This is described in Volume 7 - FIX Usage Notes of the FIX 5.0 SP2 Specification (see *Order State Change Matrices for Exchanges*, starting on page 191). Part of the documentation is the following order state change diagram which includes dotted lines representing the optimizations that were introduced with FIX 5.0 SP1.

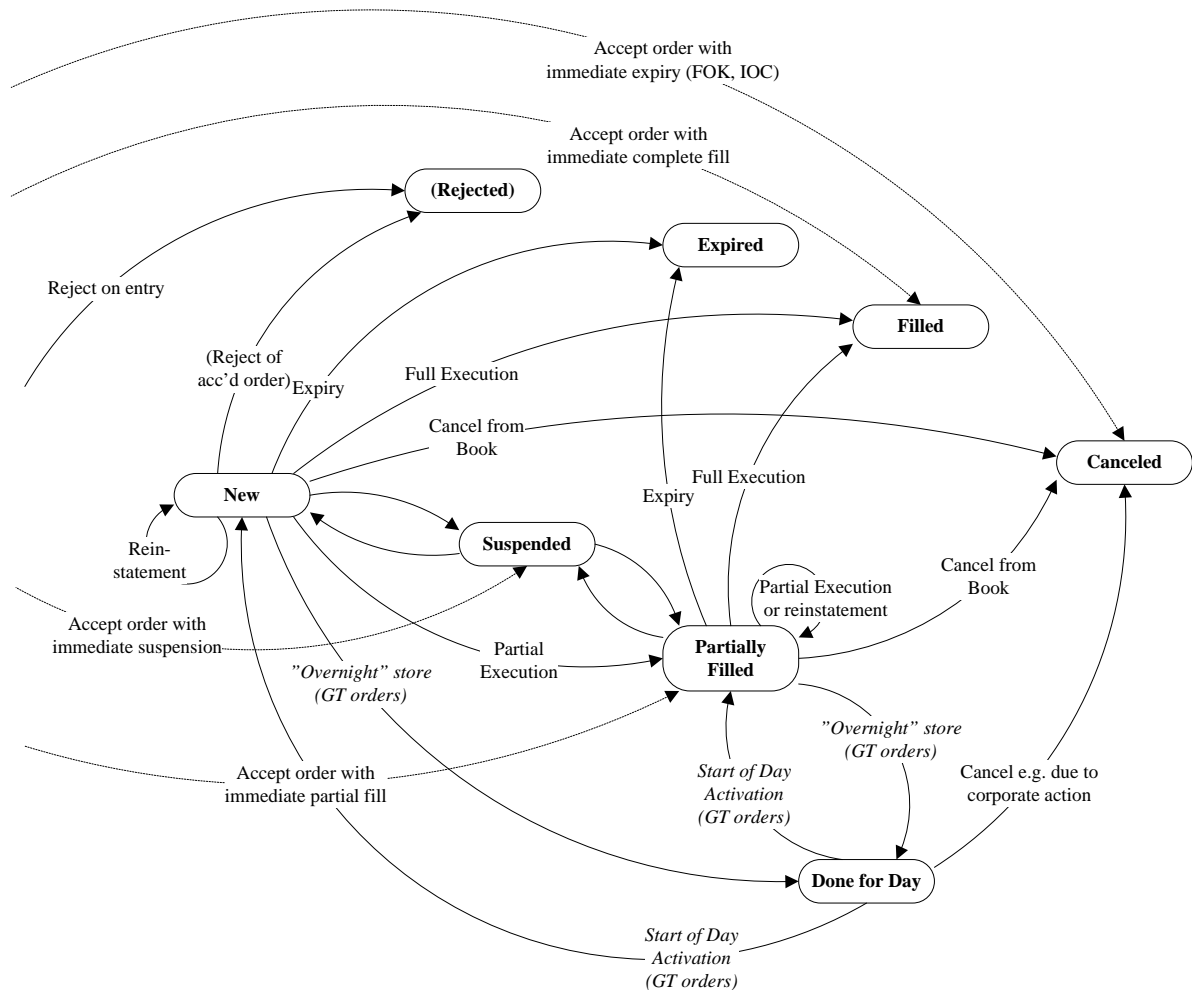


Figure 1: Order state transitions for high performance FIX

However, a comprehensive guideline for all basic use cases is required. The list of these use cases is as follows. Every use case has been assigned a basic category to capture the overall type of business transaction occurring. For example, an order that is being added and immediately executed will have a different order status than an order that simply comes to rest on the book. In both cases the business transaction was the entry of an order.

No.	Category	Use Case
1	Entry	Resting order added without execution
2	Entry	Resting order added with partial execution
3	Entry	Order added with full execution
4	Entry	Non-resting order added without execution
5	Entry	Non-resting order added with partial execution
6	Entry	Trigger order added and triggered without execution
7	Entry	Trigger order added and triggered with partial execution
8	Entry	Trigger order added and triggered with full execution
9	Entry	Order suspended upon entry
10	Modification	Unfilled order modified (remaining resting) without execution
11	Modification	Unfilled order modified (remaining resting) with partial execution
12	Modification	Partially filled order modified without execution
13	Modification	Order quantity modified at or below executed quantity
14	Modification	Order modified (remaining resting) with partial execution
15	Modification	Order modified with full execution
16	Modification	Order modified to non-resting without execution

<u>No.</u>	<u>Category</u>	<u>Use Case</u>
17	Modification	Order modified to non-resting with partial execution
18	Modification	Trigger order modified and triggered without execution
19	Modification	Trigger order modified and triggered with partial execution
20	Modification	Trigger order modified and triggered with full execution
21	Modification	Order modified during suspension (remaining suspended)
22	Modification	Suspended order modified to activate with partial execution
23	Modification	Suspended order modified to activate with full execution
24	Modification	Suspended unfilled order modified to activate without execution
25	Modification	Suspended partially filled order modified to activate without execution
26	Cancellation	Order deleted/ cancelled
27	Cancellation	Order deletion/ cancellation accepted but deferred
28	Cancellation	Actual order deletion/ cancellation after being initially deferred
29	Cancellation	Order deleted/ cancelled during suspension
30	Expiry	Order expired
31	Execution	Partial execution of order resting on book
32	Execution	Full execution of order resting on book
33	Trigger	Resting trigger order triggered without execution
34	Trigger	Resting trigger order triggered with partial execution
35	Trigger	Resting trigger order triggered with full execution
36	Suspension	Unfilled order becoming activated
37	Suspension	Partially filled order becoming activated
38	Suspension	Order becoming suspended
39	Restatement	Restatement of unfilled order
40	Restatement	Restatement of partially filled order
41	Restatement	Restatement of suspended order (unfilled or partially filled)
42	Restatement	Restatement of a cancelled order (unsolicited cancellation)

Some of these use cases will then result in the same values of OrdStatus([39](#)) and ExecType([150](#)) and hence require ExecTypeReason([TBD2431](#)) for further distinction.

GUIDELINES:

- Use a single ExecutionReport([35=8](#)) message (~~MsgType=8~~) to respond to the following use cases ~~and set with the fields~~ OrdStatus([39](#)), ExecType([150](#)) and ExecTypeReason([TBD2431](#)) as set forth below.
- Remove trigger attributes from triggered orders and set trigger category accordingly with field Triggered(1823) which is initially set to 0=Not triggered or absent.
- Non-resting orders are orders with attributes such as IOC or FOK or any other attribute that may lead to a rejection or cancellation upon entry (or a modification setting a non-resting attribute) .
- Trigger orders are orders that are inactive (hidden) until their trigger condition is met, after which they cannot go back to their hidden state.
- Suspended orders are orders that are only valid during specific periods, e.g. trading (sub_)sessions and that can switch back and forth between being active and inactive (suspended).

<u>No.</u>	<u>Category</u>	<u>Use Case</u>	<u>OrdStatus</u>	<u>ExecType</u>	<u>ExecTypeReason</u>
1	Entry	Resting order added without execution	New	New	Order added upon request
2	Entry	Resting order added with partial execution	Partially Filled	Trade	Order added upon request
3	Entry	Order added with full execution	Filled	Trade	Order added upon request

No.	Category	Use Case	OrdStatus	ExecType	ExecTypeReason
4	Entry	Non-resting order added without execution	Cancelled	Cancelled	Non-resting order added upon request
5	Entry	Non-resting order added with partial execution	Cancelled	Trade	Non-resting order added upon request
6	Entry	Trigger order added and triggered without execution	New	Triggered	Order added upon request
7	Entry	Trigger order added and triggered with partial execution	Partially Filled	Trade	Order added upon request
8	Entry	Trigger order added and triggered with full execution	Filled	Trade	Order added upon request
9	Entry	Order suspended upon entry	Suspended	New	Order added upon request
10	Modification	Unfilled order modified (remaining resting) without execution	New	Replaced	Order replaced upon request
11	Modification	Unfilled order modified (remaining resting) with partial execution	Partially Filled	Trade	Order replaced upon request
12	Modification	Partially filled order modified without execution	Partially Filled	Replaced	Order replaced upon request
13	Modification	Order quantity modified at or below executed quantity	Filled	Replaced	Order replaced upon request
14	Modification	Order modified (remaining resting) with partial execution	Partially Filled	Trade	Order replaced upon request
15	Modification	Order modified with full execution	Filled	Trade	Order replaced upon request
16	Modification	Order modified to non-resting without execution	Cancelled	Cancelled	Order replaced with non-resting order upon request
17	Modification	Order modified to non-resting with partial execution	Cancelled	Trade	Order replaced with non-resting order upon request
18	Modification	Trigger order modified and triggered without execution	New	Triggered	Trigger order replaced upon request
19	Modification	Trigger order modified and triggered with partial execution	Partially Filled	Trade	Trigger order replaced upon request
20	Modification	Trigger order modified and triggered with full execution	Filled	Trade	Trigger order replaced upon request
21	Modification	Order modified during suspension (remaining suspended)	Suspended	Replaced	Suspended order replaced upon request
22	Modification	Suspended order modified to activate with partial execution	Partially Filled	Trade	Suspended order replaced upon request
23	Modification	Suspended order modified to activate with full execution	Filled	Trade	Suspended order replaced upon request
24	Modification	Suspended unfilled order modified to activate without execution	New	Activated	Suspended order replaced upon request

<u>No.</u>	<u>Category</u>	<u>Use Case</u>	<u>OrdStatus</u>	<u>ExecType</u>	<u>ExecTypeReason</u>
25	Modification	Suspended partially filled order modified to activate without execution	Partially Filled	Activated	Suspended order replaced upon request
26	Cancellation	Order deleted/ cancelled	Cancelled	Cancelled	Order cancelled upon request
27	Cancellation	Order deletion/ cancellation accepted but deferred	Pending	Pending	Order cancellation pending
28	Cancellation	Actual order deletion/ cancellation after being initially deferred	Cancelled	Cancelled	Pending cancellation executed
29	Cancellation	Order deleted/ cancelled during suspension	Suspended	Cancelled	Suspended order cancelled upon request
30	Expiry	Order expired	Expired	Expired	Order expired
31	Execution	Partial execution of order resting on book	Partially Filled	Trade	Order book execution
32	Execution	Full execution of order resting on book	Filled	Trade	Order book execution
33	Trigger	Resting trigger order triggered without execution	New	Triggered	Resting order triggered
34	Trigger	Resting trigger order triggered with partial execution	Partially Filled	Trade	Resting order triggered
35	Trigger	Resting trigger order triggered with full execution	Filled	Trade	Resting order triggered
36	Suspension	Unfilled order becoming activated	New	Activated	Suspended order activated
37	Suspension	Partially filled order becoming activated	Partially Filled	Activated	Suspended order activated
38	Suspension	Order becoming suspended	Suspended	Suspended	Active order suspended
39	Restatement	Restatement of unfilled order	New	Restated	Order book restatement
40	Restatement	Restatement of partially filled order	Partially Filled	Restated	Order book restatement
41	Restatement	Restatement of suspended order (unfilled or partially filled)	Suspended	Restated	Order book restatement
42	Restatement	Restatement of cancelled order (unsolicited cancellation)	Cancelled	Cancelled	Unsolicited cancellation

GAPS:

Add a new field ExecTypeReason([TBD2431](#)) ~~as with the following~~ [values to identify the reason or initiating event which resulted in the ExecutionReport\(35=8\) being sent.](#)

- 1 = Order added upon request
- 2 = Order replaced upon request
- 3 = Order cancelled upon request
- 4 = Unsolicited cancellation
- 5 = Non-resting order added upon request

- 6 = Order replaced with non-resting order upon request
- 7 = Trigger order replaced upon request
- 8 = Suspended order replaced upon request
- 9 = Suspended order cancelled upon request
- 10 = Order cancellation pending
- 11 = Pending cancellation executed
- 12 = Resting order triggered
- 13 = Suspended order activated
- 14 = Active order suspended
- 15 = Order expired

Add new values to Triggered(1823) as follows:

- ~~FBD-2~~ = Stop order triggered
- ~~FBD-3~~ = One Cancels the Other (OCO) order triggered
- ~~FBD-4~~ = One Triggers the Other (OTO) order triggered
- ~~FBD-5~~ = One Updates the Other (OUO) order triggered

2.3 Immediate Order Execution Upon Entry or Modification

REQUIREMENTS:

Orders may be immediately executable when they are entered into the book or modified as a resting order. There is no need to separately confirm the entry, and (partial) execution of the order.

GUIDELINES:

- Use a single ExecutionReport(35=8) message (~~MsgType=8~~) to respond to the entry or modification of an order that is immediately executed.
- Set field ExecType-(150) = F(Trade) ~~to F=Trade~~ to indicate that the order was (partially) executed.
- Set field OrdStatus-(39) = 1(Partially filled) ~~to 1=Partially Filled~~ to indicate that the order still has a remaining quantity.
- Set field OrdStatus-(39) = 2(Filled) ~~to 2=Filled~~ to indicate that the order has been fully executed.
- Set field LeavesQty-(151) to the remaining quantity which can be zero.
- Set field CumQty-(14) to the total quantity executed across all partial fills of the match event.
- Use the component FillsGrp to convey one or more partial fills.
- Omit fields LastPx(31) and LastQty-(32)

GAPS:

- Modify field usage description of LastPx(31) and LastQty-(32) in ExecutionReport(35=8) which defines the fields as conditionally required for ExecType-(150) = F(Trade) ~~F=Trade~~.

2.4 Multiple Order Executions Upon Entry or Modification

REQUIREMENTS:

Orders may be executed more than once at different price levels when they are entered into the book or modified as a resting order. The matching engine conducts this event as an atomic transaction and it is sufficient to return a single message to convey the results of the entry or modification.

GUIDELINES:

- Use a single ExecutionReport(35=8) message (~~MsgType=8~~) to respond to the entry or modification of an order that is executed at different price levels within a single match event.
- Set field ExecType-(150) = ~~F(Trade)~~ to ~~F=Trade~~ to indicate that the order was (partially) executed.
- Set field OrdStatus-(39) = ~~1(Partially filled)~~ to ~~1=Partially Filled~~ to indicate that the order still has a remaining quantity.
- Set field OrdStatus-(39) = ~~2(Filled)~~ to ~~2=Filled~~ to indicate that the order has been fully executed.
- Set field LeavesQty-(151) to the remaining quantity which can be zero.
- Set field CumQty-(14) to the total quantity executed across all partial fills of the match event.
- Use [the](#) component FillsGrp to convey all partial fills.
- Omit fields LastPx(31) and LastQty(32)

GAPS:

- Modify field usage description of LastPx(31) and LastQty (32) in ExecutionReport(35=8) which defines the fields as conditionally required for ExecType (150) = ~~F(Trade)~~~~F=Trade~~.

2.5 Multiple Order Executions with Leg Executions

REQUIREMENTS:

Orders may be for a simple or for a complex instrument with 2 or more legs (aka multileg orders or strategies). Both can be ~~be~~ executed more than once at different price levels within a single match event. All executions need to be sent as part of a single ExecutionReport(35=8) message to the owner of the order.

GUIDELINES:

- Use [the](#) component FillsGrp to convey partial fills on the level of the order
- In case of a complex order, use [the](#) component InstrmtLegExecGrp to convey leg level executions for all partial fills in the FillsGrp component.
- Group the leg level executions in [the](#) InstrmtLegExecGrp [component](#) by partial fill of the complex order and use the same order for these groups as used for the instances of [the](#) FillsGrp [component](#).

GAPS:

- Add a new field ~~NoFillsIndex-FillRefID~~(~~TBD~~2421) to the component InstrmtLegExecGrp to reference an instance of the repeating group FillsGrp.

2.6 Non-Resting Orders

REQUIREMENTS:

Orders may carry attributes that prevent them from resting on the book, ~~For~~ example IOC (Immediate or Cancel) or FOK (Fill or Kill) orders may be (partially) executed immediately but the remainder of the order will

be cancelled and not written to the order book. The order will be in a terminal state. There is no need to separately confirm the entry, (partial) execution and cancellation of the order.

GUIDELINES:

- Use a single ExecutionReport(35=8) message (~~MsgType=8~~) to respond to the entry of non-resting orders.
- Set field ExecType-(150) = ~~F(Trade)~~ to ~~F=Trade~~ to indicate that the order was (partially) executed.
- Set field ExecType-(150) = ~~4(Cancelled)~~ to ~~4=Cancelled~~ to indicate that the order was not executed.
- Set field OrdStatus-(39) = ~~4(Cancelled)~~ to ~~4=Cancelled~~ to indicate that the order is in a terminal state
- Set field LeavesQty-(151) to zero to indicate that the order has no executable quantity left
- Set field CumQty-(14) to the total quantity executed upon order entry
- Use the component FillsGrp to convey partial fills [details of the match event](#) (if any occurred)
- Omit the fields LastPx(31) and LastQty (32).

GAPS:

- Modify field usage description of LastPx(31) and LastQty-(32) in ExecutionReport(35=8) which defines the fields as conditionally required for ExecType-(150) = ~~F(Trade)~~-~~F=Trade~~.

2.7 Order Request Identifier

REQUIREMENTS:

Requests related to individual orders do not have a dedicated request identifier with the exception of the OrderStatusRequest(35=H) message which provides OrderStatusReqID(790). Other FIX messages typically have a field which only serves as a (request) message identifier. However, in messages such as NewOrderSingle(35=D), OrderCancelReplaceRequest(35=G), or OrderCancelRequest(35=F) the entity identifier for the order ClOrdID(11) also serves as message identifier. A new request message requires the user to issue a new ClOrdID(11) value whilst passing the ~~old~~-previous value in OrigClOrdID(41) to allow identification of the order entity to be cancelled or replaced. Hence, ClOrdID(11) serves two purposes with different characteristics. A request identifier is of technical nature, mainly to tie responses back to requests. An entity identifier is associated with the order as a business entity or object, and does not have to change over the lifetime of an order. The requirement is to provide a means to separate the two types of identifiers for individual orders to increase performance.

Currently, FIX requires the user to issue a new ClOrdID(11) value every time the user submits a request to modify or delete/cancel an order. The associated concept is called *message chaining*. Note that ClOrdID(11) does not change in case of partial fills or any other unsolicited ExecutionReport(35=8). This includes modifications and deletions/cancellations conducted over a non-FIX session (e.g. GUI, on-behalf action by Market Operations) as this would require a synchronization of ClOrdID(11) values which may not be easy for the user, especially when non-FIX sessions are only used in case of an emergency.

Every new ClOrdID(11) value needs to be validated against the currently active orders in the book to ensure uniqueness. An order identifier from the user is of little value in cases where the order does not rest on the book (e.g. IOC) or where the order is simply deleted/cancelled. A request identifier is still needed in such cases but would not require a lookup in the order book. It only needs to be echoed in the response and can be ignored by the core matching engine.

Please also refer to the open issue in section 3.6 *Optionality of ClOrdID when using OrderRequestID*.

GUIDELINES:

- Use OrderRequestID(~~TBD~~2422) to identify a request to enter, modify or delete/cancel an order and echo the value on the ExecutionReport(35=8) representing the response.

- Omit OrderRequestID(~~TBD2422~~) on ExecutionReport(35=8) messages conveying unsolicited events, e.g. book order executions, order restatements at start of day or for corporate actions.
- Use ClOrdID(11) for the entry of new orders only if ~~you~~ the user intend to use message chaining for order modifications or ~~if you need to~~ when ~~modify~~ ing or ~~deleting/cancelling~~ the order prior to having received a response to the submitted order ~~entry~~.
- Allow the scope of uniqueness for ClOrdID(11) to be limited to a given session to enable order submitters to use multiple independent sessions to ~~accessing~~ the same orders, i.e. every order accepted by the recipient must be able to carry not only a ClOrdID(11) value but also a session identifier.
- Use ClOrdID(11) and OrigClOrdID(41) for the modification of existing orders if they were entered with ClOrdID(11).
- Use only OrigClOrdID(41) and omit ClOrdID(11) for the deletion/cancellation of orders that were entered with ClOrdID(11).
- Omit usage of ClOrdID(11) in case of non-resting orders or if ~~you access~~ the order is being accessed from a session that is different from the session that entered the order.
- The same guidelines apply to cross orders where CrossID(548) is semantically identical to ClOrdID(11) and OrigCrossID(551) is semantically identical to OrigClOrdID(41).

GAPS:

- Add the new field OrderRequestID(~~TBD2422~~) to ~~messages~~ NewOrderSingle(35=D), NewOrderMultileg(35=AB), NewOrderCross(35=s), OrderCancelReplaceRequest(35=G), MultilegOrderCancelReplace(35=AC), CrossOrderCancelReplaceRequest(35=t), OrderCancelRequest(35=F), CrossOrderCancelRequest(35=u), OrderCancelReject(35=9), ExecutionReport(35=8)

2.8 Mass Order

The only mechanism to submit multiple orders simultaneously within the context of the existing FIX order handling semantics was provided by ~~so-called Program Trading~~ the NewOrderList(35=E) used to support Program or Basket Trading. Only the entry of multiple orders is possible with NewOrderList(35=E) message. Orders of such a list can only be modified individually and remain members of that list. The entire list can be cancelled but not an arbitrary set of orders across multiple lists.

A new message type is needed which will facilitate mass order entry similar to mass quotes, and towards this end we are ~~The~~ proposing ~~is~~ the introduction of a ~~mass order~~ message which will allow ~~end~~ users to add, modify and delete/cancel an arbitrary collection of orders with using a single message. This message is intended to be used for orders which are immediately active and only have a very short life cycle during the course of the current business day.

REQUIREMENTS:

- Automated users, such as an algorithmic trading application working orders through for a single product, may submit multiple orders linked with a unique request identifier for the set of orders in a single mass order message.
- Orders submitted via mass order messages can be modified or cancelled at the individual order level or via mass order messages. Any order submitted individually can subsequently be modified or cancelled with a mass order message together with ~~arbitrary~~ other orders.
- A mass order message can contain any combination of order additions, modifications or deletions/cancellations.
- Orders being modified or cancelled with mass order messages can be combined arbitrarily without being tied to the initial set of orders they were a part of.
- Orders submitted via the mass order messages need at least the subset of the attributes available to orders as far as these attributes are used for high performance trading, e.g. IOC.

- Mass order messages support trading- as well as clearing-related attributes that apply to all orders in the given mass order message. Some attributes such as side (buy/sell) need to be available per order.
- Mass order messages support equity cash and carry orders which could be hundreds of orders submitted simultaneously.
- The workflow for mass order messages supports laddering. Laddering refers to multiple orders per side for the same instrument in a single mass order message.
- Mass order messages are acknowledged by means of a single message which may provide detailed or summary information on the acceptance of the individual orders. This may contain information about the orders' status after being processed including immediate execution.
- Trade details of immediate executions of orders submitted with a mass order message are provided in the same way as executions of resting orders, in a separate message.
- There is no fundamental difference in the order life cycle of orders submitted individually or as part of a mass order message.

GUIDELINES:

- Mass order messages should only be used to add, modify or delete/[cancel](#) multiple orders within one message whenever the recipient is also able to process these orders atomically without having to split up the mass order message internally and collect various responses back together for a single acknowledgement message. This will typically be a set of related instruments that are handled by a single matching engine.
- The number of orders allowed in a single mass order message should be subject to bilateral agreement between counterparties in order to avoid large message sizes and the need for [message](#) fragmentation.
- A single order should only be contained once in a single mass order message, e.g. it should not be entered and modified or modified more than once with a single message.
- The message identifier should be different for every mass order message and attached to the order by the recipient to be able to provide it with an execution of one of its orders. This allows [the order submitter](#) to determine the mass order message that led to an immediate execution.
- The root level fields of the mass order message should not be echoed back in the acknowledgement message unless it is provided as a drop copy to a user other than the submitter of the mass order message.
- The entry level fields of the acknowledgement message to identify the entry can be reduced to a single identifier (OrderEntryID([42430](#))) if established as part of the mass order message. It avoids the need to echo back ClOrdID([11](#)), OrigClOrdID([41](#)), OrderID([37](#)) or fields of the instrument component.
- Summary acknowledgement messages (or if no acknowledgement messages are sent) should only be used if they can be followed by individual ExecutionReport([35=8](#)) messages to convey the current status of the orders inside the mass order message.
- Minimum acknowledgement messages should contain status information for rejected transactions or for transactions that led to additional events beyond the action requested. This could, for example, be the immediate execution of an order that was entered or modified. An alternative is to limit the minimum to rejected transactions and send additional ExecutionReport([35=8](#)) messages for transactions that led to additional events.
- Minimum acknowledgement messages may contain status information for transactions adding or deleting orders even if there were no additional events or rejections. The use cases are new orders for which the receiver assigns an order identifier and the cancellation of orders where the actually cancelled quantity is to be conveyed. The alternative is again to send additional ExecutionReport([35=8](#)) messages to convey this information.
- The rules for mixing mass order messages with individual order messages should be carefully designed for a given implementation. Optional order attributes that are only available in the individual order messages should not be used for orders that are permitted to use mass order messages. FIX requires

[recipients](#) to remove omitted optional order attributes in a modification request instead of ignoring them. On the other hand, a root level attribute of a mass order message may be ~~overridden~~ overridden by using an individual order message to modify it.

- Order submitters should be prepared to update their order status either through mass order acknowledgement messages or through individual ExecutionReport(35=8) messages.
- CxlQty(84) can be used in the acknowledgement message (or in the ExecutionReport(35=8)) to explicitly convey the quantity that was remaining at the time of order deletion/cancellation. This may be less than at the time of the submission of the cancellation request due to partial executions occurring thereafter and prior to the actual deletion/cancellation.

GAPS:

- Add new message type MassOrder(35=~~4~~4DJ) to **bulk** add, modify and delete/cancel orders
- Add new message type MassOrderAck(35=~~4~~4DK) to acknowledge a MassOrder(35=~~4~~4DJ)
- Add new field MassOrderRequestID(~~4~~42423) to identify the mass order message
- Add new field MassOrderReportID(~~4~~42424) to identify the mass order acknowledgment message
- Add new field OrderResponseLevel(~~4~~42427) to convey desired verbosity of acknowledgement
- Add new field MassOrderRequestStatus(~~4~~42425) to return the overall result of the message
- Add new field MassOrderRequestResult(~~4~~42426) to return further status detail
- Add new component <OrderEntryGrp> to submit a group of order transactions
- Add new component <OrderEntryAckGrp> to acknowledge a group of order transactions
- Add new field OrderEntryAction(~~4~~42429) to identify the order transaction type
- Add new field OrderEntryID(~~4~~42430) to identify the individual order in the context of the mass order
- Add new field TotNoOrderEntries(~~4~~42432) to support fragmentation
- Add CxlQty(84) to ExecutionReport(35=8) for mass order requests with summary or no acknowledgement

3 Issues and Discussion Points

3.1 Usage of LastPx/LastQty and FillsGrp

Question: Should LastPx(31)/LastQty(32) be used in addition to FillsGrp to convey the last entry of FillsGrp?

Discussion: LastPx(31) and LastQty(32) contain the price and quantity of an execution at a single price level whereas the FillsGrp can contain one or more executions at different price levels. Repeating the last entry of FillsGrp in LastPx(31)/LastQty(32) is technically redundant and increases the message size. Omitting the last partial fill from FillsGrp and providing it in LastPx(31)/LastQty(32) creates complexity ~~for as~~ the recipient ~~as he~~ has to combine the two sources to get all partial fills. Additionally, the FillsGrp will be absent if there is only one execution.

A conversion between a high performance FIX interface and a traditional FIX 4.x interface has to break up the FillsGrp into individual ExecutionReport(35=8) messages, each of which will contain one instance of the FillsGrp in LastPx(31) and LastQty(32).

Answer: No, omit LastPx(31)/LastQty(32) and provide all partial fills in FillsGrp.

3.2 Usage of FillsGrp versus OrderEventGrp

Question: Should FillsGrp or OrderEventGrp [components](#) be used to convey partial fills for high performance?

Discussion: Both repeating groups can convey partial fills whereby additional events can be conveyed by OrderEventGrp. The intention of OrderEventGrp is to allow a kind of audit trail within a single message, i.e. avoiding the need for one message per event but without losing the ability to explicitly report the events. The component also has fields to convey the reason for the event type and a free text field with further detail. It is therefore more verbose than FillsGrp. Note that OrderEventGrp can be used in a way that is identical to FillsGrp and would result in the same wire format in a simple binary encoding.

FillsGrp is limited to the execution events of an order on different price levels which cannot be conveyed implicitly. Its usage better reflects the desire of high performance environments to implicitly convey order event information wherever possible, e.g. through the use of optimized state transitions (see Chapter 2.2 *Order State Transitions*).

Answer: FillsGrp should be used in conjunction with the new field ExecTypeReason([TBD2431](#)) which provides additional information on the chain of events that includes partial fills.

3.3 Data Type of NoFillsIndex

Question: What data type should the new field NoFillsIndex have, which references a partial fill in FillsGrp from within InstrmtLegExecGrp?

Discussion: NoFillsIndex will only have values from NoFills(1362) which has data type NumInGroup. However, NumInGroup fields do not have FIXML abbreviations as they are not needed there due to the XML syntax. The cardinality of a repeating group is not explicitly provided in FIXML. Horizontal referencing is possible within XML and has been introduced to FIXML with EP161 where new data types XID and XIDREF have been introduced. The objective is somewhat different because the horizontal reference is used to avoid duplication of information, e.g. settlement terms that are identical only need to be specified once in a given message.

The intended usage here only aims to associate an instance of one repeating group with an instance of another whereby the instances themselves are semantically different, i.e. a partial fill and a leg execution.

Answer: Use NumInGroup to be semantically correct and provide FIXML abbreviation for this field to allow FIXML users to implicitly associate leg executions with partial fills.

Post GTC presentation resolution: After the presentation to the GTC there was further discussion around this proposed field, particularly the naming and the data type. It was decided ~~the that~~ the proposed NoFillsIndex field be renamed to FillRefID([tbd2421](#)) with a data type of String. This would allow for either an explicit reference or implicit reference to the fill instance in the FillsGrp. An explicit reference means the FillRefID([tbd2421](#)) value would carry the FillExecID(1363) value from the instance in the FillsGrp. An implicit reference means the FillRefID([tbd2421](#)) value would be the implied position of the fill within the FillsGrp.

3.4 Distinction between Triggering and Activating an Order

Question: Should ExecType(150) have two distinct values instead of the current single value L=(Triggered or Activated by System)?

Discussion: Triggered orders slightly differ from activated orders in that the triggering can only occur once whereas activation and inactivation can occur multiple times. However, the result of ExecType(150)=L is the same for both, i.e. the order is now able to match against other orders in the book. Splitting the existing valid value into two values would break existing applications, depending on which of the two flavors would receive a new value. The existing field OrderEventType-(1796) only has a value 10=Triggered that is associated with ExecType(150)=L.

Answer: Do not create a new ExecType(150) value and use the new field ExecTypeReason-([TBD2431](#)) instead to explicitly convey whether an order was triggered or activated. Add a new value to OrderEventType-(1796) to provide a distinction.

3.5 Extension of ExecRestatementReason vs ExecTypeReason

Question: Should ExecRestatementReason(378) be extended to convey additional information for all values of ExecType(150) or should this be a new field ExecTypeReason(~~TBD~~2431)?

Discussion: The semantics of the existing field ExecRestatementReason(378) are tied to unsolicited events in a single session environment, e.g. a broker giving verbal instructions to Market Operations. The intended semantics for high performance go beyond the scope of restatements and also include solicited events. These should not be mixed in the existing field.

Answer: Introduce new field ExecTypeReason(~~TBD~~2431).

3.6 Optionality of ClOrdID when using OrderRequestID

Question (HPWG call Jan 25): Should ClOrdID(11) and OrigClOrdID(41) be made optional on order maintenance messages when the new field OrderRequestID(~~TBD~~2422) is being used?

Discussion: ClOrdID(11) serves as message and entity identifier. In order to serve as a message identifier, ClOrdID(11) has to change with every request, including deletions/cancellations. This concept is called message chaining in FIX. The change of ClOrdID(11) requires to always send the ~~old~~ previous (OrigClOrdID(41)) and the new (ClOrdID(11)) identifier which increases the message size. Furthermore, there are use cases such as IOC ~~and~~ FOK orders which do not rest on the book, i.e. the order entry only needs a message identifier to allow mapping of the response but it does not need a client issued entity identifier if OrderRequestID(~~TBD~~2422) is present.

The mandatory nature of ClOrdID(11) and OrigClOrdID(41) also prevents the modification of orders across multiple interfaces unless these all share the same range of ClOrdID(11) values which is likely to limit the performance of these sessions. A workaround for this problem is the usage of both fields having the same value, i.e. providing but not changing the identifier when accessing the order from an interface different from the one that originally submitted the order.

The downside of making client issued order identifiers optional is that message chaining represents a fundamental FIX concept, it represents a significant change to current FIX applications. It may be advisable to only allow the omission in the context of high performance interfaces with a binary encoding where it should be a general option to bilaterally agree on a minimum set of fields in requests and responses.

Answer: GTC review on Feb 7 decided to solicit feedback during public review comment period.

4 Proposed Message Flow

This chapter provides a graphical representation of the recommended message flows supporting high performance from the viewpoint of an exchange. Messages are shown with a few key fields that help to understand the overall flow. The focus, however, is on the message level and not on the field level. More detailed examples from the viewpoint of a single submitter of requests can be found in *Appendix D – Usage Examples*.

4.1 Immediate Order Execution Upon Entry

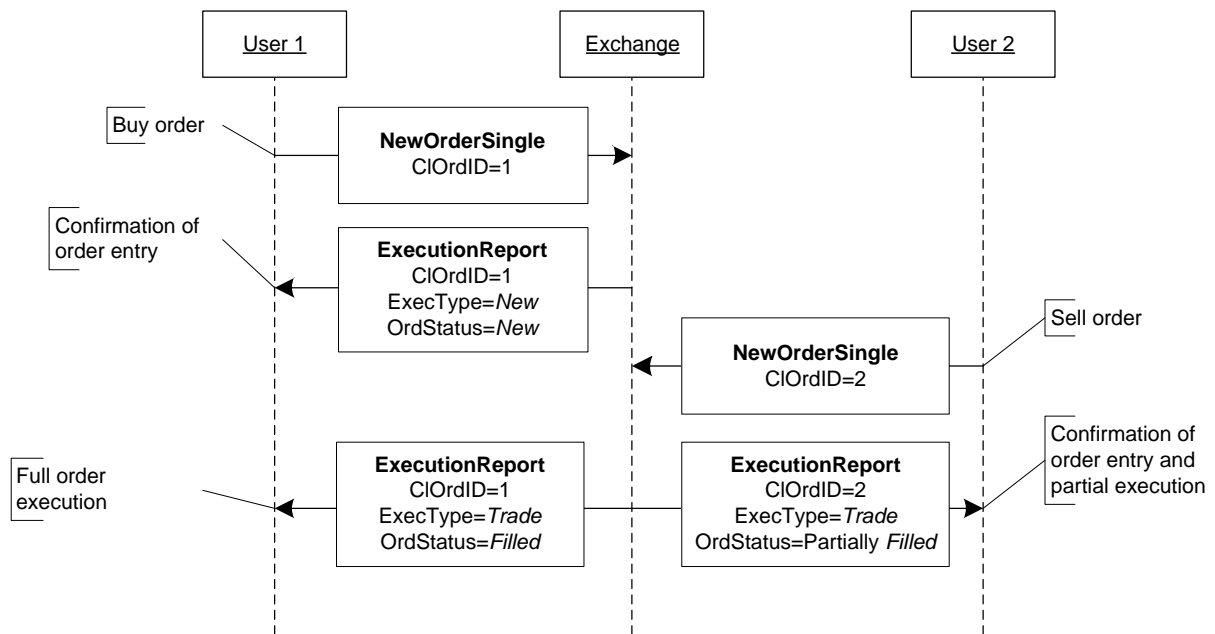


Figure 2: Immediate order execution upon entry

4.2 Immediate Order Execution Upon Modification

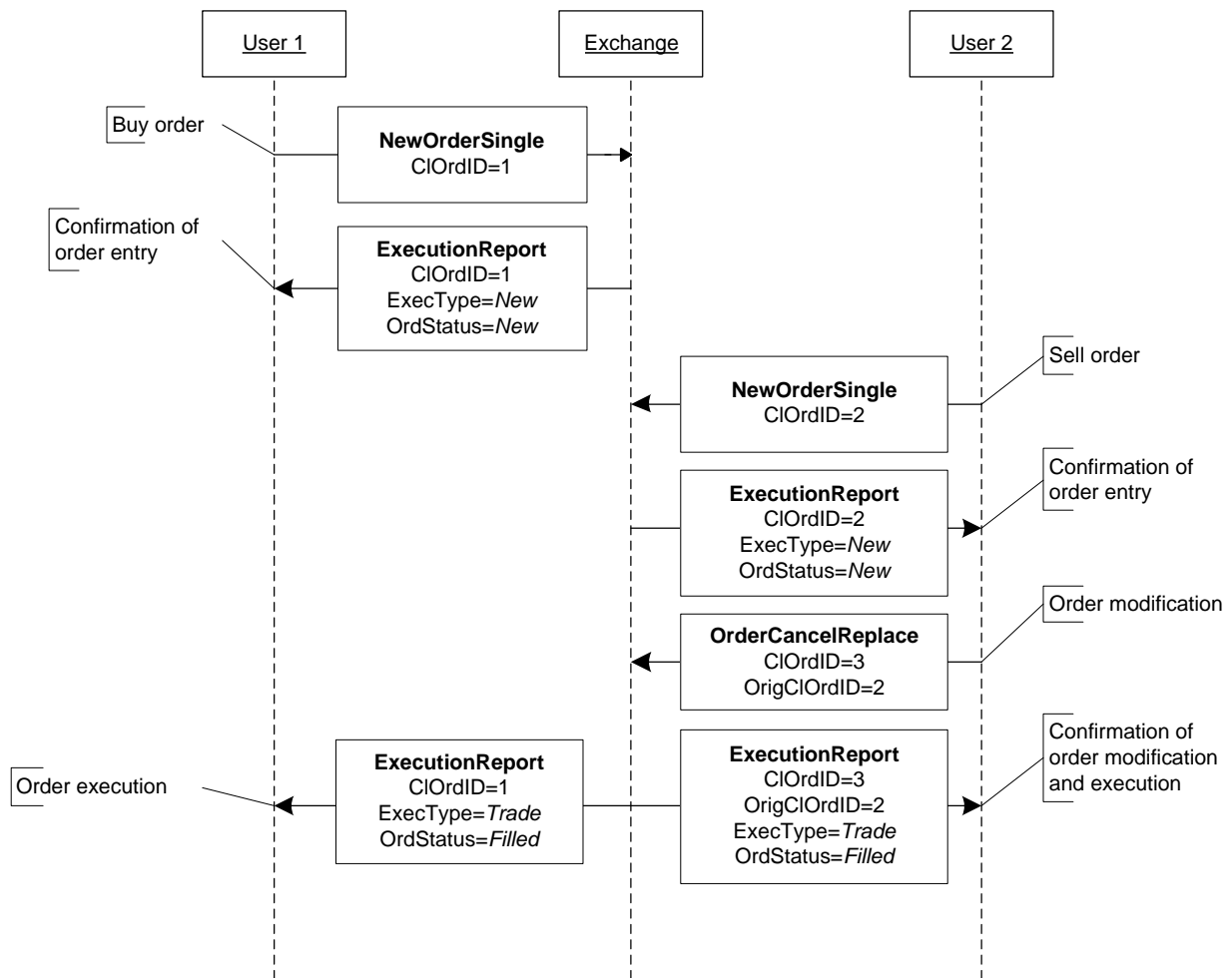


Figure 3: Immediate order execution upon modification

4.3 Non-Resting Orders

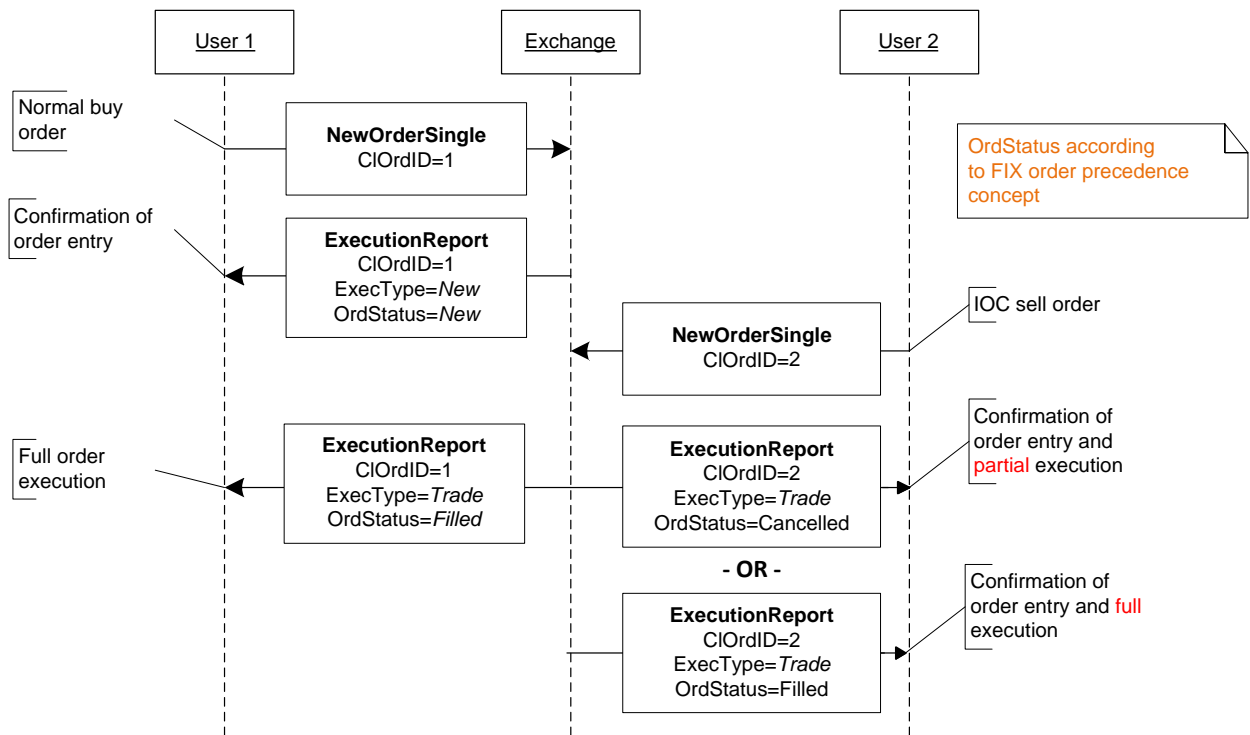


Figure 4: Non-resting orders

4.4 Order Request Identifier

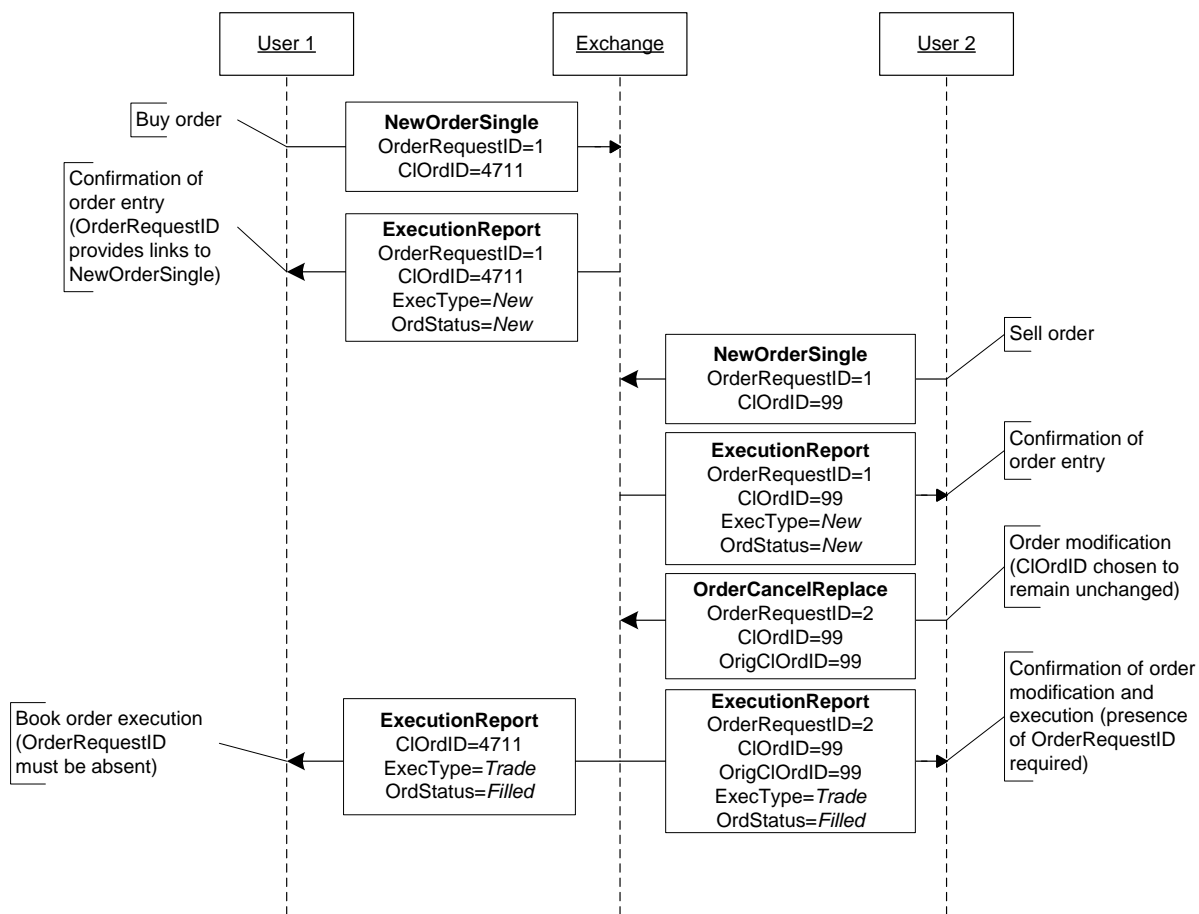


Figure 5: Order request identifier

4.5 MassOrder With Additional Events for all Entries

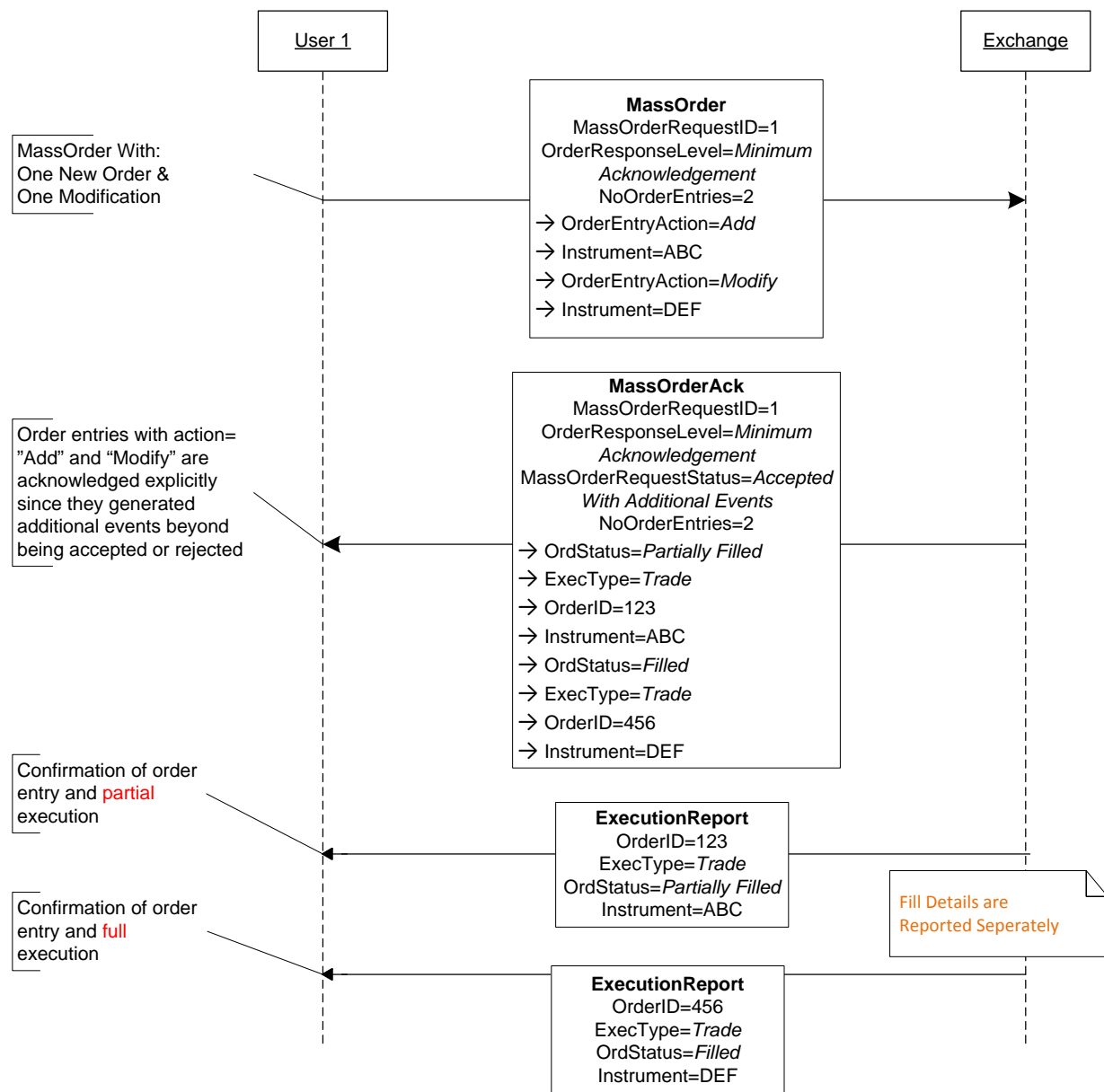


Figure 6: MassOrder with additional events for all entries

4.6 MassOrder With Additional Events for some Entries

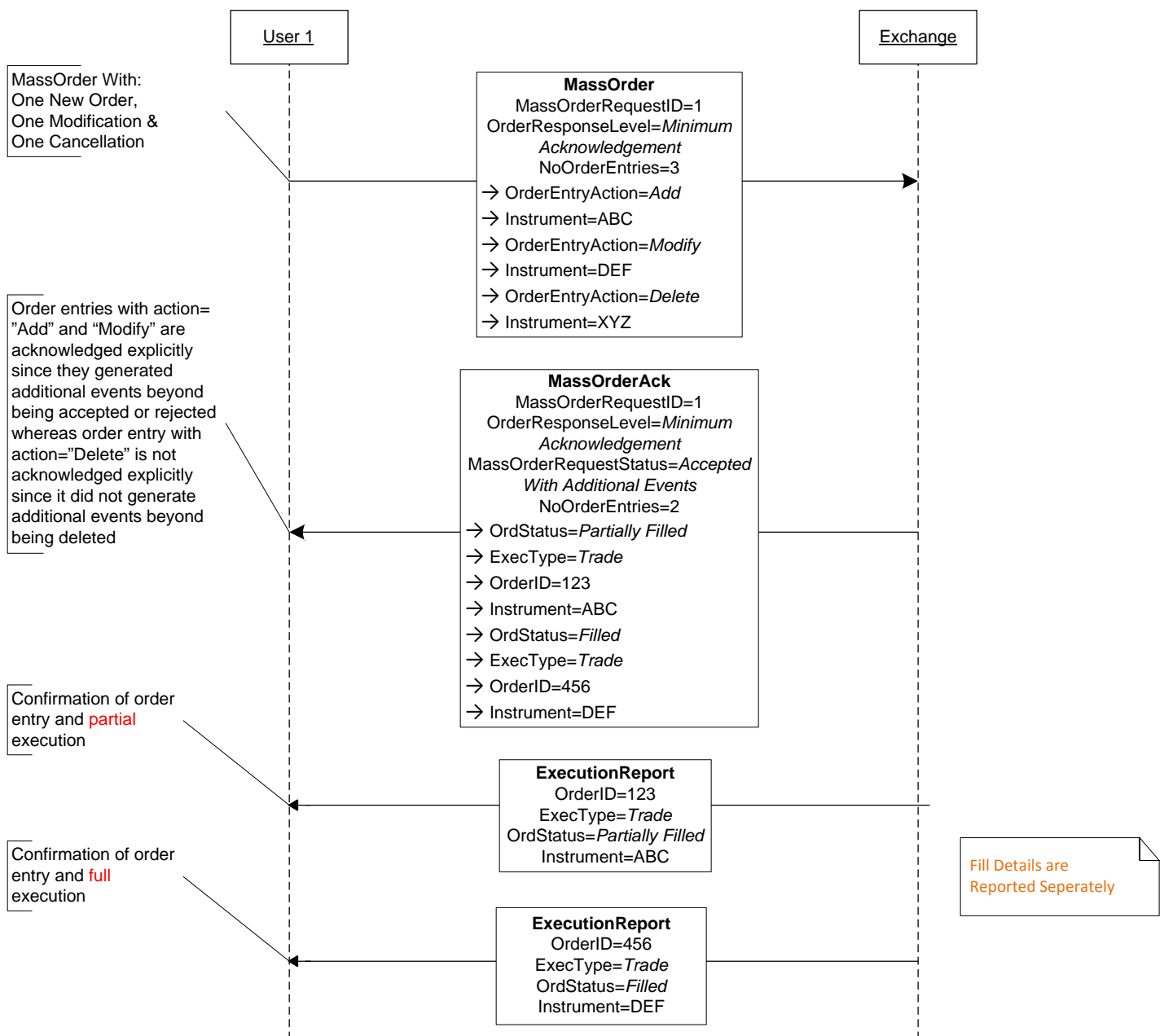


Figure 7: MassOrder with additional events for some entries

4.7 MassOrder Without Additional Events

The first message flow shows the classic case where one or more entries are rejected. Rejections are not considered to be additional events. Entries are either accepted or rejected and may lead to additional events if they are accepted. The second message flow shows a use case where the acknowledgement needs to provide an entry even though all entries were accepted. The entry is needed to return the order identifier assigned by the receiver of the new order.

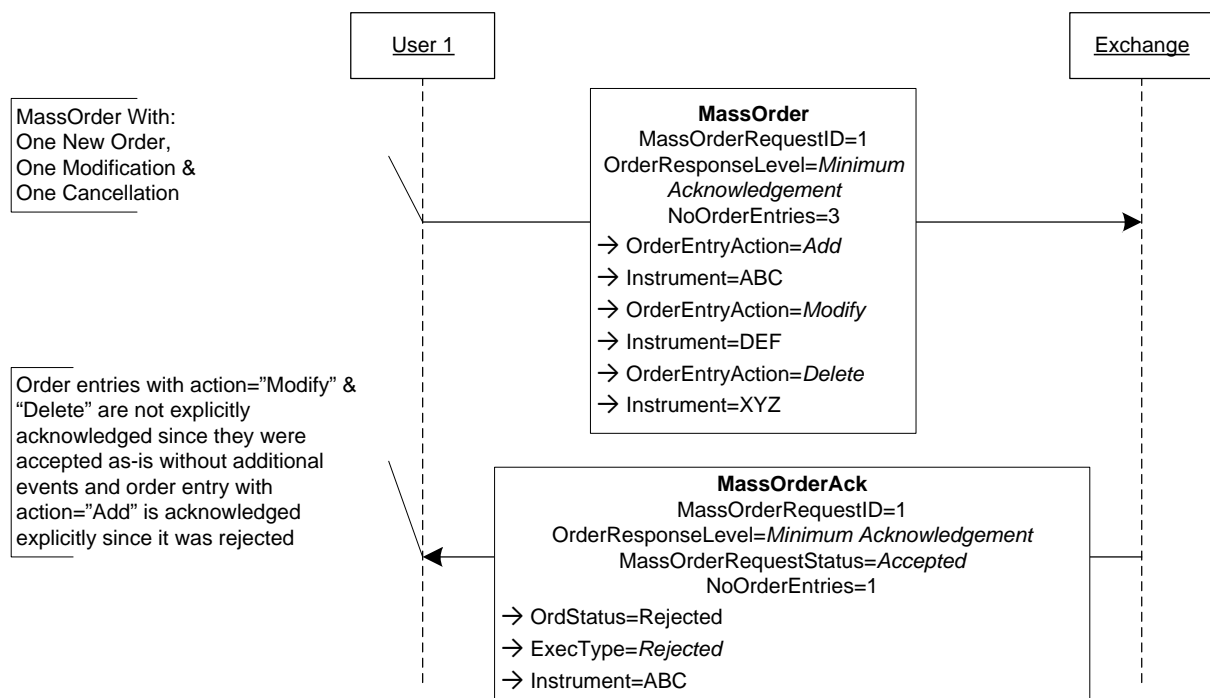


Figure 8: MassOrder without additional events but with rejections

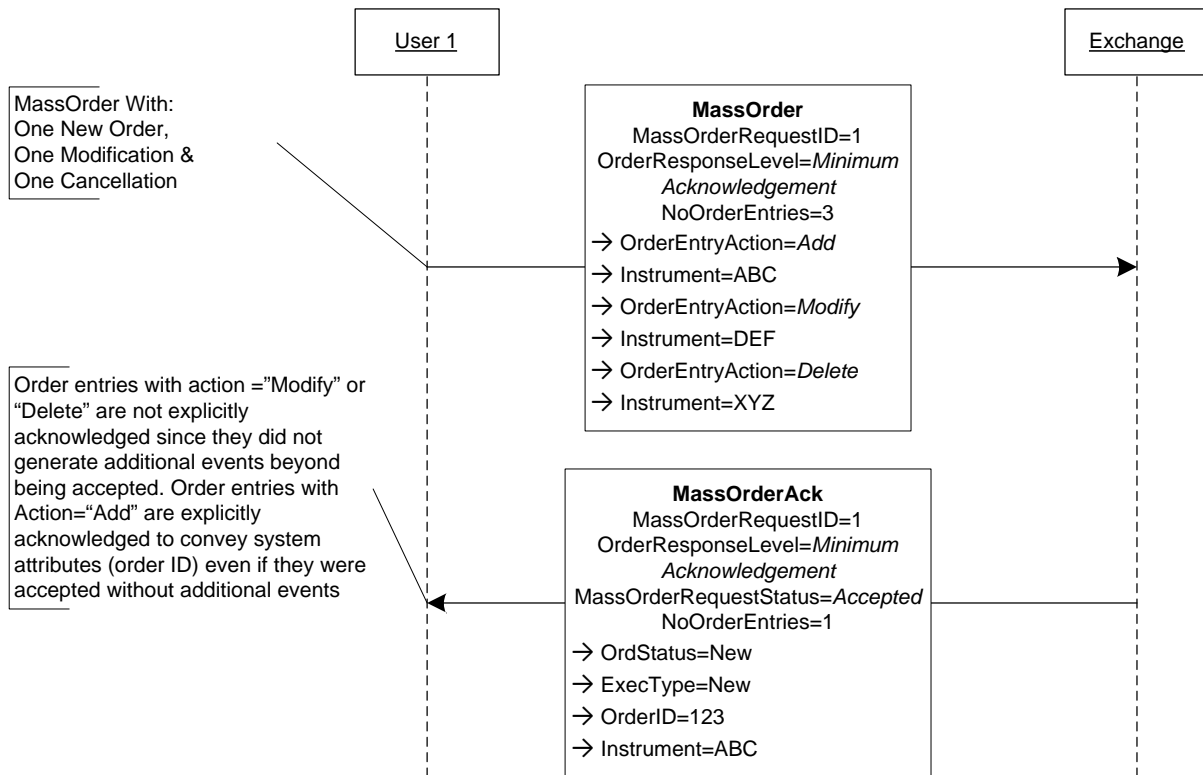


Figure 9: MassOrder without additional events and without rejections

4.8 MassOrder Rejected

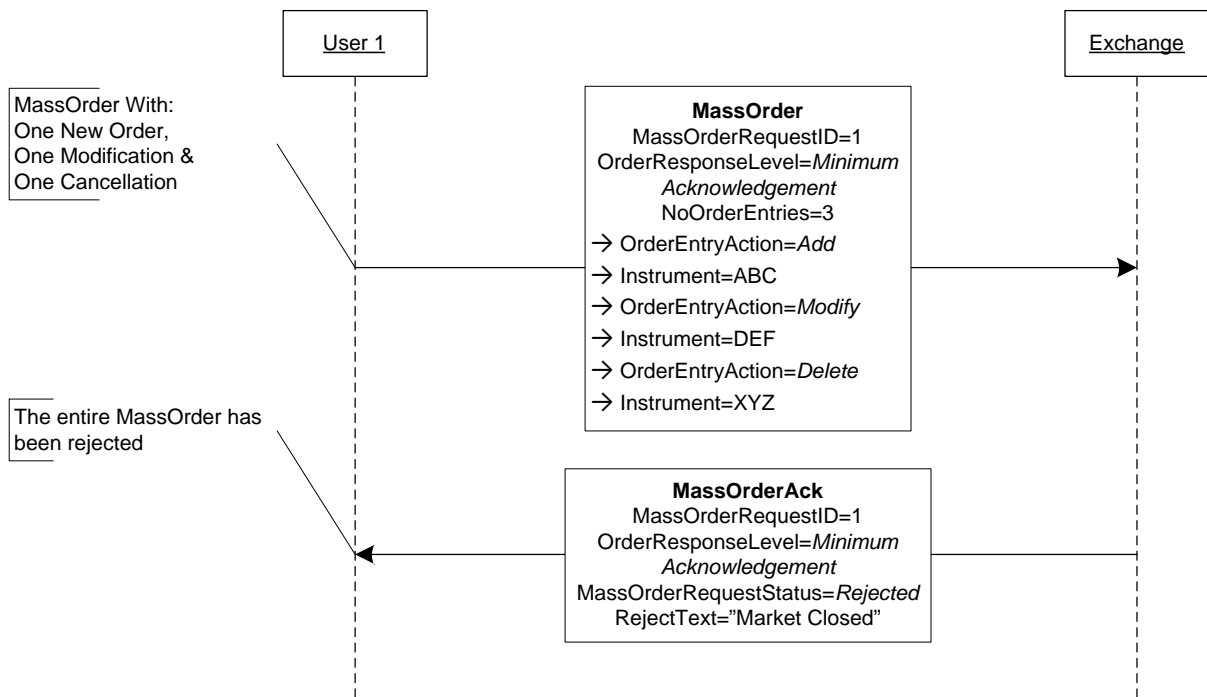


Figure 10: MassOrder rejected

4.9 MassOrder With No Acknowledgements

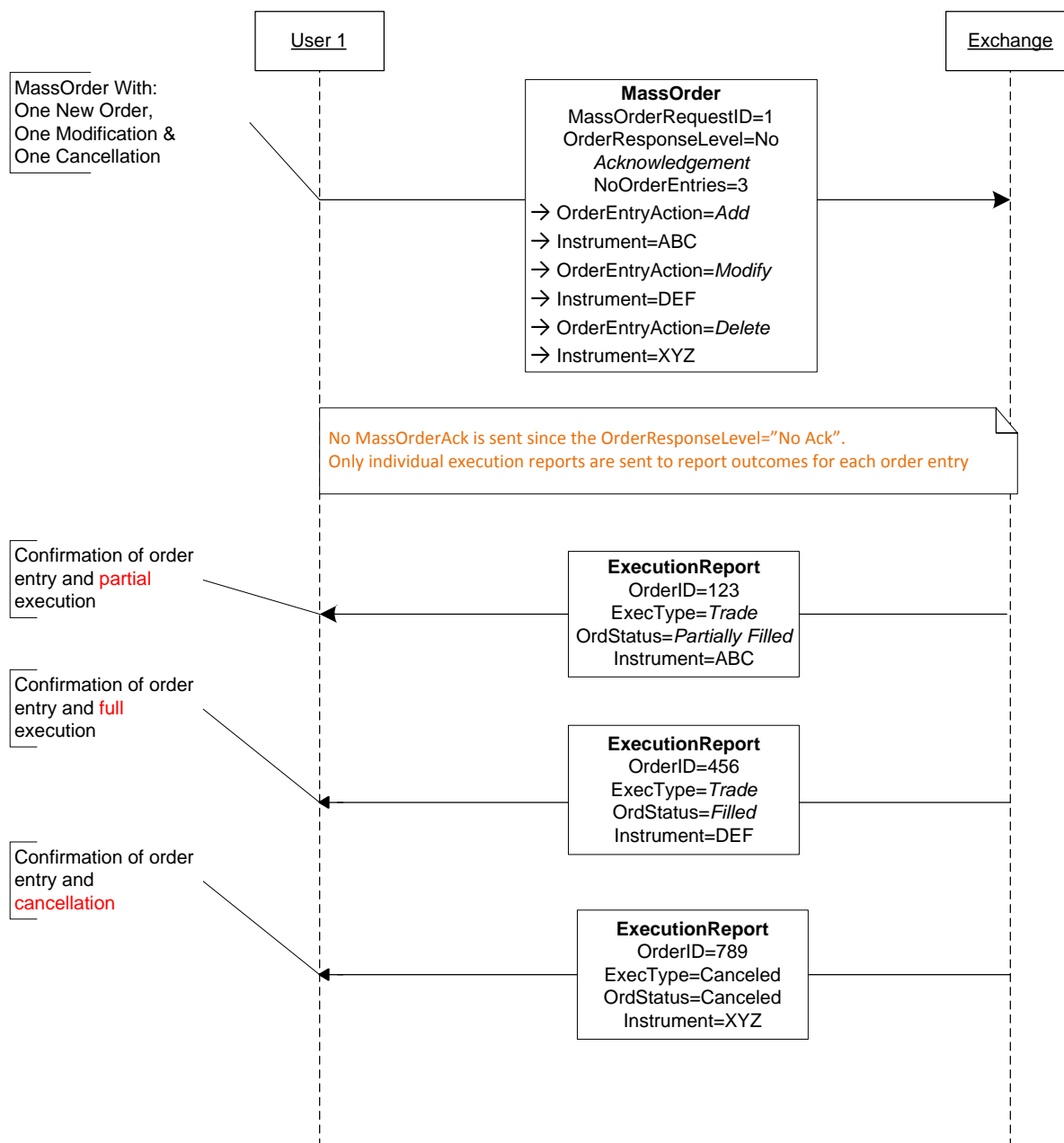


Figure 11: MassOrder with no acknowledgements

4.10 MassOrder With Acknowledgement of Each Order

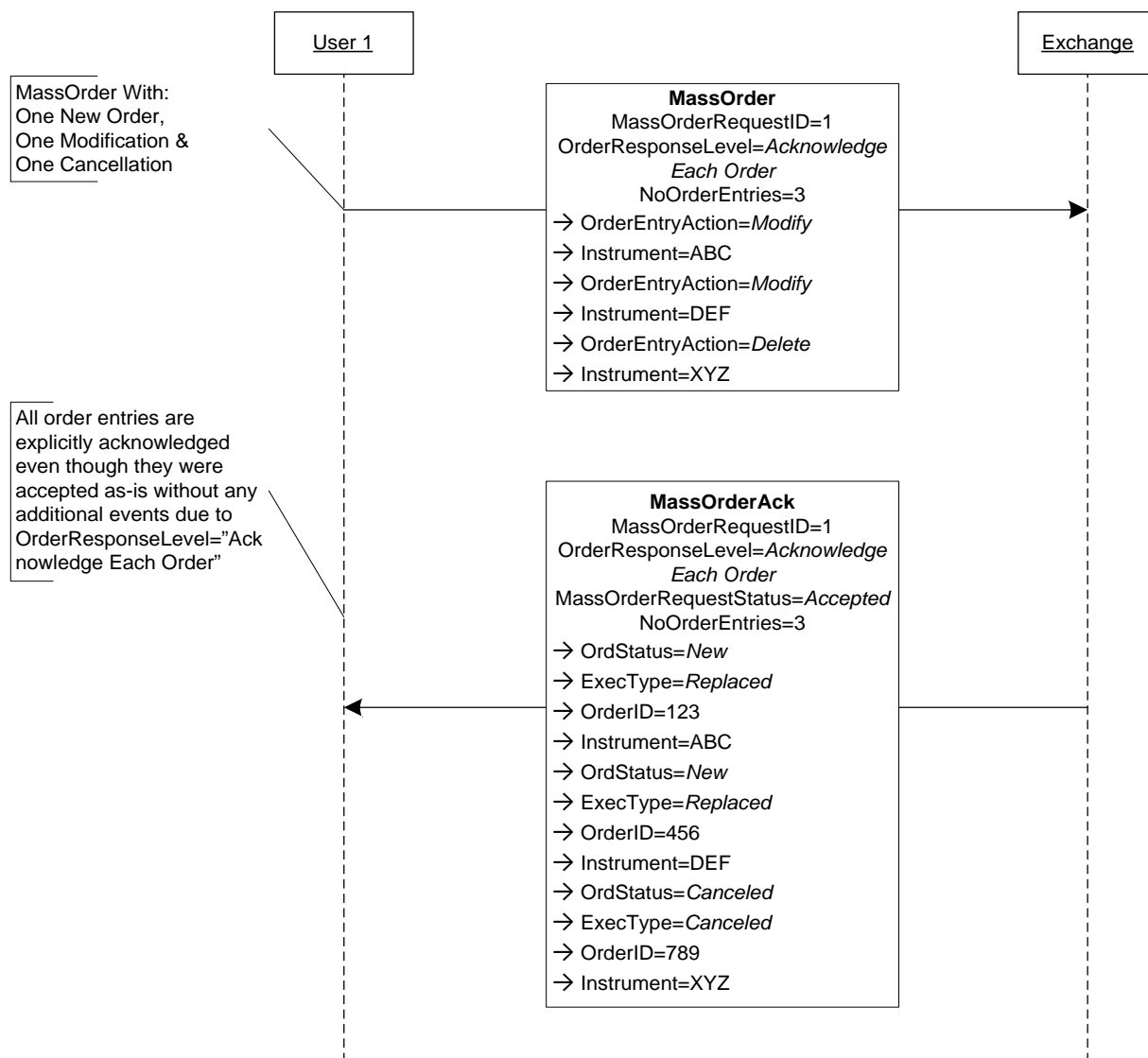


Figure 12: MassOrder with acknowledgement of each order

4.11 MassOrder With Summary Acknowledgement

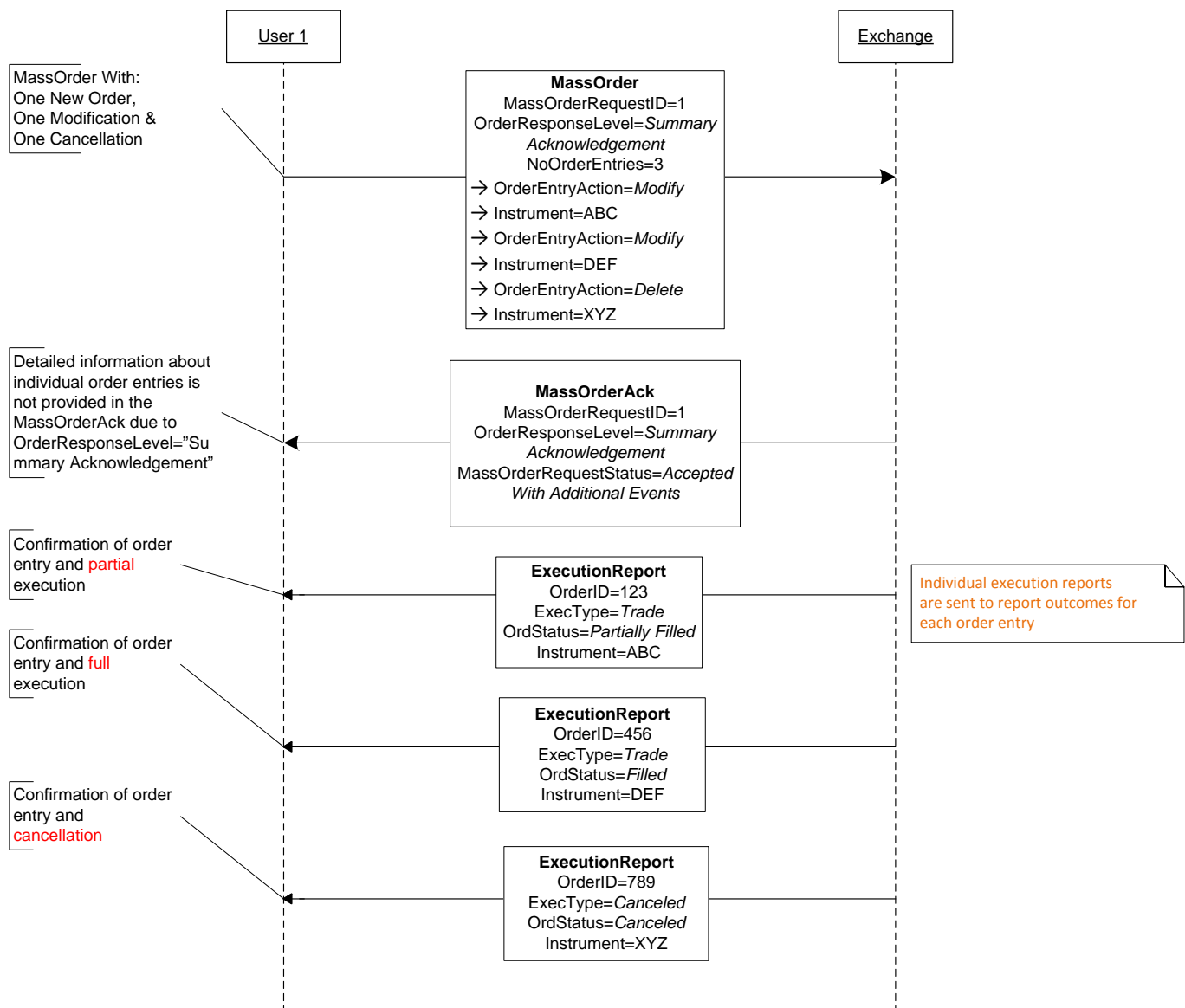


Figure 13: MassOrder with summary acknowledgement

5 FIX Message Tables

5.1 FIX Message ExecutionReport(35=8)

The ExecutionReport message is primarily being changed in terms of the field usage descriptions to increase consistency and allow optimized usages for high performance.

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	ExecutionReport
Message Abbreviated Name (for FIXML)	ExecRpt
Category	SingleGeneralOrderHandling
<u>Action</u>	<u>Change</u>
Message Synopsis	<p>The execution report message is used to:</p> <ol style="list-style-type: none"> 1. confirm the receipt of an order 2. confirm changes to an existing order (i.e. accept cancel and replace requests) 3. relay order status information 4. relay fill information on working orders 5. relay fill information on tradeable or restricted tradeable quotes 6. reject orders 7. report post-trade fees calculations associated with a trade
Message Elaboration	
To be finalized by FPL Technical Office	
MsgType(tag 35) Enumeration	8
Repository Component ID	9

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
StandardHeader		Y				MsgType = 8
component block <ApplicationSequenceControl>		N				For use in drop copy applications. NOT FOR USE in transactional applications.
37	OrderID	Y				OrderID is required to be unique for each chain of orders.
FBD 2422	OrderRequestID	N		NEW		Required if provided on the order message. Echo back the value provided in the order message.
FBD 2423	MassOrderRequestID	N		NEW		Can be used to link execution to the MassOrder(35= FBD) message.
198	SecondaryOrderID	N				Can be used to provide order

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
						id used by exchange or executing system.
526	SecondaryClOrdID	N		CHANGE		In the case of quotes can be mapped to: -QuoteID(117) of a single Quote(35=S) -QuoteEntryID(299) of a MassQuote(35=i) -OrderEntryID(TBD) of a MassOrder(TBD)
527	SecondaryExecID	N				
11	ClOrdID	N		CHANGE		Required when referring to orders that where electronically submitted over FIX or otherwise assigned a ClOrdID(11). In the case of quotes can be mapped to: - QuoteMsgID(1166) of a single Quote(35=S) - QuoteID(117) of a MassQuote(35=i). - MassOrderReportID(TBD24) of a MassOrderAck(35=TBD DK)
41	OrigClOrdID	N		CHANGE		Conditionally required for response to a Cancel or Cancel/Replace request (ExecType(150) = 6 (Pending Cancel, 5 (Replaced), or 4 (Canceled)) when referring to orders that where electronically submitted over FIX or otherwise assigned a ClOrdID(11). ClOrdID(11) of the previous accepted order (NOT the initial order of the day) when canceling or replacing an order.
583	ClOrdLinkID	N				
278	MDEntryID	N				Reference to the MDEntryID(278) of this order or quote in the market data.
693	QuoteRespID	N		CHANGE		Required if responding to a QuoteResponse(35=AJ) message. Echo back the Initiator's value specified in the message.
790	OrdStatusReqID	N		CHANGE		Required if responding to and if provided on the OrderStatusRequest(35=H) message. Echo back the value provided by the

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
						requester.
584	MassStatusReqID	N		CHANGE		Required if responding to a OrderMassStatusRequest(35=AF) . Echo back the value provided by the requester.
961	HostCrossID	N				Host assigned entity ID that can be used to reference all components of a cross; sides + strategy + legs
911	TotNumReports	N		CHANGE		Can be used when responding to an OrderMassStatusRequest(35=AF) to identify the total number of Execution Report(35=8) messages which will be returned.
912	LastRptRequested	N		CHANGE		Can be used when responding to an OrderMassStatusRequest(35=AF) to indicate that this is the last Execution Report(35=8)s message which will be returned as a result of the request.
Component block <Parties>		N		CHANGE		This Specifies party information related to the submitter of the request.
Component block <TargetParties>		N		CHANGE		Identifies Specifies parties not directly associated with or owning the order, who are to be informed to effect processing of the order.
229	TradeOriginationDate	N				
component block <ContraGrp>		N		CHANGE		Specifies contra brokers. Number of ContraBrokers repeating group instances.
66	ListID	N				Required for executions against orders which were submitted as part of a list.
548	CrossID	N				CrossID for the replacement order
551	OrigCrossID	N		CHANGE		Must match original cross order. Same order chaining mechanism as ClOrdID(11)/OrigClOrdID(41) with OrderCancelReplaceRequest(35=G) single order Cancel/Replace .
549	CrossType	N				
2334	RefRiskLimitCheckID	N				
2335	RefRiskLimitCheckIDType	N				Conditionally required when RefRiskLimitCheckID(2334) is specified.
880	TrdMatchID	N				

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
17	ExecID	Y		CHANGE		Unique identifier of execution message as assigned by sell-side (broker, exchange, ECN) (will be 0 (zero) for ExecType(150) = I (Order Status)).
19	ExecRefID	N		CHANGE		Required for ExecType(150) = H (Trade Cancel) and ExecType(150) = G (Trade Correct) ExecType(150) messages.
150	ExecType	Y				Describes the purpose of the execution- report.
FDD 2431	ExecTypeReason	N		NEW		Can be used to provide further detail for ExecType(150) field.
39	OrdStatus	Y				Describes the current state of a CHAIN of orders, same scope as OrderQty, CumQty, LeavesQty, and AvgPx
636	WorkingIndicator	N				For optional use with OrdStatus = 0 (New)
103	OrdRejReason	N				For optional use with ExecType = 8 (Rejected)
1328	RejectText	N				Reason description for rejecting the transaction request.
1664	EncodedRejectTextLen	N				Must be set if EncodedRejectText(1665) field is specified and must immediately precede it.
1665	EncodedRejectText	N				Encoded (non-ASCII characters) representation of the RejectText(1328) field in the encoded format specified via the MessageEncoding(347) field.
378	ExecRestatementReason	N				Required for ExecType = D (Restated).
828	TrdType	N				
2347	RegulatorytransactionType	N				
component block <RegulatoryTradeIDGrp>		N				
1	Account	N				Required for executions against electronically submitted orders which were assigned an account by the institution or intermediary
660	AcctIDSource	N				
581	AccountType	N				Specifies type of account
589	DayBookingInst	N				
590	BookingUnit	N				
591	PreallocMethod	N				
70	AllocID	N				
component block <PreAllocGrp>		N				Pre-trade allocation instructions.

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
63	SettlType	N				
64	SettlDate	N				Takes precedence over SettlType value and conditionally required/omitted for specific SettleType values. Required for NDFs to specify the "value date".
574	MatchType	N				
1115	OrderCategory	N				
544	CashMargin	N				
635	ClearingFeeIndicator	N				
component block <Instrument>		Y		CHANGE		Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages"
component block <FinancingDetails>		N		CHANGE		Insert here the set of "FinancingDetails" (symbology) fields defined in "Common Components of Application Messages"
component block <UndInstrmtGrp>		N				Number of underlyings
54	Side	Y				
2102	ShortMarkingExemptIndicator	N				
1688	ShortSaleExemptionReason	N				Available for optional use when Side(54) = 6 (Sell short exempt).
Component block <Stipulations>		N		CHANGE		Insert here the set of "Stipulations" (repeating group of Fixed Income stipulations) fields defined in "Common Components of Application Messages"
854	QtyType	N				
component block <OrderQtyData>		N		CHANGE		Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages" **IMPORTANT NOTE: OrderQty(38) field is required for single instrument orders unless rejecting or acknowledging an order with CashOrderQty(152) or OrderPercent(516). **
1093	LotType	N				
40	OrdType	N				
423	PriceType	N				
44	Price	N				Required if specified on the order
1092	PriceProtectionScope	N				

<i>Tag</i>	<i>FieldName</i>	<i>Req'd</i>	<i>ICR</i>	<i>Action</i>	<i>Mappings and Usage Comments</i>	<i>FIX Spec Comments</i>
99	StopPx	N				Required if specified on the order
	component block <TriggeringInstruction>	N		CHANGE		Specifies instructions for orders with complex triggers. Insert here the set of "TriggeringInstruction" fields defined in "common components of application messages"
1823	Triggered	N				
	component block <PegInstructions>	N		CHANGE		Insert here the set of "PegInstruction" fields defined in "Common Components of Application Messages"
	component block <DiscretionInstructions>	N		CHANGE		Insert here the set of "DiscretionInstruction" fields defined in "Common Components of Application Messages"
839	PeggedPrice	N				The current price the order is pegged at
1095	PeggedRefPrice	N				The reference price of a pegged order.
845	DiscretionPrice	N				The current discretionary price of the order
1740	TradePriceNegotiationMethod	N				
1742	UpfrontPrice	N				Required if specified on the order.
1741	UpfrontPriceType	N				
847	TargetStrategy	N				The target strategy of the order
	component block <StrategyParametersGrp>	N				Strategy parameter block
850	TargetStrategyPerformance	N				For communication of the performance of the order versus the target strategy
15	Currency	N				
376	ComplianceID	N				
2404	ComplianceText	N				
2351	EncodedComplianceTextLength	N				Must be set if EncodedComplianceText(2352) field is specified and must immediately precede it.
2352	EncodedComplianceText	N				Encoded (non-ASCII characters) representation of the ComplianceText(2404) field in the encoded format specified via the MessageEncoding(347) field.
377	SolicitedFlag	N				
59	TimeInForce	N				Absence of this field indicates Day order
168	EffectiveTime	N				Time specified on the order

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
						at which the order should be considered valid
432	ExpireDate	N		CHANGE		Conditionally required if TimeInForce(59) = 6 (GTD) and ExpireTime(126) is not specified.
126	ExpireTime	N		CHANGE		Conditionally required if TimeInForce(59) = 6 (GTD) and ExpireDate(432) is not specified.
1629	ExposureDuration	N				Conditionally required when TimeInForce(59)=10 (Good for Time)
1916	ExposureDurationUnit	N				
18	ExecInst	N				Can contain multiple instructions, space delimited.
1805	AuctionInstruction					
1057	AggressorIndicator	N				
528	OrderCapacity	N				
529	OrderRestrictions	N				
1815	TradingCapacity					
1091	PreTradeAnonymity	N				
1390	TradePublishIndicator					Applies to trades resulting from the order.
582	CustOrderCapacity	N				
32	LastQty	N		CHANGE		Quantity (e.g. shares) bought/sold on this (last) fill. Required if ExecType(150) = F (Trade) or ExecType(150) = G (Trade Correct) unless FillsGrp or OrderEventGrp is used. —If ExecType(150) = 7 (Stopped), represents the quantity stopped/guaranteed/protected for.
1056	CalculatedCcyLastQty	N		CHANGE		Used for FX trades to express the quantity or amount of the other side of the currency. Conditionally required if ExecType(150) = F (Trade) or G (Trade Correct) and is an FX trade.
1071	LastSwapPoints	N		CHANGE		Optionally used when ExecType(150) = F (Trade) or G (Trade Correct) and is a FX Swap trade. Used to express the swap points for the swap trade event.
652	UnderlyingLastQty	N				
1828	LastQtyVariance					
31	LastPx	N		CHANGE		Price of this (last) fill. Required if ExecType(150) = ExecType = F (Trade) or G (Trade Correct) unless

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
						<p>FillsGrp or OrderEventGrp is used.</p> <p>Should represent the "all-in" (LastSpotRate(194) + LastForwardPoints(195)) rate for F/X orders.).</p> <p>If ExecType(150) = 7 (Stopped), represents the price stopped/guaranteed/protected at.</p> <p>Not required for FX Swap when ExecType(150) = F (Trade) or G (Trade Correct) as there is no "all-in" rate that applies to both legs of the FX Swap.</p>
651	UnderlyingLastPx	N				
669	LastParPx	N		CHANGE		Last price expressed in percent-of-par. Conditionally required for Fixed Income trades when LastPx(31) is expressed in Yield, Spread, Discount or any other price type that is not percent-of-par.
631	MidPx	N				
194	LastSpotRate	N				Applicable for F/X orders
195	LastForwardPoints	N				Applicable for F/X orders
1743	LastUpfrontPrice	N				Upfront Price for CDS transactions. Conditionally required if TradePriceNegotiationMethod(1740) = 4(Percent of par and upfront amount), 5(Deal spread and upfront amount) or 6(Upfront points and upfront amount).
30	LastMkt	N		CHANGE		If ExecType(150) = F (Trade #), indicates the market where the trade was executed. If ExecType(150) = 0 (New #), indicates the market where the order was routed.
1300	MarketSegmentID	N				
100	ExDestination	N				
1133	ExDestinationIDSource	N				
336	TradingSessionID	N				
625	TradingSessionSubID	N				
943	TimeBracket	N				
29	LastCapacity	N				
	component <LimitAmts>	N				Insert here the set of "LimitAmts" fields defined in "Common Components"
151	LeavesQty	Y		CHAN		Quantity open for further

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
				GE		execution. If the OrdStatus(39) is = 4 (Canceled), 3 (Done For TheDay), C (Expired), B (Calculated), or 8 (Rejected) (in which case the order is no longer active) then LeavesQty(151) could be 0, otherwise LeavesQty(151) = OrderQty(38) - CumQty(14).
14	CumQty	Y				Currently executed quantity for chain of orders.
84	CxlQty	N		ADD		Can be used to specify the remaining quantity that was cancelled prior to order reaching terminal state (i.e. when LeavesQty(151)=0). If presentspecified, OrderQty(38) = CumQty(14) + CxlQty(84).
6	AvgPx	N				Not required for markets where average price is not calculated by the market. Conditionally required otherwise.
424	DayOrderQty	N				For GT orders on days following the day of the first trade.
425	DayCumQty	N				For GT orders on days following the day of the first trade.
426	DayAvgPx	N				For GT orders on days following the day of the first trade.
1361	TotNoFills	N		CHANGE		Used to support fragmentation. Sum of NoFills(1362) across all messages with the same ExecID(17).
893	LastFragment	N				Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.
Component block <FillsGrp>		N				Specifies the partial fills included in this ExecutionReport(35=8), mutually exclusive with OrderEventGrp component.
Component block <OrderEventGrp>		N				Specifies the order events included in this ExecutionReport(35=8), mutually exclusive with FillsGrp component.
427	GTBookingInst	N				States whether executions are booked out or accumulated

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
						on a partially filled GT order
75	TradeDate	N				Used when reporting other than current day trades.
60	TransactTime	N				Time the transaction represented by this ExecutionReport(35=8) occurred
113	ReportToExch	N				
	component block <CommissionData>	N		CHANGE		Insert here the set of "CommissionData" fields defined in "Common Components of Application Messages" Note: On a fill/partial fill message, it represents value for that fill/partial fill. On ExecType = B (Calculated), it represents cumulative value for the order. Monetary commission values are expressed in the currency reflected by the Currency(15) field.
	Component block <SpreadOrBenchmarkCurveData>	N		CHANGE		Insert here the set of "SpreadOrBenchmarkCurveData" (Fixed Income spread or benchmark curve) fields defined in "Common Components of Application Messages"
	component block <YieldData>	N		CHANGE		Insert here the set of "YieldData" (yield related) fields defined in "Common Components of Application Messages"
381	GrossTradeAmt	N				
157	NumDaysInterest	N				
230	ExDate	N				
158	AccruedInterestRate	N				
159	AccruedInterestAmt	N				
738	InterestAtMaturity	N				For fixed income products which pay lump-sum interest at maturity.
920	EndAccruedInterestAmt	N				For repurchase agreements the accrued interest on termination.
921	StartCash	N		CHANGE		For repurchase agreements the start (dirty) cash consideration.
922	EndCash	N		CHANGE		For repurchase agreements the end (dirty) cash consideration.
258	TradedFlatSwitch	N				
259	BasisFeatureDate	N				
260	BasisFeaturePrice	N				
238	Concession	N				

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
237	TotalTakedown	N				
118	NetMoney	N		CHANGE		Note: On a fill/partial fill message, it represents value for that fill/partial fill. On a ExecType(150) = B (Calculated) message, it represents cumulative value for the order. Value expressed in the currency reflected by the Currency(15) field.
119	SettlCurrAmt	N		CHANGE		Used to report results of forex accommodation trade.
120	SettlCurrency	N		CHANGE		Used to report results of forex accommodation trade. Required for Non-Deliverable Forwards.
Component block <RateSource>		N				
155	SettlCurrFxRate	N		CHANGE		Foreign exchange rate used to compute SettlCurrAmt(119) from Currency(15) to SettlCurrency(120).
156	SettlCurrFxRateCalc	N		CHANGE		Specifies whether the SettlCurrFxRate(155) should be multiplied or divided
21	HandlInst	N				
110	MinQty	N				
1822	MinQtyMethod					
1089	MatchIncrement	N				
1090	MaxPriceLevels	N				
component block <MatchingInstructions>						
component block <DisplayInstruction>		N		CHANGE		Specifies details for the visibility of reserve orders. Insert here the set of "DisplayInstruction" fields defined in "common components of application messages"
component block <DisclosureInstructionGrp>		N				Specifies instruction to disclose certain order level information in market data.
111	MaxFloor	N				
1816	ClearingAccountType	N				
77	PositionEffect	N				For use in derivatives omnibus accounting
210	MaxShow	N				(Deprecated in FIX.5.0)
775	BookingType	N				Method for booking out this order. Used when notifying a broker that an order to be settled by that broker is to be booked out as an OTC derivative (e.g. CFD or similar). Absence of this field implies regular booking.

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
58	Text	N				
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.
193	SettlDate2	N				(Deprecated in FIX.5.0) Can be used with OrdType = "Forex-Swap" to specify the "value date" for the future portion of a F/X swap.
192	OrderQty2	N				(Deprecated in FIX.5.0) Can be used with OrdType = "Forex-Swap" to specify the order quantity for the future portion of a F/X swap.
641	LastForwardPoints2	N				Can be used with OrdType (40) = "Forex-Swap" to specify the forward points (added to LastSpotRate (194)) for the future portion of a F/X swap.
442	MultiLegReportingType	N				Default is a single security if not specified.
1385	ContingencyType	N				For contingency orders, the type of contingency as specified in the order.
480	CancellationRights	N				For CIV – Optional
481	MoneyLaunderingStatus	N				
513	RegistID	N				Reference to Registration Instructions message for this Order.
494	Designation	N				Supplementary registration information for this Order
483	TransBkdTime	N				For CIV – Optional
515	ExecValuationPoint	N				For CIV – Optional
484	ExecPriceType	N				For CIV – Optional
485	ExecPriceAdjustment	N				For CIV – Optional
638	PriorityIndicator	N				
639	PriceImprovement	N				
851	LastLiquidityInd	N		CHANGE		Applicable only on OrdStatus (39) = 1 (Partially filled) or 2 (Filled).
Component block <ContAmtGrp>		N		CHANGE		Specifies the contract details. Number of contract details in this message (number of repeating groups to follow)
component block <InstrmtLegExecGrp>		N		CHANGE		Specifies the leg executions of a multi-leg order or quote. Number of legs — Identifies a Multi-leg Execution if present and non-zero.

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
797	CopyMsgIndicator	N				
	component block <MiscFeesGrp>	N				Required if any miscellaneous fees are reported.
1380	DividendYield	N				
1028	ManualOrderIndicator	N				
1029	CustDirectedOrder	N				
4030	ReceivedDeptID	N				
1031	CustOrderHandlingInst	N				
1032	OrderHandlingInstSource	N				
1724	OrderOrigination	N				
1725	OriginatingDeptID	N				
1726	ReceivingDeptID	N				
522	OwnerType	N				
	component block <TrdRegTimestamps>	N				
1188	Volatility	N				
1189	TimeToExpiration	N				
1190	RiskFreeRate	N				
811	PriceDelta	N				
1917	CoverPrice	N				
	component block <ThrottleResponse>	N				
1080	RefOrderID	N				
1081	RefOrderIDSource	N				
1806	RefClOrdID	N				
1803	AuctionType	N				
1804	AuctionAllocationPct	N				
1808	LockedQty	N				
1809	SecondaryLockedQty	N				
1807	LockType	N				
1810	ReleaseInstruction	N				
1811	ReleaseQty	N				
1819	RelatedHighPrice	N				
1820	RelatedLowPrice	N				
1821	RelatedPriceSource	N				
	StandardTrailer	Y				

5.2 FIX Message MassOrder([35=tbDJ](#))

The MassOrder([35=tbDJ](#)) is a new message being proposed to enter, modify and delete/[cancel](#) multiple orders with a single message in a high performance environment where only very few order attributes are relevant [or needed](#).

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	MassOrder
Message Abbreviated Name (for FIXML)	MassOrder
Category	OrderMassHandling
Action	New

Message Synopsis	The MassOrder(35= FBD DJ) message can be used to add, modify or delete multiple unrelated orders with a single message. Apart from clearing related attributes, only the key order attributes for high performance trading are available. <u>This distinguishes this message from NewOrderList(35=E) which provides all order attributes for basket/program/list trading.</u>
Message Elaboration	The behavior of individual orders within a MassOrder(35= FBD DJ) may vary depending upon its attributes, e.g. OrdType(40) and TimeInForce(59). Individual orders may be modified or deleted/ <u>cancelled</u> with single order messages such as OrderCancelReplaceRequest (35=G) and OrderCancelRequest(35=F). <u>Each of the orders in the MassOrder(35=FBDDJ) are to be treated as stand-alone individual orders.</u>
To be finalized by FPL Technical Office	
MsgType(tag 35) Enumeration	FBD DJ
Repository Component ID	FBD 146

Tag	Field Name	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
<i>Standard Header</i>		Y				MsgType = FBD DJ
FBD 2423	MassOrderRequestID	Y		NEW		Unique identifier of MassOrder(35= FBD DJ) message as assigned by the submitter <u>of the request.</u>
FBD 2427	OrderResponseLevel	N		NEW		
1301	MarketID	N		ADD		
1300	MarketSegmentID	N		ADD		
<i>Component Block <Parties></i>		N		ADD		This is party information related to the submitter <u>of the request.</u>
1815	TradingCapacity	N		ADD		
1816	ClearingAccountType	N		ADD		
1	Account	N		ADD		
660	AccountIDSource	N		ADD		
581	AccountType	N		ADD		
528	OrderCapacity	N		ADD		
529	OrderRestrictions	N		ADD		
582	CustOrderCapacity	N		ADD		
1028	ManualOrderIndicator	N		ADD		
1031	CustOrderHandlingInst	N		ADD		
60	TransactTime	N		ADD		
58	Text	N		ADD		
<u>354</u>	<u>EncodedTextLen</u>	<u>N</u>		<u>ADD</u>		<u>Must be set if EncodedText(355) field is specified and must immediately precede it.</u>
<u>355</u>	<u>EncodedText</u>	<u>N</u>		<u>ADD</u>		<u>Encoded (non-ASCII characters) representation of the Text(58) field in the encoded format specified via the MessageEncoding(347) field.</u>
1685	ThrottleInst	N		ADD		

Tag	Field Name	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
FBD2432	TotNoOrderEntries	N		NEW		Used to support fragmentation. Sum of NoOrderEntries(2428) within the OrderEntryGrp across all messages with the same MassOrderRequestID(2423).
893	LastFragment	N		ADD		Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.
Component Block <OrderEntryGrp>		Y		NEW		
Standard Trailer		Y				

5.3 FIX Message MassOrderAck(35=~~tb~~DK)

The MassOrderAck(35=~~tb~~DK) is a new message being proposed to acknowledge the entry, modification and deletion/cancellation of multiple orders submitted with a single MassOrder(35=~~tb~~DJ) message in a high performance environment.

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	MassOrderAck
Message Abbreviated Name (for FIXML)	MassOrderAck
Category	OrderMassHandling
Action	New
Message Synopsis	The mass order acknowledgement message is used to acknowledge the receipt of and the status for a MassOrder(35= tb DJ) message.
Message Elaboration	The content of the acknowledgement depends on the setting of the field OrderResponseLevel(tb 2427) in the MassOrder(35= tb DJ) message. Only the order status is provided and not the immediate executions which would lead to ExecutionReport messages.
To be finalized by FPL Technical Office	
MsgType(tag 35) Enumeration	FBD DK
Repository Component ID	FBD 147

Tag	Field Name	Req'd	ICR	Action	Mapping s and Usage Commens	FIX Spec Comments
Standard Header		Y				MsgType = FBD DK
Component Block <ApplicationSequenceControl>		N		ADD		For use in drop copy applications. NOT FOR USE in transactional applications.

Tag	Field Name	Req'd	IC R	Action	Mapping and Usage Comments	FIX Spec Comments
TBD2423	MassOrderRequestID	Y		NEW		Unique identifier of MassOrder(35= TBD2423) message as assigned by the submitter of the request.
TBD2424	MassOrderReportID	N		NEW		Unique identifier of MassOrder(35= TBD2424) message as assigned by the receiver.
TBD2425	MassOrderRequestStatus	Y		NEW		Message level request status.
TBD2426	MassOrderRequestResult	N		NEW		Message level request result.
TBD2427	OrderResponseLevel	N		NEW		Level of response requested from receiver of MassOrder (35= TBD2427) message.
1328	RejectText	N		ADD		
1664	EncodedRejectTextLen	N		ADD		Must be set if EncodedRejectText(1665) field is specified and must immediately precede it.
1665	EncodedRejectText	N		ADD		Encoded (non-ASCII characters) representation of the Text(1328) field in the encoded format specified via the MessageEncoding(347) field.
1301	MarketID	N		ADD		
1300	MarketSegmentID	N		ADD		
Component Block <Parties>		N		ADD		
1815	TradingCapacity	N		ADD		
1816	ClearingAccountType	N		ADD		
1	Account	N		ADD		
660	AccountIDSource	N		ADD		
581	AccountType	N		ADD		
528	OrderCapacity	N		ADD		
529	OrderRestrictions	N		ADD		
582	CustOrderCapacity	N		ADD		
1028	ManualOrderIndicator	N		ADD		
1031	CustOrderHandlingInst	N		ADD		
60	TransactTime	N		ADD		
58	Text	N		ADD		
354	EncodedTextLen	N		ADD		Must be set if EncodedText(355) field is specified and must immediately precede it.

Tag	Field Name	Req'd	ICR	Action	Mapping and Usage Comments	FIX Spec Comments
355	EncodedText	N		ADD		Encoded (non-ASCII characters) representation of the Text(58) field in the encoded format specified via the MessageEncoding(347) field.
797	CopyMsgIndicator	N		ADD		
16 85	ThrottleInst	N		ADD		
TBD2432	TotNoOrderEntries	N		NEW		Used to support fragmentation. Sum of NoOrderEntries(2428) within the OrderEntryAckGrp across all messages with the same MassOrderRequestID(2423).
893	LastFragment	N		ADD		Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented.
Component Block <ThrottleResponse>		N		ADD		
Component Block <OrderEntryAckGrp>		N		NEW		Omit when the entire MassOrder(35=TBDJ) message is rejected or when no order entries are being rejected or accepted with additional events. this component is not used. Presence of this component is also contingent upon the OrderResponseLevel(tbd427) value set by the order submitter in the OrderResponseLevel(tbd) in the MassOrder(35=tbdJ) message.
Standard Trailer		Y				

5.4 FIX Message NewOrderSingle(35=D)

To be completed at the time of the proposal – all information provided will be stored in the repository

Message Name	NewOrderSingle
--------------	----------------

Message Abbreviated Name (for FIXML)	Order
Category	(no change)
Action	Change
Message Synopsis	(no change)
Message Elaboration	(no change)
To be finalized by FPL Technical Office	
MsgType(tag 35) Enumeration	D
Repository Component ID	14

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
	StandardHeader	Y				MsgType = D
11	CIOrdID	Y				
TBD 2422	OrderRequestID	N		NEW		Required if provided on the order message. Echo back the value provided in the order message.
526	SecondaryCIOrdID	N				
<...truncated...>						
	Standard Trailer	Y				

5.5 FIX Message NewOrderMultileg(35=AB)

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	NewOrderMultileg
Message Abbreviated Name (for FIXML)	NewOrdMleg
Category	(no change)
Action	Change
Message Synopsis	(no change)
Message Elaboration	(no change)
To be finalized by FPL Technical Office	
MsgType(tag 35) Enumeration	AB
Repository Component ID	61

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
-----	-----------	-------	-----	--------	-----------------------------	-------------------

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
StandardHeader		Y				MsgType = AB
11	ClOrdID	Y				
FDD 2422	OrderRequestID	N		NEW		Required if provided on the order message. Echo back the value provided in the order message.
526	SecondaryClOrdID	N				
<...truncated...>						
Standard Trailer		Y				

5.6 FIX Message NewOrderCross(35=s)

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	NewOrderCross
Message Abbreviated Name (for FIXML)	NewOrdCrss
Category	(no change)
Action	Change
Message Synopsis	(no change)
Message Elaboration	(no change)
To be finalized by FPL Technical Office	
MsgType(tag 35) Enumeration	s
Repository Component ID	52

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
StandardHeader		Y				MsgType = s
548	CrossID	Y				
FDD 2422	OrderRequestID	N		NEW		Required if provided on the order message. Echo back the value provided in the order message.
549	CrossType	Y				
550	CrossPrioritization	Y				
Component Block <RootParties>		N				
<...truncated...>						
Standard Trailer		Y				

5.7 FIX Message OrderCancelReplaceRequest(35=G)

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	OrderCancelReplaceRequest
Message Abbreviated Name (for FIXML)	OrdCxlRplcReq
Category	(no change)
<u>Action</u>	<u>Change</u>
Message Synopsis	(no change)
Message Elaboration	(no change)
To be finalized by FPL Technical Office	
MsgType(tag 35) Enumeration	G
Repository Component ID	<u>17</u>

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
StandardHeader		Y				MsgType = G
37	OrderID	N				
188 <u>2422</u>	<u>OrderRequestID</u>	<u>N</u>		<u>NEW</u>		<u>Required if provided on the order being replaced (or canceled) message. Echo back the value provided by the requester in the order message.</u>
Component Block <Parties>		N				
<...truncated...>						
Standard Trailer		Y				

5.8 FIX Message MultilegOrderCancelReplace(35=AC)

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	MultilegOrderCancelReplace
Message Abbreviated Name (for FIXML)	MlegOrdCxlRplc
Category	(no change)
<u>Action</u>	<u>Change</u>
Message Synopsis	(no change)
Message Elaboration	(no change)
To be finalized by FPL Technical Office	
MsgType(tag 35) Enumeration	AC

Repository Component ID	62
-------------------------	--------------------

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
StandardHeader		Y				MsgType = AC
37	OrderID	N				
FBD 2422	OrderRequestID	N		NEW		Required if provided on the order being replaced (or cancelled) message. Echo back the value provided by the requester in the order message.
41	OrigClOrdID	N				
<...truncated...>						
Standard Trailer		Y				

5.9 FIX Message CrossOrderCancelReplaceRequest(35=t)

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	CrossOrderCancelReplaceRequest
Message Abbreviated Name (for FIXML)	CrssOrdCxlRplcReq
Category	(no change)
Action	Change
Message Synopsis	(no change)
Message Elaboration	(no change)
To be finalized by FPL Technical Office	
MsgType(tag 35) Enumeration	t
Repository Component ID	53

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
StandardHeader		Y				MsgType = t
37	OrderID	N				
FBD 2422	OrderRequestID	N		NEW		Required if provided on the order being replaced (or cancelled) message. Echo back the value provided by the requester in the order message.
548	CrossID	Y				
<...truncated...>						
Standard Trailer		Y				

5.10 FIX Message OrderCancelRequest(35=F)

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	OrderCancelRequest
Message Abbreviated Name (for FIXML)	OrdCxlReq
Category	(no change)
Action	Change
Message Synopsis	(no change)
Message Elaboration	(no change)
To be finalized by FPL Technical Office	
MsgType(tag 35) Enumeration	F
Repository Component ID	16

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
StandardHeader		Y				MsgType = F
444 2422	OrderRequestID	N		NEW		Required if provided on the order being cancelled message. Echo back the value provided by the requester in the order message.
41	OrigClOrdID	N				
37	OrderID	N				
<...truncated...>						
Standard Trailer		Y				

5.11 FIX Message CrossOrderCancelRequest(35=u)

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	CrossOrderCancelRequest
Message Abbreviated Name (for FIXML)	CrossOrdCxlReq
Category	(no change)
Action	Change
Message Synopsis	(no change)
Message Elaboration	(no change)

To be finalized by FPL Technical Office	
MsgType(tag 35) Enumeration	u
Repository Component ID	54

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
StandardHeader		Y				MsgType = u
37	OrderID	N				
TBD 2422	OrderRequestID	N		NEW		Required if provided on the order being cancelled message. Echo back the value provided by the requester in the order message.
548	CrossID	Y				
551	OrigCrossID	Y				
961	HostCrossID	N				
<...truncated...>						
Standard Trailer		Y				

5.12 FIX Message OrderCancelReject(35=9)

To be completed at the time of the proposal – all information provided will be stored in the repository	
Message Name	OrderCancelReject
Message Abbreviated Name (for FIXML)	OrdCxlRej
Category	(no change)
Action	Change
Message Synopsis	(no change)
Message Elaboration	(no change)
To be finalized by FPL Technical Office	
MsgType(tag 35) Enumeration	9
Repository Component ID	10

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
StandardHeader		Y				MsgType = 9
37	OrderID	Y				
TBD 2422	OrderRequestID	N		NEW		Required if provided on the order cancel or cancel/replacemessage. Echo

Tag	FieldName	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
						back the value provided by the requester in the order message.
198	SecondaryOrderID	N				
<...truncated...>						
	Standard Trailer	Y				

6 FIX Component Blocks

6.1 Component InstrmtLegExecGrp

This component is being changed to allow a reference to a partial fill on the level of the multileg order.

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	InstrmtLegExecGrp
Component Abbreviated Name (for FIXML)	Exec
Component Type	<u> </u> X <u> </u> Block Repeating <u> </u> Block
Category	Common
Action	Change
Component Synopsis	
Component Elaboration	
To be finalized by the FPL Technical Office	
Repository Component ID	2018

Component FIXML Abbreviation: <Exec >						
Tag	Field Name	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
555	NoLegs	***				Number of leg executions.
→	Component Block <InstrumentLeg>	N				Required if NoLegs(555) > 0.
→	685 LegOrderQty	N				Quantity ordered for this leg as provided during order entry.
→	690 LegSwapType	N				Instead of LegOrderQty(685) requests that the sellside calculate LegOrderQty(685) based on opposite Leg.
<...truncated...>						

→	637 138 4	LegExecInst LegLastPx	N				Used to report the execution price assigned to the leg of the multileg instrument.
→	141 8	LegLastQty	N				Quantity executed for this leg.
→	FB D24 21	FillRefID	N		NEW	<FillsGrp> entries are implicitly numbered, starting with 1 for the first repeating group instance.	Use to reference the partial execution of a multi-leg order to which this leg execution belongs.
</Exec>							

6.2 Component OrderEntryGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	OrderEntryGrp
Component Abbreviated Name (for FIXML)	OrderEntry
Component Type	_X_ Block Repeating ___ Block
Category	OrderMassHandling
Action	New
Component Synopsis	Group of order transactions across one or more instruments.
Component Elaboration	
To be finalized by intFPL Technical Office	
Repository Component ID	FB D24 2245

Component FIXML Abbreviation: <OrderEntry>						
Tag	Field Name	Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
FB D24 28	NoOrderEntries	N		NEW		
→	FB D24 29	N		NEW		Required if NoOrderEntries(FBD2428) > 0 .
→	FB D24 30	N		NEW		Unique order entry identification across all entries of a single message. Conditionally required when if neither ClOrdID(11) nor OrderID(37) is not provided.

→	11	ClOrdID	N		ADD	Conditionally required if-when neither OrderEntryID(TBD2430) nor OrderID(37) is not provided.
→	41	OrigClOrdID	N		ADD	Conditionally required if-when OrderEntryAction(TBD2429) is not "1" (Add), and ClOrdID(11) was provided in original order, and a message-chaining model is used.
→	37	OrderID	N		ADD	Conditionally required if-when OrderEntryAction(TBD2429) is not "1" (Add) and neither OrderEntryID(TBD2430) nor ClOrdID(11) is not provided.
→	40	OrdType	N		ADD	Conditionally required for-when OrderEntryAction (TBD2429) = 1 (Add) or 2 (Modify). Only a subset of <u>OrdType(40)</u> values permitted that do not require additional <u>pricing</u> fields with the exception other than of Price(44) field.
→	44	Price	N		ADD	Conditionally required for-when OrdType(40) = 2 (Limit)
→	54	Side	N		ADD	Conditionally required for-when OrderEntryAction(TBD2429) = 1 (Add) or 2 (Modify)
→	59	TimeInForce	N		ADD	Only subset of values permitted that do not require additional fields
→	<i>Component Block</i> <i><OrderQtyData></i>		N		ADD	Conditionally required for-when OrderEntryAction(TBD2429) = 1 (Add) or 2 (Modify)
→	<i>Component Block</i> <i><Instrument></i>		N		ADD	Required if <u>NoOrderEntries(TBD2432) > 0.</u>
<i></OrderEntry></i>						

6.3 Component OrderEntryAckGrp

To be completed at the time of the proposal – all information provided will be included in the repository	
Component Name	OrderEntryAckGrp
Component Abbreviated Name (for FIXML)	OrderEntryAck
Component Type	<input checked="" type="checkbox"/> Block Repeating <input type="checkbox"/> Block
Category	OrderMassHandling
Action	NEW
Component Synopsis	Acknowledgment for a group of order transactions across one or more instruments.
Component Elaboration	The acknowledgement may or may not echo back input values of from the submitter submission but it has to provide the current status of the each order including the impact of immediate executions or suspensions.
To be finalized by intFPL Technical Office	
Repository Component ID	TBD2246

Component FIXML Abbreviation: <OrderEntryAck>							
Tag	Field Name		Req'd	ICR	Action	Mappings and Usage Comments	FIX Spec Comments
TBD2428	NoOrderEntries		Y/N		NEW		
→	39	OrdStatus	Y/N		ADD		Required if NoOrderEntries(tbd2428) > 0.
→	150	ExecType	Y/N		ADD		Required if NoOrderEntries(tbd2428) > 0.
→	TBD2431	ExecTypeReason	N		NEW		
→	TBD2429	OrderEntryAction	N		NEW		
→	TBD2430	OrderEntryID	N		NEW		Conditionally required if/when neither ClOrdID(11) nor OrderID(37) is not provided.
→	11	ClOrdID	N		ADD		Conditionally required if/when neither OrderEntryID(tbd2430) nor OrderID(37) is not provided.
→	41	OrigClOrdID	N		ADD		ClOrdID(11) of the previous non rejected order (NOT the initial order of the day) when canceling or replacing an order . Conditionally required if/when ClOrdID(11) is provided and a message-chaining model is used.

→	37	OrderID	N		ADD		Conditionally required if when neither OrderEntryID(td 2430) nor ClOrdID(11) is not provided.
→	103	OrdRejReason	N		ADD		
→	14	CumQty	N		ADD		Use to explicitly provide executed quantity.
→	151	LeavesQty	N		ADD		Use to explicitly provide remaining quantity.
→	84	CxlQty	N		ADD		Use to explicitly provide cancelled quantity.
→	40	OrdType	N		ADD		
→	44	Price	N		ADD		
→	54	Side	N		ADD		
→	59	TimeInForce	N		ADD		
→	<i>Component Block</i> <OrderQtyData>		N		ADD		
→	<i>Component Block</i> <Instrument>		N		ADD		
</OrderEntryAck>							

7 Category Changes

To be completed at the time of the proposal – all information provided is stored in the repository	
Category Name	[enter the category name here]
Section	__PreTrade __Trade __PostTrade __Infrastructure
Category Synopsis	[enter the category synopsis here]
Category Elaboration	[enter the category elaboration here]
To be finalized by FPL Technical Office	
Category Filename	

Appendix A - Data Dictionary

Tag	FieldName	Action	Datatype	Description	FIXML Abbreviation	Add to / Deprecate from Message type or Component block
FBD24 21	FillRefID	NEW	String	A reference to either the value of the FillExecID(1363) or an implicit position of a fills instance in the FillsGrp component.	@FillRefID	Add to component InstrmtLegExecGrp
FBD24 22	OrderRequestID	NEW	Int	Unique message identifier for an order request as assigned by the <u>owner-submitter</u> of the <u>requestorder</u> .	@OrdReqID	Add to messages NewOrderSingle, NewOrderMultileg, NewOrderCross, OrderCancelReplaceRequest, MultilegOrderCancelReplace, CrossOrderCancelReplaceRequest, OrderCancelRequest, CrossOrderCancelRequest, OrderCancelReject, ExecutionReport
FBD24 23	MassOrderRequestID	NEW	String	Unique message identifier for a mass order request as assigned by the <u>owner-submitter</u> of the orders.	@MassOrdReqID	Add to messages MassOrder MassOrderAck ExecutionReport
FBD24 24	MassOrderReportID	NEW	String	Unique message identifier for a mass order request as assigned by the receiver of the orders.	@MassOrdRptID	Add to message MassOrderAck

Tag	FieldName	Action	Datatype	Description	FIXML Abbreviation	Add to / Deprecate from Message type or Component block
FDD24 25	MassOrderRequestStatus	NEW	int	Status of mass order request. Valid values: 1=Accepted (Symbolic name: [Accepted]) 2=Accepted with additional events (Symbolic name: [AcceptedWithAdditionalEvents]) 3=Rejected (Symbolic name: [Rejected])	@ReqStat	Add to message MassOrderAck
FDD24 26	MassOrderRequestResult	NEW	int, Reserved 100	Request Result result of mass order request. Valid values: 0=Successful (Symbolic name: [Successful]) 1=Response level not supported (Symbolic name: [ResponseLevelNotSupported]) 2=Invalid market (Symbolic name: [InvalidMarket]) 3=Invalid market segment (Symbolic name: [InvalidMarketSegment]) 99=Other (Symbolic name: [Other])	@ReqRslt	Add to message MassOrderAck
FDD24 27	OrderResponseLevel	NEW	int	The level of response requested from receiver of mass order messages. A default value should be bilaterally agreed. Valid values: 0=No acknowledgement (Elaboration: Responses are provided through one or more ExecutionReport(35=8) messages.)	@OrdRspLvl	Add to messages MassOrder MassOrderAck

Tag	FieldName	Action	Datatype	Description	FIXML Abbreviation	Add to / Deprecate from Message type or Component block
				<p>(Symbolic name: [NoAck])</p> <p>1=Minimum acknowledgement (Elaboration: The minimum is any information to explain why the requested transaction was refused or led to additional events, e.g. immediate execution of an order that was entered or modified.) (Symbolic name: [MinimumAck])</p> <p>2=Acknowledge each order (Elaboration: The number of entries in the response is identical to the number of entries in the request.) (Symbolic name: [AckEach])</p> <p>3=Summary acknowledgement (Elaboration: Responses are provided through a single MassOrderAck(35=TBDDK) without entries and one or more ExecutionReport(35=8) messages.) (Symbolic name: [SummaryAck])</p>		
TBDDK 28	NoOrderEntries	NEW	NumInGrp	Number of order entries.		Add to components: OrderEntryGrp OrderEntryAckGrp

Tag	FieldName	Action	Datatype	Description	FIXML Abbreviation	Add to / Deprecate from Message type or Component block
FBDD 24 29	OrderEntryAction	NEW	Char	Specifies the action to be taken for the given order. Valid values: 1=Add (Symbolic name: [Add]) 2=Modify (Symbolic name: [Modify]) 3=Delete / Cancel (Symbolic name: [Delete]) 4=Suspend (Symbolic name: [Suspend]) 5=Release (Symbolic name: [Release])	@OrdEntryActn	Add to messages components OrderEntryGrp OrderEntryAckGrp MassOrder MassOrderAck
FBDD 24 30	OrderEntryID	NEW	Int	Unique identifier for an order within a single MassOrder(35= FBDDJ) message that can be used as a reference in the MassOrderAck(35= FBDDK) message.	@OrdEntryID	Add to messages components OrderEntryGrp OrderEntryAckGrp MassOrder MassOrderAck
FBDD 24 31	ExecTypeReason	NEW	Int Reserved 00Plus	The initiating event when an ExecutionReport(35=8) is sent. Valid values: 1 = Order added upon request (Symbolic name: [OrdAddedOnRequest]) 2 = Order replaced upon request (Symbolic name: [OrdReplacedOnRequest]) 3 = Order cancelled upon request (Symbolic name: [OrdCxlOnRequest]) 4 = Unsolicited order cancellation (Symbolic name: [UnsolicitedOrdCxl])	@ExecTypRsn	Add to message ExecutionReport Add to components OrderEntryAckGrp

Tag	FieldName	Action	Datatype	Description	FIXML Abbreviation	Add to / Deprecate from Message type or Component block
				5 = Non-resting order added upon request (Symbolic name: [NonRestingOrdAddedOnRequest])		
				6 = Order replaced with non-resting order upon request (Symbolic name: [OrdReplacedWithNonRestingOrdOnRequest])		
				7 = Trigger order replaced upon request (Symbolic name: [TriggerOrdReplacedOnRequest])		
				8 = Suspended order replaced upon request (Symbolic name: [SuspendedOrdReplacedOnRequest])		
				9 = Suspended order cancelled upon request (Symbolic name: [SuspendedOrdCxlOnRequest])		
				10 = Order cancellation pending (Symbolic name: [OrdCxlPending])		
				11 = Pending cancellation executed (Symbolic name: [PendingCxlExecuted])		
				12 = Resting order triggered (Symbolic name: [RestingOrdTriggered])		
				13 = Suspended order activated (Symbolic name: [SuspendedOrdActivated])		
				14 = Active order suspended (Symbolic name: [ActiveOrdSuspended])		

Tag	FieldName	Action	Datatype	Description	FIXML Abbreviation	Add to / Deprecate from Message type or Component block
				15 = Order expired (Symbolic name: [OrdExpired])		
FDD24 32	TotNoOrderEntries	NEW	int	Totals number of orders for a mass order or its acknowledgment being fragmented across multiple messages.	@TotNoOrdEntries	Add to messages MassOrder MassOrderAck
1796	OrderEventType	CHANGE	int	The type of event affecting an order. The last event type within the OrderEventGrp component indicates the ExecType(150) value resulting from the series of events (ExecType(150) values are shown in brackets). Valid values: ... 10=Triggered (L=Triggered or Activated by System) FDD11=Activated (L=Triggered or Activated by System) (Symbolic name: [Activated]) ...	@Typ	

Tag	FieldName	Action	Datatype	Description	FIXML Abbreviation	Add to / Deprecate from Message type or Component block
1823	Triggered	CHANGE	Int	<p>Indicates whether order has been triggered during its lifetime. Applies to cases where original information, e.g. OrdType(40), is modified when the order is triggered.</p> <p>Valid values: 0=Not triggered (default) 1=Triggered (Elaboration: Order triggered by any (other) trigger condition.)</p> <p>TBD-2 = Stop order triggered (Symbolic name: [StopOrderTriggered])</p> <p>TBD-3 = One Cancels the Other (OCO) order triggered (Symbolic name: [OCOOrderTriggered])</p> <p>TBD-4 = One Triggers the Other (OTO) order triggered (Symbolic name: [OTOOrderTriggered])</p> <p>TBD-5 = One Updates the Other (OUO) order triggered (Symbolic name: [OUOOrderTriggered])</p>	@Trgrd	

Appendix B - Glossary Entries

Term	Definition	Field where used

Appendix C - Abbreviations

Term	Proposed Abbreviation	Proposed Messages, Components, Fields where used

Appendix D - Usage Examples

Immediate Order Execution Upon Entry or Modification

Order entered with immediate partial fill

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	<Fills Grp>	Comment
1	New Order(X)				100				
2		Execution(X)	Rejected	Rejected	100	0	0		If order is rejected by sell-side (broker, exchange, ECN)
2		Execution(X)	Trade	Partially Filled	100	5	0	FillQty=5 FillPx=99.5	Single message used to convey that order was immediately partially filled

Order entered with immediate complete fill

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	<Fills Grp>	Comment
1	New Order(X)				100				
2		Execution(X)	Rejected	Rejected	100	0	0		If order is rejected by sell-side (broker, exchange, ECN)
2		Execution(X)	Trade	Filled	100	100	0	FillQty=100 FillPx=99.5	Single message used to convey that order was immediately fully filled

Order entered without execution, followed by modification and immediate partial fill

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u><Fills Grp></u>	<u>Comment</u>
1	New Order(X)				100				Price entered at 99.6
2		Execution(X)	Rejected	Rejected	100	0	0		If order is rejected by sell-side (broker, exchange, ECN)
2		Execution(X)	New	New	100	0	100		Order entered without execution
3	OrderCancel Replace(Y,X)				100				Price modified to 99.5
4		OrderCancel Reject(X)	Rejected	Rejected	100				If order modification is rejected
4		Execution(X)	Trade	Partially Filled	100	5	0	FillQty=5 FillPx=99.5	Single message used to convey that order was immediately partially filled

Order entered without execution, followed by modification and immediate complete fill

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u><Fills Grp></u>	<u>Comment</u>
1	New Order(X)				100				Price entered at 99.6
2		Execution(X)	Rejected	Rejected	100	0	0		If order is rejected by sell-side (broker, exchange, ECN)
2		Execution(X)	New	New	100	0	100		Order entered without execution
3	OrderCancel Replace(Y,X)				100				Price modified to 99.5
4		OrderCancel Reject(X)	Rejected	Rejected	100				If order modification is rejected
4		Execution(X)	Trade	Filled	100	100	0	FillQty=100 FillPx=99.5	Single message used to convey that order was immediately fully filled

Multiple Order Execution Upon Entry or Modification

The following examples show orders that may rest on the book but are executed more than once upon entry or modification as part of a single match event.

Order entered with more than one immediate partial fill but not a complete fill

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStatus</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u><Fills Grp></u>	<u>Comment</u>
1	New Order(X)				100				
2		Execution(X)	Rejected	Rejected	100	0	0		If order is rejected by sell-side (broker, exchange, ECN)
2		Execution(X)	Trade	Partially Filled	100	20	0	FillQty=5 FillPx=99.5 FillQty=15 FillPx=99.6	Single message used to convey that order was immediately and partially filled at two different prices

Order entered with more than one immediate partial fill resulting in a complete fill

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStat us</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u><Fills Grp></u>	<u>Comment</u>
1	New Order(X)				100				
2		Execution(X)	Rejected	Rejected	100	0	0		If order is rejected by sell-side (broker, exchange, ECN)
2		Execution(X)	Trade	Filled	100	100	0	FillQty=20 FillPx=99.5 FillQty=80 FillPx=99.6	Single message used to convey that order was immediately and fully filled at two different prices

Non-Resting Orders

The following examples show Fill-Or-Kill as well as Immediate-Or-Cancel orders that are either not filled at all or (partially) filled at more than one price. They cannot rest on the book, i.e. the remainder is cancelled upon entry.

FOK order entered without execution

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStat us</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u><Fills Grp></u>	<u>Comment</u>
1	New Order(X)				100				Order is FOK
2		Execution(X)	Rejected	Rejected	100	0	0		If order is rejected by sell-side (broker, exchange, ECN)
2		Execution(X)	Canceled	Canceled	100	0	0		Single message used to convey that order cannot be immediately and fully filled

FOK order entered with multiple executions

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStat us</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u><Fills Grp></u>	<u>Comment</u>
1	New Order(X)				100				Order is FOK
2		Execution(X)	Rejected	Rejected	100	0	0		If order is rejected by sell-side (broker, exchange, ECN)
2		Execution(X)	Trade	Filled	100	100	0	FillQty=20 FillPx=99.5 FillQty=80 FillPx=99.6	Single message used to convey that order was immediately and fully filled at two different prices (Note that OrdStatus Filled takes precedence over Canceled)

IOC order entered without execution

<u>Time</u>	<u>Message Received</u> (ClOrdID, OrigClOrdID)	<u>Message Sent</u> (ClOrdID, OrigClOrdID)	<u>Exec Type</u>	<u>OrdStat us</u>	<u>Order Qty</u>	<u>Cum Qty</u>	<u>Leaves Qty</u>	<u><Fills Grp></u>	<u>Comment</u>
1	New Order(X)				100				Order is IOC
2		Execution(X)	Rejected	Rejected	100	0	0		If order is rejected by sell-side (broker, exchange, ECN)
2		Execution(X)	Canceled	Canceled	100	0	0		Single message used to convey that order cannot be immediately filled

IOC order entered with partial and multiple executions

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exec Type	OrdStatus	Order Qty	Cum Qty	Leaves Qty	<Fills Grp>	Comment
1	New Order(X)				100				Order is IOC
2		Execution(X)	Rejected	Rejected	100	0	0		If order is rejected by sell-side (broker, exchange, ECN)
2		Execution(X)	Trade	Canceled	100	20	0	FillQty=5 FillPx=99.5 FillQty=15 FillPx=99.6	Single message used to convey that order was immediately and partially filled at two different price levels (Note that OrdStatus Canceled takes precedence over Filled)

Mass Order Transactions with and without Additional Events

MassOrder for new orders with additional events (OrderResponseLevel 1=Minimum acknowledgement)

Time	Message Received (MassOrder RequestID)	Message Sent (MassOrder RequestID)	Order Entry Action	Instrument	Mass Order Request Status	Ord Status	Exec Type	Order ID	Comment
1	Mass Order(X)		Add	ABC					MassOrder submitted with three order entries, the first one being an IOC order that is partially filled
			Add	DEF					
			Add	XYZ					
2		MassOrderAck (X)			Rejected				If mass order is rejected by sell-side (broker, exchange, ECN)
2		MassOrderAck (X)	Add	ABC	Accepted With Additional Events	Canceled	Trade	123	MassOrderAck conveys explicit acknowledgements for all three order entries since all were at least partially filled
			Add	DEF		Partially Filled	Trade	456	
			Add	XYZ		Filled	Trade	789	

MassOrder for new and existing orders with additional events and rejections (OrderResponseLevel 1=Minimum acknowledgement)

Time	Message Received (MassOrder RequestID)	Message Sent (MassOrder RequestID)	Order Entry Action	Instrument	Mass Order Request Status	Ord Status	Exec Type	Order ID	Comment
1	Mass Order(X)		Add	ABC					MassOrder submitted with three order entries
			Modify	DEF					
			Delete	XYZ					
2		MassOrderAck (X)			Rejected				If mass order is rejected by sell-side (broker, exchange, ECN)
2		MassOrderAck (X)	Add	ABC	Accepted With Additional Events	Rejected	Rejected	456	MassOrderAck conveys explicit acknowledgement only for rejected new order and modification resulting in a partial fill
			Modify	DEF		Partially Filled	Trade		

**MassOrder for existing orders without additional events
 (OrderResponseLevel 1=Minimum acknowledgement)**

<u>Time</u>	<u>Message Received</u> (MassOrder RequestID)	<u>Message Sent</u> (MassOrder RequestID)	<u>Order Entry Action</u>	<u>Instrument</u>	<u>Mass Order Request Status</u>	<u>Ord Status</u>	<u>Exec Type</u>	<u>Order ID</u>	<u>Comment</u>
1	Mass Order(X)		Modify	ABC					MassOrder submitted with three order entries
			Modify	DEF					
			Modify	XYZ					
2		MassOrderAck (X)			Rejected			If mass order is rejected by sell-side (broker, exchange, ECN)	
2		MassOrderAck (X)			Accepted			MassOrderAck with empty repeating group as all modifications were accepted and did not lead to any (partial) fills	

**MassOrder for new and existing orders without additional events but with rejections
 (OrderResponseLevel 1=Minimum acknowledgement)**

<u>Time</u>	<u>Message Received</u> (MassOrder RequestID)	<u>Message Sent</u> (MassOrder RequestID)	<u>Order Entry Action</u>	<u>Instrument</u>	<u>Mass Order Request Status</u>	<u>Ord Status</u>	<u>Exec Type</u>	<u>Order ID</u>	<u>Comment</u>
1	Mass Order(X)		Add	ABC					MassOrder submitted with three order entries
			Modify	DEF					
			Delete	XYZ					
2		MassOrderAck (X)			Rejected			If mass order is rejected by sell-side (broker, exchange, ECN)	
2		MassOrderAck (X)	Add	ABC	Accepted	Rejected	Rejected		MassOrderAck conveys explicit acknowledgement only for rejected order transactions
			Modify	DEF		Rejected	Rejected	456	

**MassOrder for new and existing orders without additional events and without rejections
 (OrderResponseLevel 1=Minimum acknowledgement)**

<u>Time</u>	<u>Message Received</u> (MassOrder RequestID)	<u>Message Sent</u> (MassOrder RequestID)	<u>Order Entry Action</u>	<u>Instrument</u>	<u>Mass Order Request Status</u>	<u>Ord Status</u>	<u>Exec Type</u>	<u>Order ID</u>	<u>Comment</u>
1	Mass Order(X)		Add	ABC					MassOrder submitted with three order entries
			Add	DEF					
			Delete	XYZ					
2		MassOrderAck (X)			Rejected			If mass order is rejected by sell-side (broker, exchange, ECN)	
2		MassOrderAck (X)	Add	ABC	Accepted	New	New	123	MassOrderAck conveys explicit acknowledgement only for new orders to return exchange order ID
			Add	DEF		New	New	456	

Mass Order Transactions with Different Levels of Acknowledgement

MassOrder With No Acknowledgement (OrderResponseLevel = 0)

Time	Message Received (MassOrder RequestID)	Message Sent (CIOrdID, OrigCIOrdID)	Order Entry Action	Instrument	Mass Order Request Status	Ord Status	Exec Type	Order ID	Comment
1	Mass Order(X)		Add	ABC					MassOrder submitted with three order entries
			Modify	DEF					
			Delete	XYZ					
2		MassOrderAck (X)			Rejected				If mass order is rejected by sell-side (broker, exchange, ECN)
3		Execution(X)		ABC		Partially Filled	Trade	123	Confirmation of order entry and partial execution
4		Execution(X)		DEF		Filled	Trade	456	Confirmation of order entry and full execution
5		Execution(X)		XYZ		Canceled	Canceled	789	Confirmation of order entry and cancellation

MassOrder With Minimum Acknowledgement (OrderResponseLevel = 1)

Time	Message Received (MassOrder RequestID)	Message Sent (MassOrder RequestID)	Order Entry Action	Instrument	Mass Order Request Status	Ord Status	Exec Type	Order ID	Comment
1	Mass Order(X)		Add	ABC					MassOrder submitted with three order entries
			Add	DEF					
			Delete	XYZ					
2		MassOrderAck (X)			Rejected				If mass order is rejected by sell-side (broker, exchange, ECN)
2		MassOrderAck (X)	Add	ABC	Accepted	New	New	123	MassOrderAck conveys explicit acknowledgement only for new orders to return exchange order ID
			Add	DEF		New	New	456	

MassOrder With Acknowledgement of Each Order Entry (OrderResponseLevel = 2)

Time	Message Received (MassOrder RequestID)	Message Sent (MassOrder RequestID)	Order Entry Action	Instrument	Mass Order Request Status	Ord Status	Exec Type	Order ID	Comment
1	Mass Order(X)		Modify	ABC					MassOrder submitted with three order entries
			Modify	DEF					
			Delete	XYZ					
2		MassOrderAck (X)			Rejected				If mass order is rejected by sell-side (broker, exchange, ECN)
2		MassOrderAck (X)	Modify	ABC	Accepted	New	Replaced	123	MassOrderAck conveys explicit acknowledgement for all order entries regardless of whether they were accepted with additional events or not
			Modify	DEF		New	Replaced	456	
			Delete	XYZ		Canceled	Canceled	789	

MassOrder With Summary Acknowledgement (OrderResponseLevel = 3)

<u>Time</u>	<u>Message Received</u> (MassOrder RequestID)	<u>Message Sent</u> (CIOrdID, OrigCIOrdID)	<u>Order Entry Action</u>	<u>Instrument</u>	<u>Mass Order Request Status</u>	<u>Ord Status</u>	<u>Exec Type</u>	<u>Order ID</u>	<u>Comment</u>
1	Mass Order(X)		Modify Modify Delete	ABC DEF XYZ					MassOrder submitted with three order entries
2		MassOrderAck (X)			Rejected				If mass order is rejected by sell-side (broker, exchange, ECN)
3		MassOrderAck (X)			Accepted With Additional Events				MassOrderAck does not have detailed information about individual order entries
4		Execution(X)		ABC		Partially Filled	Trade	123	Confirmation of order entry and partial execution
5		Execution(X)		DEF		Filled	Trade	456	Confirmation of order entry and full execution
6		Execution(X)		XYZ		Canceled	Canceled	789	Confirmation of order entry and cancellation