GPW WATS 2.02 FIX Order Gateway Specification (FIX 5.0)

Date: 21.02.2025 | Version: GPW1.5.1

Fundusze Europejskie
Rzeczpospolita Polska
Unia Europejska Europejskie Fundusze Strukturalne i Inwestycyjne
NCBR Narodowe Centrum Badań i Rozwoju

CONTENTS

1. DISCLAIMER

This document is for information purposes only and does not form any part of contractual documentation.

Reasonable care has been taken to ensure details contained within are accurate and not misleading at the time of publication. Warsaw Stock Exchange is not responsible for any errors or omissions contained in this document.

Warsaw Stock Exchange reserves the right to treat information contained in this document subject to later change without prior notice.

This document contains confidential information to Warsaw Stock Exchange and may not be reproduced, disclosed, or used in whole or part, in any manner, without prior written consent from the owner of this document. Information included in this document shall be maintained and exercised with adequate security measures necessary to protect confidential information from unauthorized access or disclosure.

In case of sections of documentation at a **High** level work progress according to the current version of *GPW WATS Advancement of Documentation*, Warsaw Stock Exchange will endeavor to limit changes to these sections of documents to those related to:

1. correcting errors in the documentation or in the software;
2. clarification of the documentation content or removing ambiguity;
3. implementation of approved change requests or;
4. regulatory changes.

2. PREFACE

This document has been prepared by Warsaw Stock Exchange in order to help in the implementation process of GPW WATS trading platform.

2.1. TARGET AUDIENCE

This document has been prepared for development staff, Independent Software Vendors who produce software integrated with GPW WATS, analysts, market participants and, all clients who want to deepen their knowledge about GPW WATS.

2.2. DOCUMENT PURPOSE

This document provides a technical description of FIX Order Gateway available in GPW WATS.

Bearing in mind this document has been prepared by Warsaw Stock Exchange in order to assist customers in their change deployment process which results from the trading platform replacement program and all efforts have been taken to ensure the accuracy of the document, some of the proposals mentioned in this document may still be subject to change or may require further supplements. If the final approach has not yet been taken it is clearly stated.

2.3. ASSOCIATED DOCUMENTS

GPW WATS FIX Order Gateway Specification is a part of GPW WATS documentation set.

Please check the following documents to learn about the construction of Trading System.

- GPW WATS 1.01 Trading System.

Please check the documentation of the trading protocols supported by GPW WATS.

- GPW WATS 2.01 Native Order Gateway Specification,
- **GPW WATS 2.02 FIX Order Gateway Specification** (this document).

Please check the description of the communication with Data Distribution Service.

- GPW WATS 3.01 Market Data Protocol.

Please check the description of the communication with Internet Data Distribution System.

- GPW WATS 3.02 Internet Data Distribution System,
- GPW WATS 3.03 Streaming Messages for IDDS,
- GPW WATS 3.04 Rest API Messages for IDDS.

Please check the additional documentation, which explains other services provided within GPW WATS.

- GPW WATS 4.01 Drop Copy Gateway,
- GPW WATS 4.02 Post Trade Gateway,
- GPW WATS 5.01 Risk Management Gateway.

Please check the additional documentation describing the following:

- GPW WATS 2.03 Rejection Codes,
- GPW WATS 2.04 BenDec Message Definition Format,

- GPW WATS 4.03 Contract Notes,
- GPW WATS 6.01 Connectivity,
- GPW WATS 6.02 (ENG) Short Code Record Keeping,
- GPW WATS 6.02 (PL) Mapowanie Short Code,
- GPW WATS 6.03 Short-Long Mapper User Guide.

It is recommended to read ***GPW WATS 1.01 Trading System*** document first.

3. Document History

Version	Date	Description
0.51	29.06.2023	The initial publication of the documentation.
0.52	26.07.2023	Publication of v0.52.
0.53	18.08.2023	Message change: ExecutionReport field Price (44) has been change to required="N".
0.54	20.09.2023	Messages change: Logon (A): Fixed order of messages. Logout (5): SessionStatus field added. Header: ApplVerID field added.
0.56	10.11.2023	TestEvent (UT) message has been added.
0.57	30.11.2023	The following changes has been added: 4.2. Party Identifiers • updates(Clearing Firm and Contra Firm added, additional info added for shortcodes parties) 7. Messages • TestEvent message deleted. • PartyIDSource extended for messages(New order Single, AE, AR) to cover: B = BIC N = Legal Entity Identifier (LEI) • BusinessRejectReason enhancements, • SelfMatchPreventionID enabled as already implemented. 4.6. New chapter describing quotes handling. • 4.4.11 Self-Trade prevention chapter added, • 4.4.6 – Adnotation to new field 2431(ExecTypeReason).
0.58	18.12.2023	Publication of v0.58.
0.59	25.01.2024	Execution Report (8) message: ExecRestatementReason (378) has been replaced with ExecTypeReason (2431) field. **4.4.11** STP mechanism description has been extended and new ExecTypeReason=107 added. **5.3.** Message throttling description has been extended. **7.16** SecondaryTradeReportID has been added to Trade Capture Report message. **7.17** SecondaryTradeReportID has been added to Trade Capture Report Ack message. **4.5.1.2** SecondaryTradeReportID has been added.
0.62	25.03.2024	The document has been updated with: • modifications of orders in terms of GTD validity (4.4.3.3), • GTT TIF added to Iceberg, Stop Limit, Stop Loss order types, • 4.4.11 Self-Trade Prevention; ExecTypeReaseon field set to 103 - Cancel by STP, • enhancement of the description in section 4.5 Trade Capture for Cross and Block trades, • new section 4.8 Mass Cancellation, • Diagrams have been corrected accordingly to messages v0.62. The following messages in section 7 have been updated: **Order Cancel Replace (G)** • ExpireDate field added. **Trade Capture Report**

Version	Date	Description
		• ExecType field: New value added - 0-New.
1.0	30.04.2024	Section 4.6 MassQuote has been upgraded. Section 4.7 Cancel On Disconnect has been upgraded. Section 5.2 Failover and Recovery has been delated. Instead, chapter 6.5 Recovery has been added. Businnes Message Reject (J) message: • BusinessRelectRefID - field added • Reject (3) message: • RefTagID - field added • RefAppVerID - field added New Order Single (D) message: BIC value deleted from PartyIDSource • 1 - Executing Firm and 17 - Contra Firm values have been added to PartyRole field • 0 - None value has been deleted from PartyRoleQualifier • ExecInst - field added • FeeStructureID - field added Execution Report (8) message: • BIC value deleted from PartyIDSource • 1 - Executing Firm and 17 - Contra Firm values have been added to PartyRole field • 0 - None value has been deleted form PartyRoleQualifier (2376) • ExecTypeReason - field added • AvgPx - field added • PriorityIndicator - field added • CopyMsgIndicator - field added • ExecInst - field added • FeeStructureID - field added Order Cancel Replace Request (G) message: • NoPartyIDs - field added • PartyID - field added • PartyIDSource - field added • PartyRole - field added • PartyRoleQualifier - field added • InitialDisplayQty - field added Order Cancel Reject message • RejectText - field deleted Order Cancel Request message: • NoPartyIDs - field added • PartyID - field added • PartyIDSource - field added • PartyRole - field added • PartyRoleQualifier - field added Trade Capture Report (AE) message • AlgorithmicTradeIndicator - field added • FeeStructureID - field added • MatchStatus - field deleted • GrossTradeAmt - field deleted • TrdRptStatus - field deleted

Version	Date	Description
		Trade Capture Report Ack (AR) message • MarketSegmentID - field added • MarketID - field added • Text - field added Mass Quote (i) message • QuoteType - field deleted • QuoteResponseLevel - field deleted • FeeStructureID - field added • SelfMatchPreventionID - field added Mass Quote Ack(B) • QuoteType - field deleted • QuoteResponseLevel - field deleted • FeeStructureID - field added • SecurityID - field added • SecurityIDSource - field added • BidPx - field added • OfferPx - field added • BidSize - field added • OfferSize - field added • SelfMatchPreventionID - field added Request For Execution (UE) message added Market Maker Command (UM) message added Market Maker Command Ack (UC) message added Order Mass Cancel Report (r) message added Order Mass Cancel Request (r) message added BidOfferUpdate(UU) message added Kinematics have been updated accordingly to the latest version.
1.1	28.06.2024	Exec_type description corrections - chapter 7.16. Tables of messages have been updated and edited: new style has been implemented for readability purposes. Max length and data types have been populated for some fields. In Trade Capture Report (AE) message the field feeStructureId has changed the location in the message. In Trade Capture Report Ack (AR) message the field feeStructureId has changed the location in the message.
1.1.2.	9.08.2024	**Trade Capture Report Ack (AR):** • New field: RejectText. **4.4.8 Order Modifications**: Parties modification has been deleted from the document as it is not consistent with the System. Short Code modification has been added to the list of modifications. Other minor changes and adjustments.
1.2	18.09.2024	**Trade Capture Report Ack (AR):** • Conditional of SettlDate (64) field has been amended. TradeReportRefId and TradeReportId field lengths have been increased to 21 characters. **Reject (3)** • BusinessRejectReason field name has been changed to SessionRejectReason One-party – Accepted by Counterparty diagram has been amended.
1.3	17.10.2024	**GTC Order - Restatement** diagram has been updated.
1.4	6.12.2024	Unpublished version. All changes in this version have been documented in v1.5.

Version	Date	Description
1.5	3.02.2025	total size = number of Lots price = price per unit (smallest portion) of instrument Section 4.2 Party Identifiers has been updated Messages have been updated to the latest version of the Fix contract. The changes include: **Standard Header:** • ApplVerID - deleted • SenderLocationID - added • LastMsgSeqNumProcessed - deleted **Logon:** • RawDataLength - deleted • RawData - deleted • MaxMessageSize - added • TestMessageIndicator - added • Username - added • Password - added • NewPassword - added **Business Message Reject (J):** • BusinessRejectReason - values updated **Reject (3)** • SessionRejectReason - values updated **New Order Single (D)** • Parties group has been updated **Execution Report (8)** • Parties group has been updated • RejectText – deleted • AvgPx - deleteg **Order Cancel Replace Request (G)** • Parties group has been updated **Order Cancel Reject (9)** • Parties group has been updated **Order Cancel Request (F)** • Parties group has been updated **Trade Capture Report (AE)** • Parties group has been updated **Trade Capture Report Ack (AR)** • Parties group has been updated • RejectText - deleted **Mass Quote** • Parties group has been updated **Mass Quote Ack (B)** • Parties group has been updated
1.5.1	21.02.2025	**Standard Header:** • MsgType - values updated • ApplVerID - field added • SenderSubID - field deleted • SenderLocationID - field deleted • TargetSubID - field deleted

Version	Date	Description
		Logon: • RawDataLength - field added • RawData - field added • ResetSeqNumFlag - values changed to implemented • NextExpectedMsgSeqNum - Conditional added • MaxMessageSize - field deleted • TestMessageIndicator - field deleted • Username - field deleted • Password - field deleted • NewPassword - field deleted • SessionStatus - conditional and description updated; new values added • Text - Conditional and decription updated. **Logout (5):** • SessionStatus - new values added

4. Service Description

4.1. FIX Gateway Service Description

FIX Order Gateway service provides a trading interface based on a FIX standard protocol.

FIX Order Gateway is a point-to-point service which relies on the TCP-IP protocol.

The main service functionalities are:

- order submission,
- order modification,
- order cancellation,
- transaction submission (block and cross trades),
- mass quotes,
- mass cancellation.

FIX Order Gateway service consists of two main layers:

Session layer- designed to maintain reliable and repeatable messaging between peers.

There are number of session layer messages to establish and maintain FIX connection between Trading Client Application and Order Gateway:

- Logon (A) and Logout (5) messages to start and terminate FIX connection,
- Heartbeat (0) and TestRequest (1) messages to maintain a FIX connection in periods of inactivity,
- ResendRequest (2) and SequenceReset (4) to recover lost messages,
- Reject (3) message to inform that received message cannot be further processed because it does not comply with the session layer rules.

All messages (including business messages) are examined on a session layer level, and only when verification is successful, messages are further processed.

The document follows a convention where messages used for exchanging information about orders and transactions are referred to as *business messages*. In the FIX specification, these messages are known as *application messages*.

Business layer – designed to achieve a business purpose, such as submitting and managing orders or transactions.

After FIX connection is established Trading Client Application and Order Gateway can exchange business layer messages. Each business message consists of a number of fields (standard header and trailer) as follows.

Trading Client Application sends business layer messages such as:

- New Order Single (D) to submit a new order,
- Order Cancel Replace Request (G) to change order data,
- Order Cancel Request (F) to cancel a previously submitted order,
- Trade Capture Report (AE) to report a block/cross trade,
- Mass Quote (i) to submit mass quote.

4.2. Party Identifiers

Members use short codes to safely exchange data about parties required by MiFID2/MiFIR for reporting purposes. Participants must additionally provide real data referred by the short codes, in a separate document.

A short code ranges from 4 to 4.294.967.295 (2^32 – 1). The values 0, 1, 2, and 3 are reserved for special purposes. In FIX messages PartyID (448) field contains short codes.

Field PartyRole (452) specifies the role represented by the short code provided in the PartyID (448) field:

- Client member (Party Role = 3),

  Client party role provides information about short codes which identify a natural person or firm or legal entity. Client party role is mandatory if the order is submitted with Order Capacity (528)=A or R. In case of aggregated orders or pending allocations the PartyID (448) field is set to 1 or 2 and PartyRoleQualifier should not be present(is not set). PartyIDSource (447) is set to P – Short Code Identifier.

- Investment decision maker (Party Role = 122),

  An Investment decision maker party role provides information about short codes which identify a natural person or algorithm responsible for the investment decision. An Investment decision maker identifier is mandatory if an order is submitted with Order Capacity (528) = P.

  The PartyID for Investment decision maker value can range from 4 to 4.294.967.295. An Investment decision maker short code is linked to PartyRoleQualifier equal to 24 or 22. PartyRoleQualifier (24) is used when a natural person within the member is responsible for the investment decision. PartyRoleQualifier (22) is used when an algorithm is responsible for the investment decision. PartyIDSource (447) is set to P – Short Code Identifier.

- Executing trader (Party Role = 12),

  An Executing trader party role provides information on short codes which identify a natural person or algorithm responsible for an order execution. The party role Executing trader is mandatory if Order Capacity is equal to both Agency and Principal(528)=A, R .or P). PartyID for Executing trader value ranges from 4 to 4.294.967.295. Executing trader short code is linked to PartyRoleQualifier equal to 24 or 22. PartyRoleQualifier (24) is used when a natural person within the member is responsible for order execution. PartyRoleQualifier (22) is used when an algorithm is responsible for order execution. When the time and location of order execution are determined by the client of the participant then the Executing trader's Party ID has the reserved code =3 (NONE) and PartyRoleQualifier should not be present. PartyIDSource (447) is set to P – Short Code Identifier.

Parties section can also be used to specify additional information, PartyRole (452) can be used to specify role:

- Clearing Firm (Party Role = 4),

  Clearing Firm party role can be optionally used for both CLOB orders and CROSS or BLOCK transactions on FIX trading Gateway, PartyID (448) represents code identifying Clearing Firm and PartyIDSource (447) specify type of code provided in PartyID field. PartyID (448) is set to one of subsequent values: D = Proprietary / Custom code,N = Legal Entity Identifier (LEI),PartyRoleQualifier (2376) is not set(not present) as this value is not applicable to Clearing Firm Party Role.

- Contra Firm (Party Role = 17),

Contra Firm Party Role is mandatory for Single Sided TradeCaptureReports to identify second side of reported transaction. PartyID(448) represents code identifying Contra Firm Participant(second side of the transaction) and PartyIDSource (447) specify type of code provided in PartyID field and is set to D(Proprietary / Custom code).

- Interested Party (Party Role = 33),

  Interested Party is optional and can be set to all market models (CLOB, BLOCK, CROSS, HYBRID).

  PartyID(448) represents 3th party interested in this order or trade and it's value is limited to 8 bytes. PartyRoleQualifier (2376) is not present as this value is not applicable to Interested Party Role.. PartyIDSource (447) is set to D = Proprietary / Custom code.

Field PartyRoleQualifier (2376) provides further qualification of PartyRole (452):

- 22 = Algorithm (applicable to PartyRole values 12 or 122),
- 23 = Firm or legal entity (LEI) (applicable to PartyRole value 3),
- 24 = Natural person (applicable to PartyRole values 3, 12, 122).

The table below provides the summary of the combination of fields in FIX Parties repeating group used to specify short codes.

Party Identifier	FIX Tags
Client – Natural person (National ID)	PartyID (448) = short code Party Role (452) = 3 PartyRoleQualifier (2376) = 24
Client – Legal entity (LEI)	PartyID (448) = short code Party Role (452) = 3 PartyRoleQualifier (2376) = 23
Client's aggregated orders	PartyID (448) = 1 (AGGR) Party Role (452) = 3 PartyRoleQualifier (2376) not set
Client's pending allocation	PartyID (448) = 2 (PNAL) Party Role (452) = 3 PartyRoleQualifier (2376) not set
Investment Decision Maker - Natural person (National ID)	PartyID (448) = short code Party Role (452) = 122 PartyRoleQualifier (2376) = 24
Investment Decision Maker - Algorithm	PartyID (448) = short code Party Role (452) = 122 PartyRoleQualifier (2376) = 22
Executing Trader - Natural person (National ID)	PartyID (448) = short code Party Role (452) = 12 PartyRoleQualifier (2376) = 24
Executing Trader - Algorithm	PartyID (448) = short code Party Role (452) = 12 PartyRoleQualifier (2376) = 22
Executing Trader - Order execution is determined by the client	PartyID (448) = 3 (Client) Party Role (452) = 12 PartyRoleQualifier (2376) not set

4.3. Security Identification

Instruments are identified in FIX protocol messages using numerical GPW WATS identifier. This identifier corresponds to InstrumentID in Market Data messages. The list of available Instrument IDs along with their ISINs and other basic characteristics will be provided on a daily basis via Market Data services (Instrument message).

GPW WATS instruments identifiers will not change over the entire life of an instrument and will not be reused after instrument termination for newly created instruments.

Relevant FIX tags:

SecurityIDSource (22) = 8 (Exchange Symbol)
SecurityID (48) = to InstrumentID in Market Data messages

4.4. Order Management

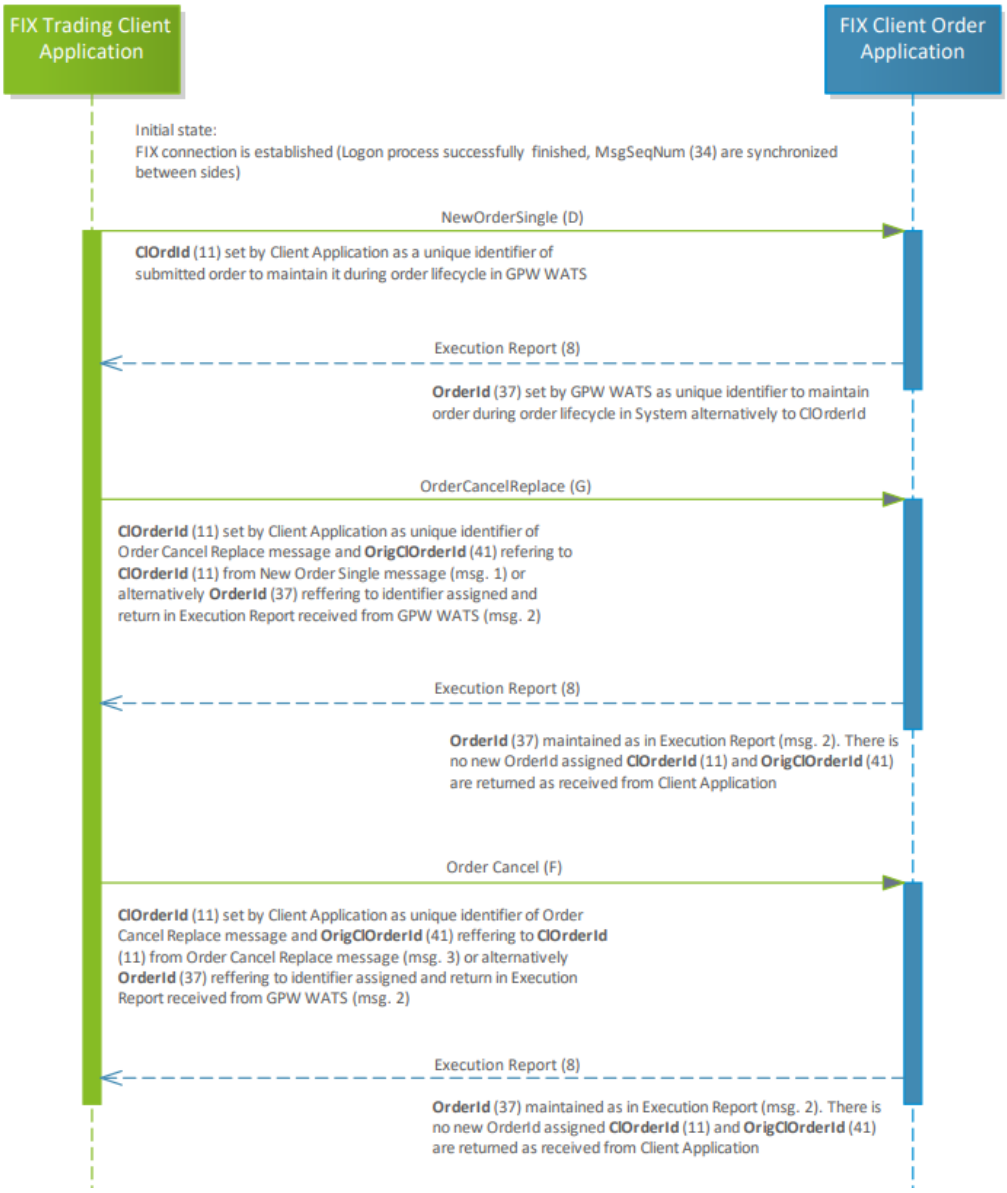
After FIX connection is established (see chapter 7. Establishing Connection) Participant is allowed by GPW WATS to submit new orders and manage existing orders.

New Order Single (35=D) message is used to submit New Order, Order Cancel Replace Request (35=G) is used to modify existing orders (which belongs to Participant), Order Cancel Request (35=F) concludes order lifecycle. After order is canceled, it is removed from GPW WATS. Participant cannot revert its status.

For each of the above messages, if it is received, successfully analyzed GPW WATS responds with Execution Report message (35=8) which reflect the current status and last action done on order.

Each of the above messages can be rejected on either session (35=3), business validation (35=j), application (Execution Report (35=8) or OrderCancel Reject (35=9)) levels due to different reasons.

Figure 1. – Order lifecycle



Order lifecycle of a new order in GPW WATS starts when a new order is submitted and ends when it is cancelled (by user or System), expired by System based on its time validity parameter or when it is fully filled (after one or many trades).

4.4.1. ORDER IDENTIFIERS

The following Tags are utilized across FIX messages to identify Orders in GPW WATS:

- ClOrdID (11),
- OrderID (37),
- MDEntryID (278),
- OrigClOrdID (41).

4.4.1.1. ClOrdID (11)

ClOrdID is a mandatory field for each of the messages dedicated to create and manage orders sent by Participants:

- New Order Single (D),
- Order Cancel/Replace Request (G),
- Order Cancel Request (F),
- Order Mass Cancel Request(q).

or sent by GPW WATS:

- Execution Report (8),
- Order Cancel Reject(9),
- Order Mass Cancel Report(r).

ClOrdID is alphanumeric ASCII free-format string, which can include any character except a control character.

ClOrdID is assigned by Exchange Member, GPW WATS populates previously assigned ClOrdID when it sends an Execution Report back to Exchange Member.

ClOrdID is a transitional unique identifier of Client order. Thus ClOrdID reference to Order can be updated by each accepted modification ClOrdID message which refers to a particular order.

Client Order can be accessed by ClOrdID of the last message which successfully submits or modifies Participant order.

ClOrdID must be unique amongst all Patricipant's orders.

4.4.1.2. OrderID (37)

OrderID is a numeric format unique identifier of an order assigned by Trading System.

OrderID is unique across all trading days, all order books and all Exchange Members and it is persistent for the entire life of an order (i.e. it does not change after order modification).

GPW WATS provides newly generated OrderID in Execution Report returned to Participant each time a new order is submitted, subsequently the same OrderID is returned in Execution Reports when the order is modified or cancelled.

OrderID can be used by Exchange Member (interchangeably with OrigClOrdID) when requesting an order modification (Order Cancel/Replace Request (G) message) or order cancellation (Order Cancel Request (F) message) to identify the relevant order which rests in Trading System. If Exchange Member provides both OrderID and OrigClOrdID in the request to modify or cancel, Trading System will process the OrderID only and ignore OrigClOrdID.

4.4.1.3. MDEntryID (278)

The order identifier assigned by System and published in Market Data (public information). Corresponds to the publicOrderID field in Market Data OrderAdd (9), OrderModify (10), OrderDelete (11), OrderExecute (12) messages.

MDEntryID is a numeric format unique identifier of an order assigned by Trading System.

MDEntryID is unique across all order books and all Exchange Members, but only within a single trading day. On the next trading day, the numbering is reset and starts afresh (i.e. MDEntryID is not persistent for the entire life of an order). Moreover it also changes with each refill of an Iceberg order or each restatement of transfer orders (e.g. orders with time in force set to Good Till Cancel or Good Till Date).

Gateway provides MDEntryID for an order in FIX Execution Report (8) message only. PublicOrderID can be used to relate submitted orders to Market Data, however, orders are modified or cancelled only using the OrderID (37) or ClOrdID (11) fields. MDEntryID is a conditionally required field, when an order is not published in Market Data it is not assigned to an order. STOP orders before activation, orders with TimeInForce (59)= 3 (Immediate Or Cancel) or 4 (Fill Or Kill) which were fully cancelled are not published in Market Data and MDEntryID is not present in Execution Report.

4.4.1.4. OrigClOrdID (41)

OrigClOrdID is used to point order by referring to present ClOrdID of an order.

OrigClOrdID format is identical to ClOrdID - it is alphanumeric ASCII free-format string, which can include any character except a control character.

OrigClOrdID can be used by Exchange Member (interchangeably with OrderID) when requesting an order modification (Order Cancel/Replace Request (G) message) or order cancellation (Order Cancel Request (F) message) to identify the relevant order which rests in Trading System. If Exchange Member provides both OrderID and OrigClOrdID to modify or cancel, Trading System will process the OrderID only and ignore OrigClOrdID.

Figure 2. – Order identifiers usage an example

4.4.2. ORDER TYPES

Exchange Members may submit the following Order Types in the New Order Single (D) message via FIX Order Gateway:

- Limit,
- Market,
- Market To Limit,
- Iceberg (as Limit order type with DisplayQty),
- Stop Limit,
- Stop Loss.

4.4.2.1. Limit Order

Limit order is an order to buy or sell at a stipulated limit or better price, thus for buy order at its limit or lower price and for sell order at its limit or higher price.

Limit order can be executed with one or more counterpart resting orders until it is fully filled.

An unexecuted remainder of a limit order is added to the order book or is cancelled depending on TimeinForce (59).

Relevant FIX tags:

OrdType (40) = 2 (Limit)
Price (44) = Limit Price
TimeInForce (59) = 0 (Day), or 1 (Good Till Cancel), or 2 (At The Opening), or 3 (Immediate Or Cancel), or 4 (Fill Or Kill),or 6 (Good Till Date), or 7 (At The Closing)

4.4.2.2. Market Order

Market order is an order to buy or sell without a specified price. Market order is executed at the best opposing prices with one or more counterpart resting orders.

Unexecuted remainder of a market order is immediately cancelled.

During Continuous Trading market phase, market orders can only be submitted with IOC, FOK, VFA, VFC validity attributes, i.e. TimeInForce (59) = 2, 3, 4, 7.

During Auction market phase, market orders can only be submitted with VFA and VFC validity attributes, i.e. TimeInForce (59) = 2, 7.

Price (44) is not present for Market order type.

Relevant FIX tags:

OrdType (40) = 1 (Market)
TimeInForce (59) = 2 (At The Opening), or 3 (Immediate Or Cancel), or 4 (Fill Or Kill), or 7 (At The Closing)

4.4.2.3. Market To Limit

Market to limit order is an order to buy or sell without a specified price. Market to limit order is executed at the best opposing price level with one or more counterpart resting orders. Market to limit order cannot be executed on many price levels.

Unexecuted remainder of a market to limit order is immediately cancelled.

During Continuous Trading market phase, market to limit orders can only be submitted with IOC, FOK, VFA, VFC validity attributes, i.e. TimeInForce (59) = 2, 3, 4, 7.

During Auction market phase, market to limit orders can only be submitted with VFA and VFC validity attributes, i.e. TimeInForce (59) = 2, 7.

Price (44) is not present for Market to limit order type.

Relevant FIX tags:

OrdType (40) = K (Market to limit)
TimeInForce (59) = 2 (At The Opening), or 3 (Immediate Or Cancel), or 4 (Fill Or Kill), or 7 (At The Closing)

4.4.2.4. Iceberg

Iceberg order is a limit order, where only part of the total quantity is disclosed to the market via Market Data. The rest of the order is hidden (invisible) to the general market. Exchange Member submitting an iceberg order must additionally specify the quantity that will be displayed initially, i.e. DisplayQty (1138), which:

- must be lower than the total order quantity,
- must be greater or equal to the minimum display quantity set in Trading System configuration,
- may be randomized by Trading System within a specified range.

Once the displayed quantity is completely filled, a new portion of the order is disclosed to the Market according to the displayQty (1138) requested at the order entry (with possible randomization).

The value of each Iceberg Order (i.e. Price x OrderQty) at the point of entry or during any modification must be equal or greater than the minimum iceberg size set in the system configuration.

Iceberg orders can only be submitted with DAY, GTC, GTD, GTT validity attributes, i.e. TimeInForce (59) = 0, 1, 6.

Relevant FIX tags:

OrdType (40) = 2 (Limit)
Price (44) = Limit Price
DisplayQty (1138) = Initially Displayed Quantity, expressed in number of Lots
TimeInForce (59) = 0 (Day), or 1 (Good Till Cancel), or 6 (Good Till Date)

4.4.2.5. Stop Limit

Stop limit is an order to buy or sell at a stipulated limit or better price, which is activated and inserted into the order book upon reaching or exceeding (up for buy and down for sell) a predefined price level (called trigger price) by last traded price on the market. Until activation, the stop limit is hidden and cannot interact with the order book. Once activated, the stop limit is processed in the same manner as a limit order with the same validity attribute that the original stop order had prior to activation (nonetheless maintains its original order type = 4).

When submitting or modifying stop limit order, the following conditions must be met:

- for buy order: Price (44) >= Trigger Price (1102) > Last Traded Price (LTP),
- for sell order: Price (44) <= Trigger Price (1102) < Last Traded Price (LTP).

Stop limit can only be submitted with DAY, GTC, GTD, GTT validity attributes, i.e. TimeInForce (59) = 0, 1, 6.

Trigger price (1102) may be modified for an inactive stop order, but once the order is activated, trigger price can no longer be modified.

Relevant FIX tags:

- OrdType (40) = 4 (Stop Limit),
- Price (44) = Limit Price,
- TriggerPrice (1102) = Activation Price,
- TimeInForce (59) = 0 (Day), or 1 (Good Till Cancel), or 6 (Good Till Date),
- Triggered (1823) = 0 (Not triggered) or 2(Stop order activated) - returned in Execution Report (8),
- ExecType (150) = L (Triggered) - returned one time after order is activated.

4.4.2.6. Stop Loss

Stop loss is an order to buy or sell without a specified price, which is activated and inserted into the order book upon reaching or exceeding (up for buy and down for sell) a predefined price level (called trigger price) by the last traded price on the market. Until activation, stop loss is hidden and cannot interact with the order book. Once activated, stop loss is processed in the same manner as a market order with immediate or cancel (IOC) validity attribute (nonetheless maintains its original order type = 3).

When submitting or modifying stop loss order, the following conditions must be met:

- for buy order: trigger price (1102) > last traded price (LTP),
- for sell order: trigger price (1102) < last traded price (LTP).

Stop loss can only be submitted with DAY, GTC, GTD, GTT validity attributes, i.e. TimeInForce (59) = 0, 1, 6.

Trigger price (1102) may be modified for inactive stop order, but once the order is activated, trigger price can no longer be modified.

Relevant FIX tags:

- OrdType (40) = 3 (Stop Loss),
- TriggerPrice (1102) = Activation Price,
- TimeInForce (59) = 0 (Day), or 1 (Good Till Cancel), or 6 (Good Till Date),
- Triggered (1823) = 0 (Not triggered) or 2(Stop order activated)-returned in Execution Report (8),
- ExecType(150) = L (Triggered) - returned one time after order is activated.

4.4.3. ORDER VALIDITIES

Order validity determines an event or period after which an order is expired and is dropped from GPW WATS order books.

An order can be valid in the point in time when it is received by the system (IOC, FOK), until the precise timestamp current day (GTT), until the end of the current day (Day), until the end of specified day in the future (GTD), until it is cancelled (GTC), or only during the Trading Phase (VFA, VFC).

Exchange Members may submit orders with the following validity attributes in the New Order Single (D) message via FIX Order Gateway:

- Day (DAY),
- Good Till Cancel (GTC),
- Good Till Date (GTD),
- Good Till Time (GTT),
- Immediate Or Cancel (IOC),
- Fill Or Kill (FOK),
- Valid For Auction (VFA),
- Valid For Closing (VFC).

Order validity is determined by TimeInForce (59) tag in conjunction with ExpireDate (432) or ExpireTime (126) tags.

Not all validity attributes may be used with each Order Type (40) throughout the various market phases.

4.4.3.1. Day (DAY)

Day order is valid only until the end of the current trading day. Unexecuted day orders are expired by Trading System at the end of the trading day during post-session processing (in Trading Closed market phase).

Day validity attribute may be used with each order type, except for market order and market to limit.

Relevant FIX tags:

TimeInForce (59) = 0 (Day)

4.4.3.2. Good Till Cancel (GTC)

Good till cancel order is valid until it is fully executed or canceled by the submitter or market operations (whatever comes first).

Good till cancel validity attribute may be used with each order type, except for market order and market to limit.

Relevant FIX tags:

TimeInForce (59) = 1 (Good Till Cancel)

4.4.3.3. Good Till Date (GTD)

Good Till Date order is valid until the end of the specified day.

If the expiry date specified for an order is not a trading day, the order becomes expired at the end of the previous trading day.

Good till date validity attribute may be used with each order type, except for Market Order and Market To Limit.

ExpireDate attribute is in a date format (without time component) e. g. “20220102” and can only be used in conjunction with (59) = 6.

It is possible to modify orders with GTD validity type. Modification of the date is possible from the current date to the limit for GTD (otherwise the modification is rejected). Modified orders do not change priority.

Relevant FIX tags:

TimeInForce (59) = 6 (Good Till Date)
ExpireDate (432) = Expiration Date

4.4.3.4. Good Till Time (GTT)

Good till time order is valid until the specified time of the current trading day.

If the expiry time specified for an order is later than the end of the last market phase, the order will become expired at the end of the current trading day.

Good till time validity attribute cannot be used with the future Expire Date (432).

Good till time validity attributes may be used with each order type, except for market order and market to limit.

Relevant FIX tags:

TimeInForce (59) = 6 (Good Till Date)
ExpireTime (126) = Expiration Time (UTC timestamp with current date)

4.4.3.5. Immediate Or Cancel (IOC)

Immediate or cancel order must be executed immediately, in full or partially, upon order entry. IOC orders may be matched against one or more opposing orders with the price limits within trade price collars. Unexecuted part of IOC order is automatically canceled by Trading System and Execution Report informing about cancellation is marked with ExecTypeReason (2431)=102 (Cancel IOC/FOK order).

Immediate or cancel orders may be submitted during Continuous Trading market phase only.

Immediate or cancel validity attribute may be used with each order type, except for stop limit, stop loss and iceberg.

Relevant FIX tags:

TimeInForce (59) = 3 (immediate or cancel)
ExecTypeReason(2431)=102(Cancel IOC/FOK order)

4.4.3.6. Fill Or Kill (FOK)

Fill or kill order has to be executed in full, immediately upon order entry. FOK orders may be matched against one or more opposing orders with the price limits within trade price collars. Unexecuted FOK order is automatically canceled by Trading System and Execution Report informing about cancellation is marked with ExecTypeReason(2431)=102(Cancel IOC/FOK order).

Fill or kill orders may be submitted during Continuous Trading market phase only.

Fill or kill validity attribute may be used with each order type, except for stop limit, stop loss and iceberg.

Relevant FIX tags:

TimeInForce (59) = 4 (Fill Or Kill)
ExecTypeReason(2431)=102(Cancel IOC/FOK order)

4.4.3.7. Valid For Auction (VFA)

Valid for auction order is activated and inserted into the order book at the beginning of the next Auction market phase (whether scheduled or unscheduled). VFA order submitted in Continuous Trading market phase, is hidden until activation (i.e. it is not published via Market Data) and cannot interact with the order book. Once activated, VFA order is valid until the end of the Auction and becomes expired by Trading System immediately after the Uncrossing (if left unexecuted). VFA orders are valid for the current trading day only. Activated VFA orders retain their original priority timestamp from the point of submission.

Valid For Auction validity attribute may be used with each Order Type, except for Stop Limit, Stop Loss and Iceberg.

Relevant FIX tags:

TimeInForce (59) = 2 (At The Opening)

4.4.3.8. Valid For Closing (VFC)

Valid For Closing order is activated and only inserted into the order book at the beginning of the Closing Auction market phase, when Call sub-phase begins. VFC order submitted in Continuous Trading market phase or Auction other than Closing, is hidden until activation (i.e. it is not published via Market Data) and cannot interact with the order book. Once activated, VFC order is valid until the end of the Closing Auction and becomes expired by Trading System immediately after the Uncrossing (if left unexecuted). VFC orders are valid for the current trading day only. Activated VFC orders retain their original priority timestamp from the point of submission.

Valid For Closing validity attribute may be used with each Order Type, except for Stop Limit, Stop Loss and Iceberg.

Relevant FIX tags:

TimeInForce (59) = 7 (At The Closing)

4.4.4. ORDER CAPACITY

OrderCapacity (528) defines the capacity of the firm that is submitting an order. These are the possible values:

- P - Principal mapped as DEAL (Dealing on own account),
- A - Agency mapped as AOTC (Any other capacity),
- R - Riskless Principal mapped as MTCH (Matched principal).

It is required to set the correct capacity field value by Exchange Member.

4.4.5. MARKET MAKING

The liquidity provision activity is provided by the regular market makers under the terms of market making scheme agreement concluded with the Market and by the Issuer Market Maker agreements concluded with the Issuer of the instrument which is being traded. Regular Market Makers who submit their orders must set optional field OrderRestrictions (529) = 5 (Acting as Market Maker or Specialist).

According to MiFID2/MiFIR Regulations all orders submitted to the Market as part of the liquidity provision activity must be assigned with the "Liquidity Provision Activity" flag.

The liquidity provision activity can be performed by members of the Market and non-member firms which is differentiated by setting the OrderCapacity (528) value to "P" (Principal) and "A" (Agency) respectively.

The table below provides a summary of the different combinations for the liquidity provision activity with FIX fields:

FIX Field(tag)	Regular market maker		Issuer market maker	
	Member	Non-member	Member	Non-member
OrderCapacity(528)	P = Principal	A = Agency	P = Principal	A = Agency
OrderAttributeType(2594)	2 = Liquidity provision activity order	2 = Liquidity provision activity order	2 = Liquidity provision activity order	2 = Liquidity provision activity order
OrderAttributeValue(2595)	Y	Y	Y	Y

FIX Field(tag)	Regular market maker		Issuer market maker	
	Member	Non-member	Member	Non-member
PartyID - client short code (448)	NOT SET	SET	NOT SET	SET
OrderRestrictions(529)	5 = Acting as Market Maker or Specialist in the security	5 = Acting as Market Maker or Specialist in the security	NOT SET	NOT SET

4.4.6. ORDER STATUSES

During the order lifecycle FIX Order Gateway sends Execution Report (8) each time something happens to an order. There are two attributes, one OrdStatus (39) which reflect current order status and another ExecType (150) which refers to an event which has triggered Execution Report generation.

Order status can be unchanged across many Execution Reports, once ExecType reflects the last action performed on an order.

Information about the events on an order and its status is conveyed in FIX Execution Report (8) messages in the following tags:

4.4.6.1. OrdStatus (39)

Exchange Member may receive from the FIX Order Gateway the following statuses for orders which have been submitted:

- New (0) - when an order is accepted into the order book until there is any execution on it(trade) or until an order is cancelled or expired. In case aggressive order was partially or fully filled there is no New status assigned to order. To help Exchange Member identify such a situation, new field ExecTypeReason(2431) = 110 (First trade on aggressive order) had been added to Execution Report,
- Partially Filled (1) - when there was at least one execution on an order until the order is fully filled (there is no remaining quantity on it) or until the order is cancelled or expired,
- Filled (2) - when an order is fully executed (there is no remaining quantity on an order),
- Canceled (4) - when an order is canceled by Exchange Member, Trading System itself (including situation when an incoming IOC/FOK order cannot be fully executed upon arrival) or by the Market Operations. Additional information about the reason for unsolicited cancellation cancellation is provided in the ExecTypeReason (2431) tag,
- Rejected (8) - when an order is rejected by Trading System upon entry due to various business validations,
- Expired (C) - when an order is automatically removed from the order book due to its validity condition specified in TimeInForce (59) tag (including removal of unexecuted part of VFA/VFC order after the Auction).

4.4.6.2. ExecType (150)

Exchange Member may receive the following types of Execution Types from the FIX Order Gateway:

- New (0) - returned as a response to New Order Single when an order was submitted into book and not executed,

- Canceled (4) - returned as a response to Order Cancel(35=F) when an order was successfully cancelled by Client, Trading System itself (including situation when an incoming IOC/FOK order cannot be fully executed upon arrival) or by market operations. Additional information about the reason for unsolicited cancellation is provided in the ExecTypeReason (2431) tag,
- Replaced (5) - returned as a response to successful Order Cancel Replace(35=G) request,
- Rejected (8) - returned as a response to New Order Single (35=D) or Order Cancel Replace(35=G) or Order Cancel (35=F) when there was an issue with message processing.,
- Expired (C) - when an order is automatically removed from the order book due to its validity condition specified in TimeInForce (59) tag (including removal of unexecuted part of VFA/VFC order after the Auction),
- Restated (D) - returned when a new portion of an Iceberg order is disclosed to the market or at the beginning of a new trading day for orders submitted in the past but still valid (not cancelled or fully filled orders with validities such as GTC, GTD),
- Trade (F) - returned as a response to Partial or Full trade on order,
- Trade Cancel (H),
- Triggered (L) - returned when order with additional activation condition is activated such as valid for auction / valid for closing orders (TimeInForce = 2 or 7) and stop loss / stop limit orders (OrdType = 3 or 4).

4.4.6.3. ExecTypeReason (2431)

ExecTypeReason (2431) provides additional information in the following situations:

- Order Restatement (100 - GT order restatement),
- Unsolicited Order Cancellation (ExecType = 4),
- Partial or full trade (ExecType = F).

ExecTypeReason (2431) can take following values:

- 100 - GT order restatement,
- 101 - Iceberg order refill,
- 102 - Cancel IOC/FOK order,
- 103 - Cancel by STP,
- 104 - Cancel by Market Operations,
- 105 - Cancel on Trading Halt,
- 106 - Cancel on Disconnect,
- 107 - Cancel by Corporate Action,
- 108 – Cancel by Mass Cancel,
- 110 - First trade on aggressive order.

4.4.6.4. Triggered (1823)

Triggered is an optional tag which is provided in Execution Report (8) messages, only relating to valid for auction / valid for closing orders (TimeInForce = 2 or 7) and stop loss / stop limit orders (OrdType = 3 or 4), and informs whether the order is already activated or not (at the point of Execution Report dispatch).

Triggered tag can take the following values:

- not triggered (0),
- triggered (1),

- stop order triggered (2).

4.4.7. CANCELLATION

Each submitted by New Order Add request active order can be terminated by OrderCancel (35=F) request.

Only orders with Status (39) New and Partially Filled are active and can be cancelled. If an order was already Filled, Cancelled, Expired or Rejected it cannot be cancelled since GPW WATS treats such order as non-existent. An order can be referenced by one of two attributes:

OrderID (37) which is system unique ID which remains unchanged during the order lifecycle.

origClOrdID which is Client unique order reference. Since it can be changed during the order lifecycle when an order is modified by Participant, it should always be last clOrdID.

If Exchange Member provides both orderID and origClOrdID in the OrderCancel(35=F) request, Trading System will process the OrderID only and ignore origClOrdID.

4.4.8. MODIFICATION

Each active order submitted by NewOrderSingle(D) request can be adjusted by OrderCancelReplace (35=G) request.

Only orders with Status (39) New and Partially Filled are active and can be modified. If an order was already Filled, Cancelled, Expired or Rejected it cannot be cancelled since GPW WATS treats such order as non-existent. An order can be referenced by one of two attributes:

- OrderID (37) which is a system unique ID remains unchanged during the order lifecycle,
- origClOrdID which is Client unique order reference. Since it can be changed during the order lifecycle when an order is modified by Participant, it should always be last clOrdID.

If Exchange Member provides both orderID and origClOrdID in the OrderCancelReplace (35=G), Trading System will process the OrderID only and ignore origClOrdID.

Exchange Members are allowed to modify the following order characteristics:

- Total quantity,
- Initial displayed quantity (in case of Iceberg Orders only),
- Price,
- Trigger price (in case of Stop Orders only),
- ExpireDate(432)(in case of TimeinForce (59)=6 (Good Till Date),
- Short Codes.

In case of order modification all Parties MiFID Fields shall be provided and are subject of validation identical as for New Order Single.

Validation checks of Parties fields being modified are referenced to Order Capacity value in the original new order message.

4.4.9. ACCOUNT STRUCTURE

There are two FIX conditional fields that can be used by Trading Member for clearing purposes: AccountType (581) and Account (1).

There are two possible types of accounts (i.e. AccountType (581) field):

1. Customer - account is carried on customer side of the books,
2. House – house trader.

4.4.10. ORDER BOOK RESTATEMENT

At the beginning of each trading day, Exchange Members receive Execution Report messages from Gateway informing them about orders that were submitted in the past but still valid (not cancelled, not fully filled orders with validities such as GTC, GTD, not expired).

More information can be found in the document D01.1 GPW WATS Trading System, chapter Trading Schedules.

Restated orders remain almost all attribute values from previous session such as OrderID (11), ClOrdID (37), OrdStatus(39), LeavesQty (151), CumQty (14). One exception is publicOrderID which is newly assigned returned in MDEntryID (278) in Execution Report.

Restated orders are marked in ExecutionReport with ExecType=D.

Relevant FIX tags:

ExecType (150) = D (Restated)

4.4.11. SELF-TRADE PREVENTION

The Self-Trade Prevention (STP) functionality enables Exchange Members to avoid an unintentional execution of orders at the submitted by the same Exchange Member. Exchange Member shall provide the STP ID value at the order entry level to enable the trading platform to activate the STP functionality. There is no need for any additional configuration for Exchange Members in the trading system. In FIX protocol in the New Order Single (D) and in the Execution Report (8) messages, an optional field SelfMatchPreventionID carrying the STP ID key will be used. The exchange will not dictate or validate the value of the STP ID as long as it is compliant with the format of the field. Allocation of the STP ID values is at the sole discretion of Exchange Member.

Self-Trade Prevention mechanism works in scope of assigned SelfMatchPreventionID value.

In case incoming aggressive order could be executed with already present (passive) order System cancels passive order and Exchange Member receives the Execution Report with ExecTypeReason (2431) set to 103 - Cancel by STP and OrdStatus(39)=4.

Relevant FIX tag

SelfMatchPreventionID (2362)- numeric values

ExecTypeReason (2431) = 103 - Cancel by STP

4.4.12. UNSOLICITED CANCELLATIONS

Orders can be cancelled by the Client or unsolicited by Market Operations or by System.

When unsolicited cancellation takes place, Exchange Member receives the Execution Report with ExecTypeReason (2431) set to one of following values:

- 102 - Cancel IOC/FOK order,

- 103 - Cancel by STP,
- 104 - Cancel by Market Operations,
- 105 - Cancel on Trading Halt,
- 106 - Cancel on Connection Loss,
- 107 - Cancel by Corporate Action,
- 108 – Cancel by Mass Cancel.

4.5. TRADE CAPTURE REPORTS

The Trade Capture Reports (AE) are messages sent through FIX Order Gateway and through Post Trade Gateway. The Trade Capture Reports sent through FIX Order Entry Gateway are used to report Block Trades and Cross Trades to Market Operator for regulatory and post-trade transparency reasons. The Trade Capture Reports sent through FIX Post-trading Gateway are used for downstream processing to relay confirmed trades to parties involved in clearing and settling trades. The Trade Capture Reports sent through FIX Post-trading Gateway are not within the scope of this document.

4.5.1. TRADE CAPTURE REPORT IDENTIFIERS

4.5.1.1. Trade Report ID

The TradeReportID (tag 571) is assigned by the submitter of the Trade Capture Report message and is used as a pure message identifier. In the one-sided TCR in the applied model "one-party for pass through" the initiator and the counterparty assign their own TradeReportID independently of each other. TradeReportID must be unique across the trading day for each submitter separately. TradeReportID is assigned separately by Market Operator in outbound messages where the Trade Capture Report is generated e.g. as the notification sent to the counterparty or as a trade confirmation sent to the parties to the trade.

4.5.1.2. Secondary Trade Report ID

The SecondaryTradeReportID (818) is a unique identifier in a numeric format of a first leg to the trade (Trade Capture Report with tradeReportTransType: New).

GPW WATS provides newly generated SecondaryTradeReportID in TradeCaptureReportAck returned to Exchange Member each time a new first leg to the trade is submitted.

Gateway provides the same SecondaryTradeReportID in Trade Capture Report to notify counterparty of first leg to trade(tradeReportType: Alleged).

Counterparty uses the same SecondaryTradeReportID to accept the trade (initiated by first leg). Subsequently Gateway provides the same SecondaryTradeReportID in Trade Capture Report to notify both parties (initiator and counterparty) that trade has been finalized.

SecondaryTradeReportID can also be used as a unique reference to cancel a first leg to the trade (tradeReportTransType: Cancel).

4.5.1.3. Trade Report Reference ID

The TradeReportRefID (572) is assigned by the submitter when the Trade Capture Report is referred to the previous message. This field is filled in when e.g. Market Operator generates Trade Capture Report as the trade confirmation to the parties or when the Trade Capture Report is cancelled by the submitter.

4.5.2. BLOCK MARKET MODEL

Block trading is a market model in which trades are concluded bilaterally in the process of the negotiations between the counterparties (beneficiary clients). The beneficiary clients may belong to one or a few Exchange Members. The negotiations usually take place beyond Trading System infrastructure. Block trading is concluded when the respective trade entry is recorded in Trading System based on the Trade Capture Report message/messages submitted by Exchange Members.

Block Trade facility enables Exchange Members to report pre-arranged large trades through the Market Operator Trading System.

The value of Block Trades must not be lower than the value e.g. Large In Scale value. The price of the Block Trade must be within thresholds set by the exchange. Price thresholds can be set on different levels depending on the session phase of the same instrument in the Central Limit Order Book.

Block Trade features:

- MiFID 2 tick size regime does not apply to Block Trades,
- Both parties to the Block Trade must be Exchange Member,
- Block Trades are not pre-transparent thus Trade Capture Reports are not disseminated in the market data stream,
- Block Trades are post-trade transparent thus confirmed and matched Block Trades are immediately disseminated in the Market Data stream,
- Block Trades do not update the Last Trade Price,
- Block Trades are not taken into account when calculating the opening price or the closing price.

4.5.2.1. Dual-Sided Trade Capture Report

Dual-sided Trade Capture Report is applied when both beneficiary parties to the trade belong to the reporting firm (Exchange Member). In the Trade Capture Report message the value of the field TradeHandlingInstr (1123) is set at 1 = Two-party Report. The Trade Capture Report is accepted by Gateway by sending back Trade Capture Report Acknowledgment (TCR AR) with the field TrdRptStatus (939) = 0 (Accepted) followed by the Trade Capture Report which confirms the trade with the field ExectType = Trade. If Trade Capture Report is rejected, then Exchange Member receives the Trade Capture Report Acknowledgment with TrdRptStatus (939) = 1 (Rejected) with the respective information about the reason for rejection provided in the field TradeReportRejectReason (751).

The table below provides dual-sided Trade Capture Report messages flow covering different scenarios which include:

- Exchange Member's TCR rejection by Gateway,
- Exchange Member's TCR accepted by Gateway.

Operation	IN / OUT	Message	TradeReportID (TradeReportRefID)	TradeID	TradeReport-TransType	TradeReport-Type	TrdType	TradeHandling-Instr	ExecTYpe	TrdRptStatus
			Fix Tag 571 (Fix Tag 572)	Fix Tag 1003	Fix Tag 487	Fix Tag 856	Fix Tag 828	Fix Tag 1123	Fix Tag 150	FIX Tag 939
Exchange Member submits dual-sided TCR	IN	TCR (AE)	Exchange Member assigned	n/a	0 = New	0 = Submit	38 = Block Trade	1 = Two-party report	n/a	n/a
Dual-sided TCR accepted by TV	OUT	TCR Ack (AR)	Exchange Member assigned (echoed back)	n/a	0 = New	0 = Submit	38 = Block Trade	1 = Two-party report	n/a	0 = Accepted
Dual-sided TCR rejected by TV	OUT	TCR Ack (AR)	Exchange Member assigned (echoed back)	n/a	0 = New	0 = Submit	38 = Block Trade	1 = Two-party report	n/a	1 = Rejected
Trade	OUT	TCR (AE)	TV assigned	TV assigned	2 = Replace	0 = Submit	38 = Block Trade	1 = Two-party report	F = Trade	n/a

4.5.2.2. One-Sided Trade Capture Report

One-sided Trade Capture Report is applied when the beneficiary parties to the trade belong to different Exchange Members. The applied one-sided model is the "one-party report for pass-through". The initiator submits a report (first leg to the trade) which is accepted by the counterparty to the trade. In the Trade Capture Report message the value of the field TradeHandlingInstr (1123) shall be set at 3 = one-party report for pass-through. Below is shown the basic TCR flow enabling Exchange Members to conclude the block trade.

1. The initiator submits the first leg to the trade by sending Trade Capture Report with the values:
    - TradeReportTransType (487) = 0 (New),
    - TradeReportType (856) = 0 (Submit).
2. The first leg is accepted by WATS System by sending Trade Capture Report Ack (TCR AR) with the TrdRptStatus (939) = 0 (Accepted),
3. WATS System forwards the Trade Capture Report to the counterparty by sending Trade Capture Report (notification) with the values:
    - TradeReportTransType (487) = 0 (New),
    - TradeReportType (856) = 1 (Alleged).
4. The counterparty accepts the first leg by sending Trade Capture Report with the values:
    - TradeReportTransType (487) = 2 (Replace),
    - TradeReportType (856) = 2 (Accept).
5. GPW WATS sends the Trade Capture Reports informing the initiator and the counterparty about the concluded trade with the values:
    - TradeHandlingInstr (1123) = 0 (Trade Confirmation),
    - Exec Type (150)=F.

The table below provides one-sided Trade Capture Report messages flow covering different scenarios.

Operation	IN / OUT	Message	TradeReportID (TradeReportRefID)	TradeID	TradeReport-TransType	TradeReportType	TrdType	TradeHandling-Instr	ExecTYpe	TrdRptStatus for TCR Ack only
			Fix Tag 571 (Fix Tag 572)	Fix Tag 1003	Fix Tag 487	Fix Tag 856	Fix Tag 828	Fix Tag 1123	Fix Tag 150	FIX Tag 939
Exchange Member first leg accepted or rejected by TV										
Exchange Member submits first leg	IN	TCR (AE)	Exchange Member assigned	n/a	0 = New	0 = Submit	38 = Block Trade	3 = One-party	n/a	n/a
First leg rejected by TV	OUT	TCR Ack (AR)	Exchange Member assigned (echoed back)	n/a	0 = New	0 = Submit	38 = Block Trade	3 = One-party	n/a	1 = Rejected
First leg accepted by TV	OUT	TCR Ack (AR)	Exchange Member assigned (echoed back)	n/a	0 = New	0 = Submit	38 = Block Trade	3 = One-party	n/a	0 = Accepted
TV forwards TCR to C-pty (notification)	OUT	TCR (AE)	TV assigned	n/a	0 = New	1 = Alleged	38 = Block Trade	3 = One-party	n/a	n/a
Exchange Member cancels first leg before accepting by C-pty	IN	TCR (AE)	Exchange Member assigned (Exchange Member's previous)	n/a	1 = Cancel	6 = Trade Report Cancel	38 = Block Trade	3 = One-party	n/a	n/a
First leg cancellation rejected by TV	OUT	TCR Ack (AR)	Exchange Member assigned (Exchange Member's previous)	n/a	1 = Cancel	6 = Trade Report Cancel	38 = Block Trade	3 = One-party	n/a	1 = Rejected
First leg cancellation accepted by TV	OUT	TCR Ack (AR)	Exchange Member assigned (Exchange Member's previous)	n/a	1 = Cancel	6 = Trade Report Cancel	38 = Block Trade	3 = One-party	n/a	0 = Accepted
TV forwards cancelled first leg TCR to C-pty	OUT	TCR (AE)	TV assigned (571) TV previous (572)	n/a	1 = Cancel	6 = Trade Report Cancel	38 = Block Trade	3 = One-party	n/a	n/a
Counterparty declines Exchange Member's first leg										

Operation	IN / OUT	Message	TradeReportID (TradeReportRefID)	TradeID	TradeReport-TransType	TradeReportType	TrdType	TradeHandling-Instr	ExecTYpe	TrdRptStatus for TCR Ack only
C-pty declines Exchange Member first leg	IN	TCR (AE)	C-pty assigned	n/a	2 = Replace	3 = Decline	38 = Block Trade	3 = One-party	n/a	n/a
C-pty decline is rejected by TV	OUT	TCR Ack (AR)	C-pty assigned(echoed back)	n/a	2 = Replace	3 = Decline	38 = Block Trade	3 = One-party	n/a	1 = Rejected
C-pty decline accepted by TV	OUT	TCR Ack (AR)	C-pty assigned(echoed back)	n/a	2 = Replace	3 = Decline	38 = Block Trade	3 = One-party	n/a	0 = Accepted
TV submits TCR to Exchange Member (informing about C-pty decline)	OUT	TCR (AE)	TV assigned	n/a	2 = Replace	3 = Decline	38 = Block Trade	3 = One-party	n/a	n/a
Counterparty accepts Exchange Member's first leg										
C-pty accepts Exchange Member first leg	IN	TCR (AE)	C-pty assigned	n/a	2 = Replace	2 = Accept	38 = Block Trade	3 = One-party	n/a	n/a
C-pty accept is rejected by TV	OUT	TCR Ack (AR)	C-pty assigned(echoed back)	n/a	2 = Replace	2 = Accept	38 = Block Trade	3 = One-party	n/a	1 = Rejected
C-pty accept is accepted by TV	OUT	TCR Ack (AR)	C-pty assigned(echoed back)	n/a	2 = Replace	2 = Accept	38 = Block Trade	3 = One-party	n/a	0 = Accepted
TV match info to Exchange Member	OUT	TCR (AE)	TV assigned Ref to first leg	TV assigned	2 = Replace	0 = Submit	38 = Block Trade	0 = Trade Confirm	F = Trade	n/a
TV match info to C-pty	OUT	TCR (AE)	TV assigned Ref to second leg	TV assigned	2 = Replace	0 = Submit	38 = Block Trade	0 = Trade Confirm	F = Trade	n/a

4.5.3. CROSS TRADES

Cross Trades are privately negotiated trades brought onto Market Operator. Cross Trades must comply with Market Operator rules in terms of price and value. Cross Trades are reported by using dual-sided Trade Capture Report. Cross Trades in the Trade Capture Report messages are defined in the field TrdType (tag 828) with the value set at "22" (Privately negotiated trade).

Cross trades can only be submitted in the Continuous Trading phase (variable price and fixed price), i.e. in:

- Continuous price and time (ContinuousPriceTime),
- Continuous time at reference price (ContinuousPriceTime),
- Trade at last (ContinuousLastAuctionTime).

Cross trades do not change Last Trade Price for an instrument with the same ISIN code in the central order book.

Kinematics of Cross transaction:

- Transmission of a TCR Cross (AE) to System with a TradeReportType field (856) with the value 0 = Submit.
- For a valid TCR Cross, System sends a TCR Ack with an Accepted status (TrdRptStatus 939 = 0 (Accepted))
- For an invalid TCR Cross, System sends a TCR Ack with status Rejected (TrdRptStatus 939 = 1 (Rejected)) In subsequent WATS system versions for an accepted TCR Cross, System sends a TCR (AE) with field Exec Type(150)=F.

Relevant FIX tag

TrdType (828) = 22

NoSides (552) = 2

SettlDate (64) should not be set for CROSS TCRs sent to FIX gateway.

4.6. QUOTE HANDLING

Market makers can send quotes into a market using a Mass Quote(i) message. For a given instrument, it is possible to send one quotation. The MassQuote message enables sending of up to 30 quotations.

The trading platform sends a confirmation of the accepted and rejected quotations by transmitting a MassQuoteAck(b) message.

Quotations rules:

- Two-sided quotes are allowed,
- Only one market maker quotation is allowed per instrument,
- Mass Quote Ack (b) message includes all submitted quotations along with their acceptance status (e.g. Accepted, Rejected),
- Modification of submitted quotations involves placing another Mass Quote message with changed values for price or quantity,

- Cancellation of submitted quotations involves placing another Mass Quote message with the quantity values set to 0.

Mass Quote messages can be used:

- On CLOB(order driven) market to provide additional liquidity on Instruments
- On Hybrid market where Market Makers quotes presence is vital to conclude trades, act as thresholds mechanism to manage volatility as they restrict sudden price movement while no transaction can be made beyond the limits set by these quotes. In case of possible execution of an order against Market Maker's quote, Order Gateway sends to Market Maker RequestForExecution (UE) message. Upon receipt of RFE(UE) Market Maker can confirm, modify or cancel current quotes or do nothing to proceed with execution. Market Maker can:
  - Change phase of Instrument from Hybrid to HybridBuyOnly by sending MarketMakerCommand [UB] with field MarketMakerCommandAction (20009) = CHANGE_TO_HYBRID_BUY_ONLY and then use one-sided quotes.,
  - Change phase of Instrument from HybridBuyOnly to Hybrid by providing two-sided quote,
  - Suspend Hybrid Instrument by sending MarketMakerCommand [UB] with field MarketMakerCommandAction (20009) = CHANGE_TO_KNOCK_OUT,
  - Unsuspend Hybrid Instrument by sending MarketMakerCommand[UB] with field MarketMakerCommandAction(20009)= REVOKE_KNOCK_OUT.

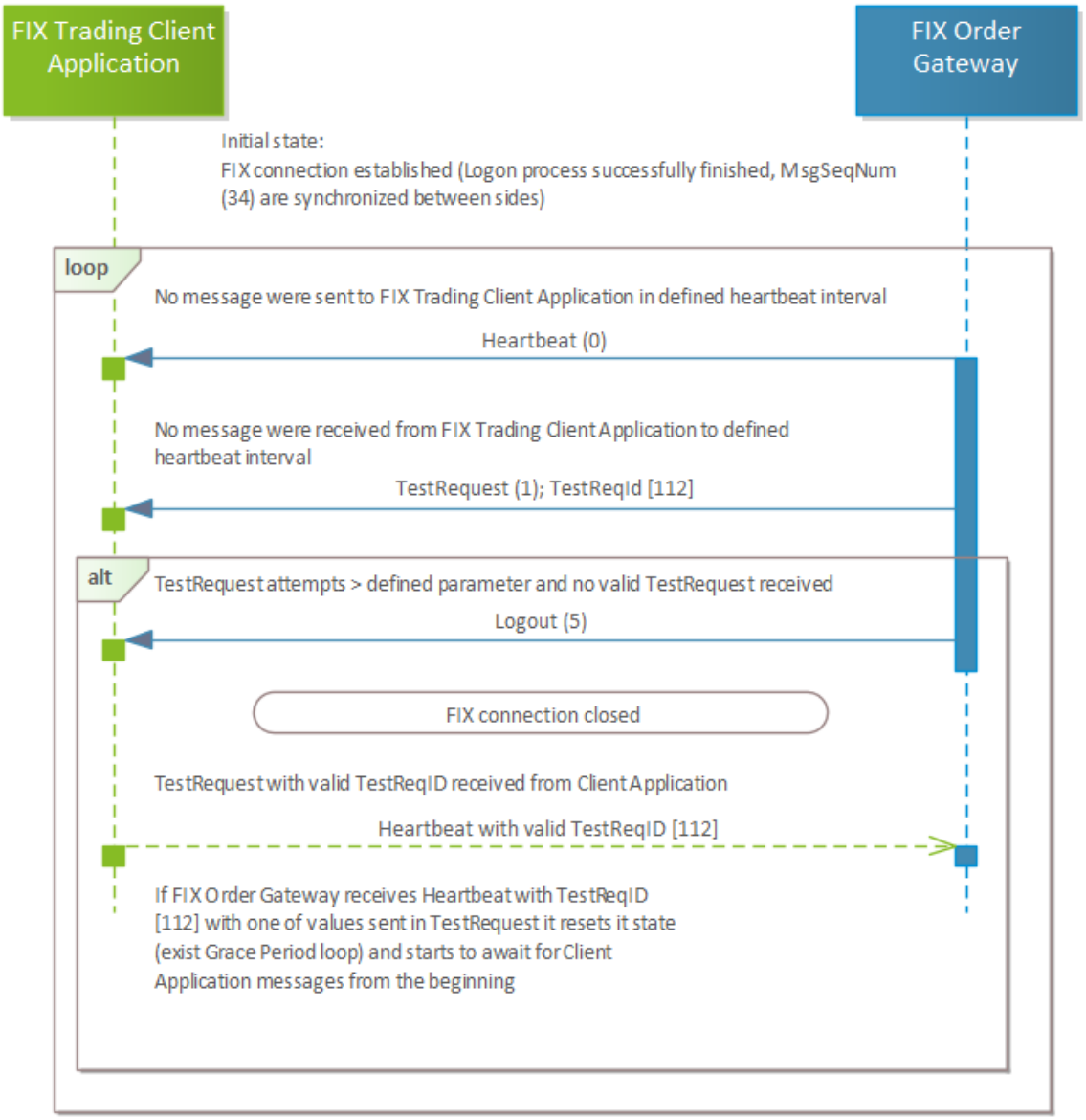
4.7. Cancel On Disconnect

Cancel on disconnect is a mechanism designed to protect Exchange Member's from potential losses which can happen when they unintentionally lost control over their orders. CoD service is setup on connection level not on Participant level. If the user has more than one connection, then the mechanism can be activated on each of these connections or only on chosen connections. The GPW is responsible for the configuration of the service and performs it at Exchange Member request. When it is switched on and FIX Gateway discovers member's unintentional unresponsiveness on particular connection it cancels all open orders submitted on this connection and field ExecInst=o (cancel on connection loss). If ExecInst field is not set on Order level (in submitted NewOrderSingle(35=D)) COD mechanism won't cancel such order. . As soon as client application reestablishes FIX connection(after unintentional disconnector or control lost) with FIX Gateway it receives Execution Reports with information of unsolicited cancellations (ExecType (150)=4 - Canceled, OrdStatus (39) = 4 - Canceled, ExecTypeReason (2431) = 106 - Cancel on Disconnect.Gateway assess that Client Application lost control in two situations:

- Connection with the Client Application was dropped without Logout procedure, but not from GPW WATS initiative. When Client Application performed Logout successfully or GPW WATS failover Cancel on Disconnect is not activated,
- Physical connection is active but Participant does not send Heartbeat messages for a period of 30 seconds (so-called grace period).

When Gateway detects no message (including heartbeat) from Client Application for a period longer than heartbeat interval it sends Test Request and starts Grace Period process (figure below). GPW WATS sends Test Requests (1) up to 3 times once every 30 seconds and if Client Application does not respond with Heartbeat (0) message Gateway sends Logout and terminates FIX connection. When responding to Test message, Client Application should include TestReqID (112) field with value received in the last received Test Request from GPW WATS.

Figure 3. – Grace period mechanism



4.8. Mass Cancellation

Mass Cancel functionality allows Exchange Member to cancel all active orders according to the selection criteria they have specified. In the case of a previously partially executed order, only the remaining unexecuted part of the order is cancelled. This means that Mass Cancel doesn't affect the already executed volume of the order. Mass Cancel functionality doesn't work for BLOCKS and CROSS markets..

The Mass Cancel functionality does not include the cancellation of unmatched Trade Capture Reports.

This functionality can only be used by Exchange Member in respect of orders that they have sent themselves. Sending a Mass Cancel message is not possible via the DEA (Direct Electronic Access). Exchange Member must indicate the Executing Trader performing the operation (PartyRole= Executing Trader) and provide shortcode in PartyID field.

Exchange Member performs Mass Cancel by sending the OrderMassCancelRequest (q) message. The OrderMassCancelRequest (q) message allows the cancellation of all active orders placed on all

connections (ConnectionID) of a Participant. OrderMassCancelRequest message contains fields that define the selection criteria for cancelled orders.

Selection Criteria

Scope of cancellation can refer to all active orders or can be limited to particular Instrument or Market Segment by setting mandatory MassCancelRequesrType(530) field:

- 1 = Cancel orders for security,
- 7 = Cancel all orders,
- 9 = Cancel orders for a Market Segment.

In case member chooses to limit orders to Instrument or Market Segment he also needs to set SecurityID(48) and SecurityIDSource(22) or MarketSegmentID(1300) respectively.

In addition,Order Mass Cancel Request can include one of a number of narrowing criteria contained in the optinal TargetParties sectionby filling it for one of following TargetpatyRoles:

- 35 = Liquidity Provider,
- 54 = Sender Location,
- 66 = Market Maker,
- 3 = Client ID,
- 12 = Executing Trader,
- 122 = Investment Decision Maker.

with corresponding TargetPartyIDSource and TargetPartyID..

It is allowed to select only one narrowing criterion from the TargetParties group, which automatically means that the value of the NoTargetPartyIDs field must always be equal to 1.

OrderMassCancelReport is sent via Drop Copy service.

Message Identifiers

The Participant assigns to each message OrderMassCancelRequest with the unique identification ClOrdID (11) which is echoed back in the message OrderMassCancelReport (R). It is recommended that the uniqueness is kept across Participants(all connections) all FIX messages applied to orders' management (NewOrderSingle, OrderCancel/Replace, OrderCancel).

Trading System in the message OrderMassCancelReport provides identifier in the field MassActionReportID (FIX tag 1369).

OrderMassCancelReport

In response to OrderMassCancelRequest (Q) Trading System generates outgoing message OrderMassCancelReport (R).

The field MassCancelResponse (531) provides information about the result of the Mass Cancellation processing.

If the OrderMassCancelRequest is rejected the MassCancelResponse message provides the reject reason in the field Mass CancelRejectReason (532).

Execution Report

If the OrderMassCancelRequest is accepted then apart from the MassCancelResponse message Fix Gateway issues the Execution Reports for each successfully canceled order with ExecTypeReason=108(Cancel by Mass Cancel).

4.9. Reject Messages

After each message is received GPW WATS performs a number of checks in a specific order. This is crucial to report the most basic issues at first, and limit processing steps for broken messages. Depends on detected issue FIX Gateway can:

- Immediately close connection without any Logout

  This happens when BeginString (8), SenderCompID (49), TargetCompID (56) in Logon message is invalid or system receives Logon for SenderCompID (49) with an already open FIX connection.

- Ignore message without any response (system treats such message as it was never received). System treats message as garbled when:
  - BeginString (8), BodyLength (9), MsgType (35) are not the first three tags in received message, Checksum (10) is not the last tag in message,
  - At least one of tags: BeginString (8), BodyLength (9), Checksum (10) has invalid value.

  In such a case (Gateway ignores incoming message) when Gateway receives next valid message with incremented MsgSeqNum (34) it detects a gap and requests Exchange Member to resend missing messages. by sending ResendRequest (2). To avoid resent infinity loop after 3 attempts if FIX Gateway still detects gap in MsgSeqNum (34) it sends Logout message and close the connection.

- Return session level reject (35=3) message.

  Session level reject message consists of RefSeqNum (45) with MsgSeqNum of rejected message, SessionRejectReason (373) with code which identifies detailed reject reason.

  Reject (35=3) message is returned by GPW WATS in case:
  - incoming message type is invalid (tag 35) → SessionRejectReason (373) = 11,
  - message does not contain mandatory, properly formatted field → SessionRejectReason (373) = 1,
  - message contain invalid, undefined or undefined for this message type tag → SessionRejectReason (373) = 0,3,2,
  - message contains tag without value, value is incorrect or provided value format is incorrect → SessionRejectReason (373) = 4, 5, 6,
  - message contains tag which appears more than once → SessionRejectReason (373) = 13,
  - message contains tags in repeating group in invalid order or number of groups is invalid → SessionRejectReason (373) = 14, 15, 16,
  - SendingTime (52) in message is out of acceptable tolerance (either from the future or in the past) → SessionRejectReason (373) = 10,
  - SenderCompID (49) for message different than Logon is different than expected → SessionRejectReason (373) = 9,

- Return reject message on Business level reject (35 = j),

If validation on business level detects an error condition then a rejection is returned.

- Return Execution Report with OrdStatus=8(Rejected) and OrdRejReason filled with rejection code.

All rejection codes are documented in Rejection codes document.

5. CONNECTIVITY

This chapter provides information regarding: logon authentication, failover and recovery.

5.1. SESSION IDENTIFICATION

Establishing a FIX session with GPW WATS requires logging into System. To do this, it is necessary to have a token associated with a unique SenderCompID (49). The token and SenderCompID (49) are fields of the Logon(A) message. Both parameters are obtained from GPW. The token value should be provided in FIX field RawData (96) in Logon message.. SenderCompID (49) with assigned value should be included in each FIX message as it is a part of FIX standard header.

The limit of failed login attempts is set to 5. If Participant exceeds the number of login attempts, Gateway blocks the account and the Logon message in will be set to AccountLocked. In the situation of a failed login attempt, Exchange Member is advised to contact the GPW service desk for further assistance.

Client can connect and disconnect with FIX Gateway in sequence many times but no more frequently than once every 3 seconds. it's not allowed to establish multiple FIX connections simultaneously on the same SenderCompID (49). After successful Logon (A) any other FIX connection attempts with Logon message with the same SenderCompID (49) causes disconnection.

5.2. MESSAGE THROTTLING

The throttling mechanism protects Gateway against overloading with an excessive number of input messages.

There are 2 limits set in Gateway:

1. business throttling - after exceeding the business limit level, next messages are rejected using Reject (j) messages with SessionRejectReason (380) = 8 (Throttle limit exceeded).
2. technical throttling - after exceeding the technical limit level FIX Gateway sends Reject (j) message with SessionRejectReason (380) = 9 (Throttle limit exceeded, session is disconnected) and then sends Logout (5) message with error message Text (58) = "Anti-flooding penalty" and connection is closed.

Limit calculations take into account all messages, however, messages Logout (35=5) and OrderCancel (35=F) aren't rejected.

The MassQuote messages are treated differently. When calculating limits for MassQuote, each side of each quotation is counted as a separate message. For example, a MassQuote message containing 8 quotations is counted as 16 messages.

The above limits are calculated over a period of time configurable in Gateway (i.e., measurement period).

- The throttling parameters are configurable at Gateway service as follows:
    - The business limit level is configured for each connection and depends on the level of service provision agreed upon between System Operator and Exchange Member. The measurement period - last 10 seconds.
    - The technical limit level is configured as an internal Gateway parameter and is set to 5000 messages per second. The measurement period - last 100 milliseconds.

In addition after exceeding the technical limit level, it is not possible to log into System for 10 seconds.

Note:
The measurement period means the period of time during which messages are counted and the limits, which have possibly been exceeded, are determined.

5.3. Connectivity Policy

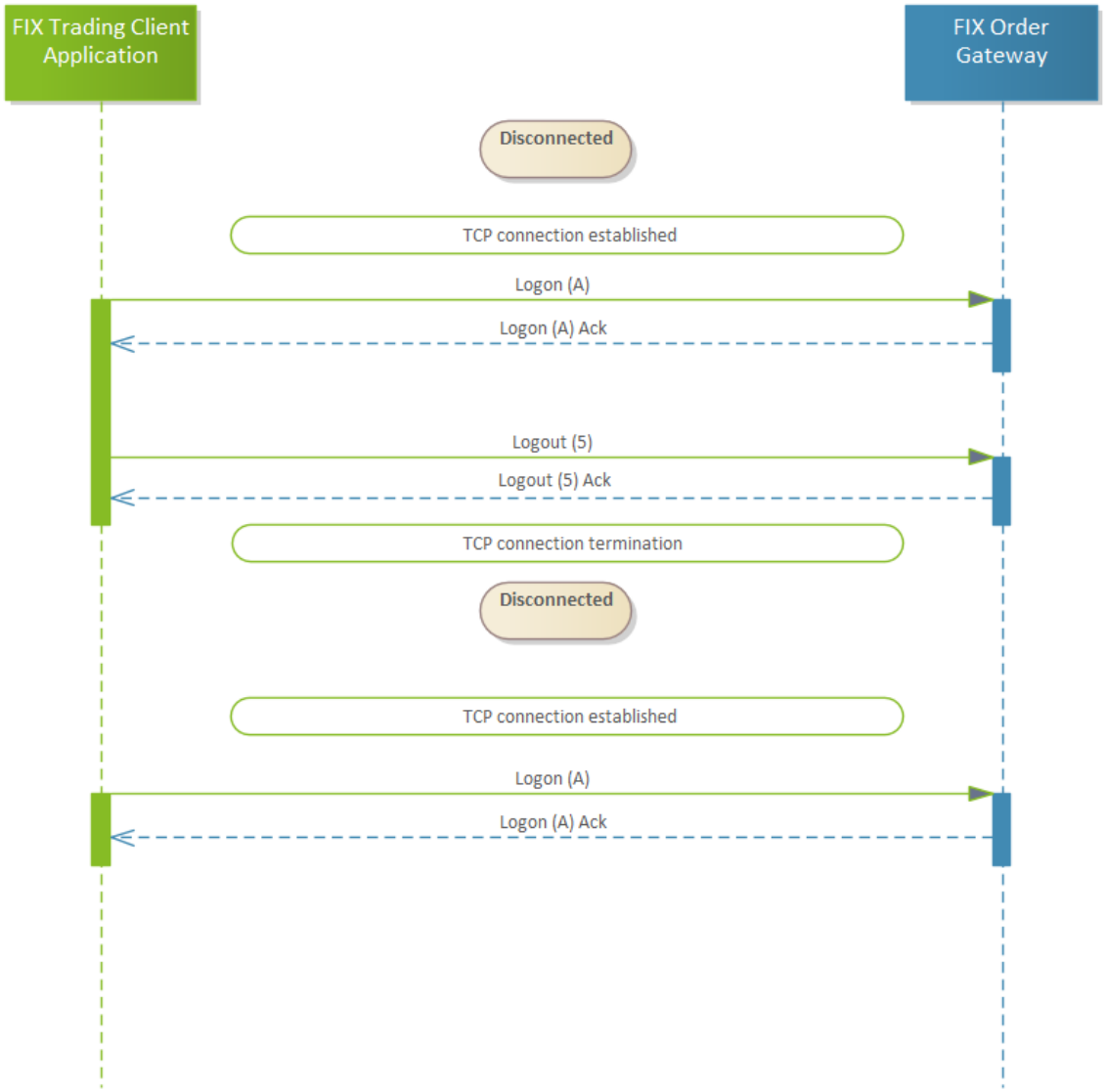
Connectivity policy is described in the document **GPW WATS 6.01 CONNECTIVITY**.

6. Session Layer

This chapters aims to elaborate FIX session layer implemented in GPW WATS.

A FIX session is a bi-directional stream of sequenced messages between two parties of which one establishes the connection and initiates the session (Exchange Member) and another accepts the connection (GPW WATS FIX Gateway). Each party sends messages identified by a unique (across FIX session) sequence number. A single FIX session can exist across multiple sequential (not concurrent) FIX connections, thus peers may intentionally or unintentionally connect and disconnect numerous times while maintaining a single FIX session.

Figure 4. - FIX session with multiple connections



A FIX connection consists of three parts: logon process, message exchange (inclusive of resynchronization of state), and logout process over a transport layer connection. A FIX connection may be concluded by either a logout process or the unrecoverable loss of the transport layer, a system failure, or an application failure.

Each FIX session exists in boundaries of one day. It starts on first Logon and ends in the end of trading day.

6.1. Establishing a FIX Connection

Establishing a FIX Connection consists of three subsequent stages:

1. Exchange Member establishes a TCP connection with FIX Order Gateway. There is limited time since TCP connection is established to perform the second step where Exchange Member sends Logon. If Logon is not sent in 5 seconds then FIX Gateway drops the TCP connection.
2. Exchange Member sends Logon (35=A) message, which should always be the first message transmitted when new FIX connection is established. FIX Gateway authenticates and when authentication is successful it accepts connection and sends a Logon Acknowledgement message to Exchange Member. Exchange Member should not transmit any messages until Logon Acknowledgement has been received and processed by Exchange Member.

Logon message should include HeartBtInt (108) set to 30 representing heartbeat interval.

If authentication fails, FIX Gateway (depending on cause) either sends Logout with information describing the reason for failure and drops connection or drops connection without any message in return. In the majority of cases FIX Gateway sends Logout (35=5) message prior to the connection being terminated and provides a descriptive reason of Logon process failure so Exchange Member can diagnose the issue.

Possible GPW WATS responses to Logon message sent by Exchange Member:

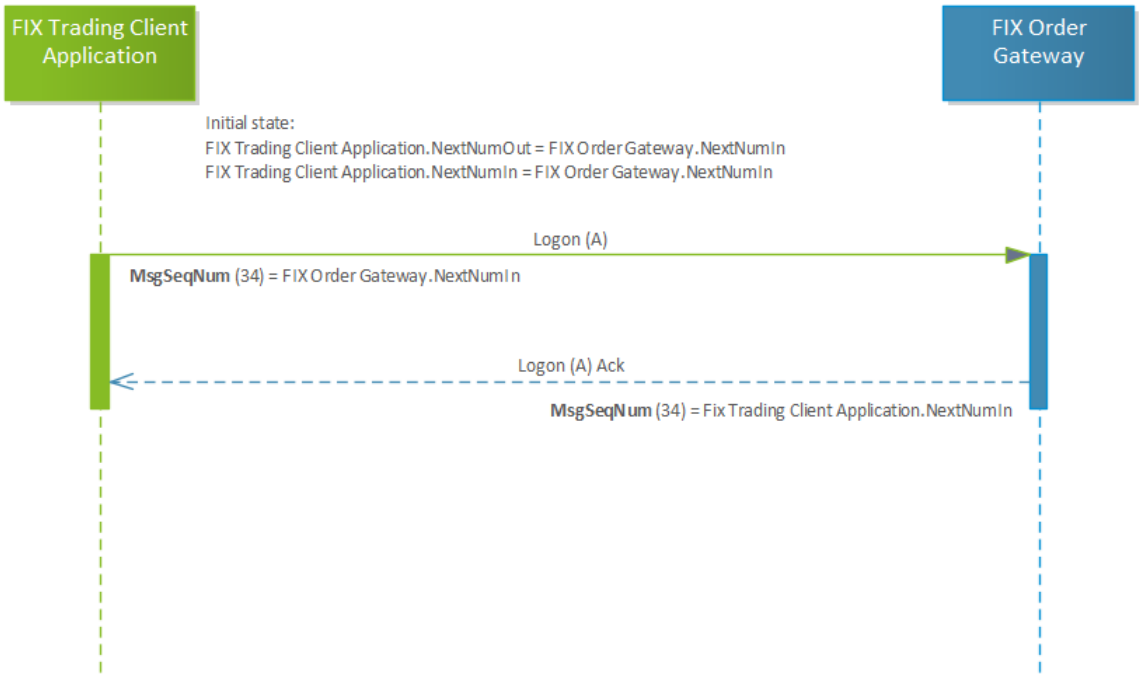
- Logon Acknowledge when authentication is successful,
- Logout message and immediate connection termination in case MsgSeqNum (34) present in Logon message is too low or when MsgSeqNum (34) is not present,
- Logout message and immediate connection termination when HeartBtInt (108) value sent by Client Application is not valid. FIX Gateway provides reason for terminating the connection in Text (58) field by setting it to: "Invalid HeartBtInt (108), expected value 30 seconds",
- Terminating immediate connection without sending Logout message when BeginString (8), SenderCompID (49), TargetCompID (56), or IP address in the Logon (35=A) request is invalid,
- Terminating immediate connection without sending Logout message when there is an existing opened FIX connection for SenderCompID (49).

When TCP connection is established and Exchange Member sends message different than valid Logon (35=A) FIX Gateway terminates connection without sending Logout (35=5) message.

3. If authentication was successful (Logon Acknowledge was sent by FIX Gateway) both Exchange Member and FIX Gateway should synchronize their messages before they continue to send any messages (new or queued). FIX Gateway compares MsgSeqNum (34) received in Logon message to NextNumIn (sequence maintained on FIX Gateway side). The initiator should compare MsgSeqNum (34) received in Logon Ack to NextNumIn maintained on Exchange Member site.

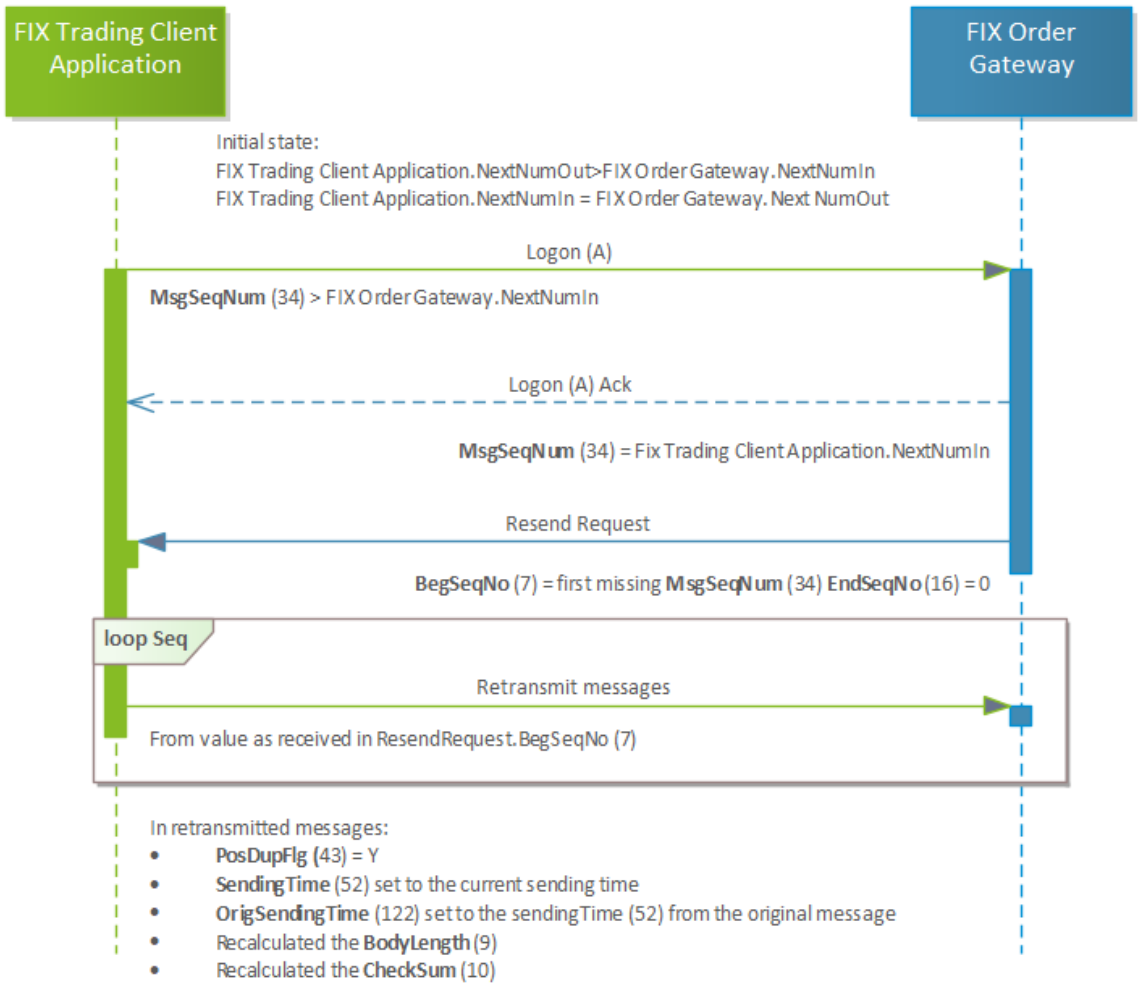
If MsgSeqNum (34) from received messages (Logon and Logon Acknowledge) equals to NextNumIn (MsgSeqNum expected to be received) maintained on both sites, then messages are synchronized.

Figure 5. – Successful logon



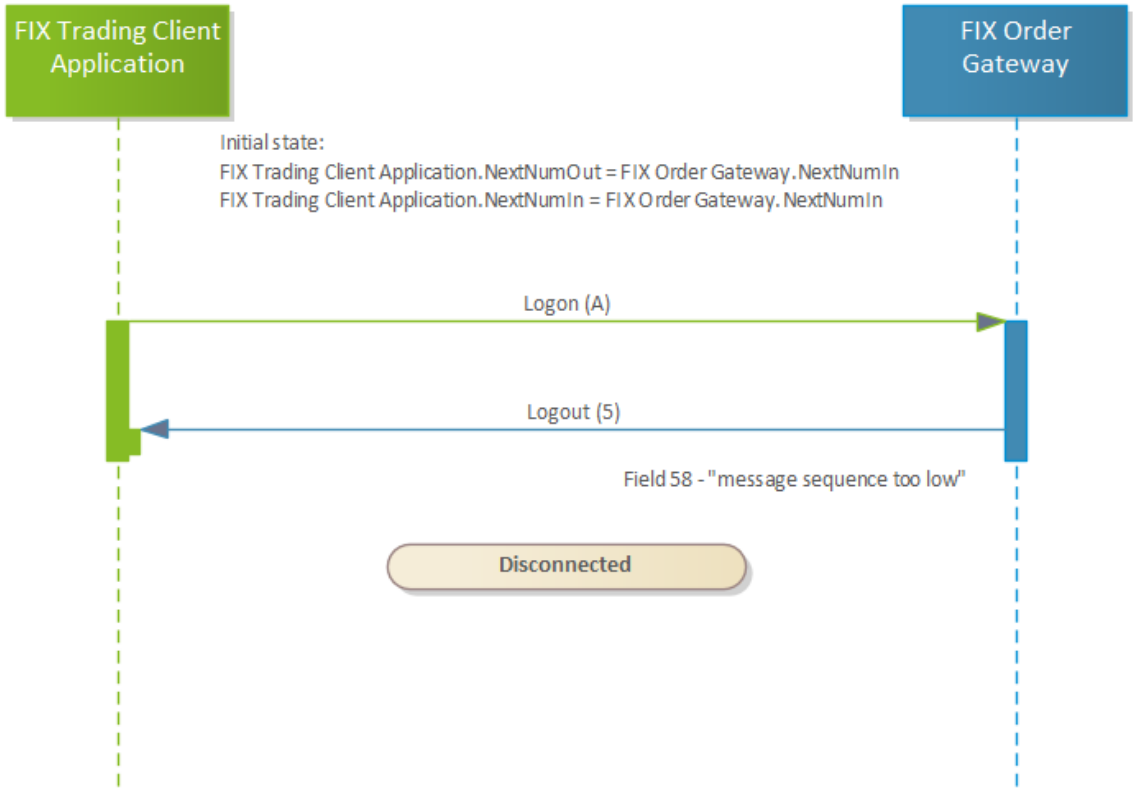
When Exchange Member sends Logon with MsgSeqNum (34) higher than expected by FIX Gateway. FIX Gateway requests Exchange Member to resend missing messages and sends ResendRequest (35=2). When FIX Gateway sends Logon Acknowledge with MsgSeqNum (34) higher than expected by Exchange Member, Exchange Member should request FIX Gateway to resend missing messages.

Figure 6 – Logon with gap fill procedure



When Exchange Member sends Logon with MsgSeqNum (34) lower than expected by FIX Gateway, FIX Gateway sends Logout message and terminates TCP connection. In such a situation connection GPW WATS terminates connection assuming there is an invalid session state on one of sites (Exchange Member or FIX Gateway).

Figure 7. – Unsuccessful logon due to message sequence number too low

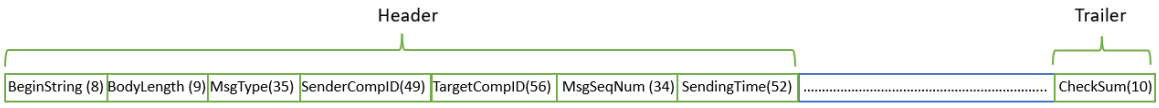


6.2. MAINTAINING A FIX SESSION

GPW FIX Gateway implements a number of mechanisms to enable successful message exchange between Exchange Member and FIX Gateway since FIX session is established until it is finished. FIX Gateway controls such aspects as if Exchange Member is active and responsive, all messages sent by Exchange Member are received, messages are not garbled, received message is correct. These control mechanisms are based on:

- dedicated messages such as Heartbeat, TestRequest, ResentRequests,
- fields present in each message in header and trailer section:

Figure 8. – Header and trailer section



6.2.1. SEQUENCE NUMBER

Each party sends messages identified by unique (across FIX session) and independent sequence numbers. Within each session sequence number should start from 1 and should be incremented by 1 when each message is sent. FIX Gateway maintains NextNumIn (next expected MsgSeqNum (34)) and for each incoming message verifies if MsgSeqNum (34) = NextNumIn.

When MsgSeqNum (34) equals NextNumIn and message is not garbled GPW WATS continues with message processing and increments NextNumIn.

When MsgSeqNum (34) is missing or lower than expected and PossDupFlag (43) is not set to "Y" FIX Gateway sends Logout message (35=5) with Text (58) field describing error and terminates connection.

When MsgSeqNum (34) is higher than expected FIX Gateway assumes there are lost messages from Exchange Member and sends ResendRequest (35=2) to rebuild gap in messages.

Identical validation should be done by Exchange Member. Sequence number maintenance and control is crucial to prevent FIX message losses or duplicate messages.

6.2.2. BODY LENGTH AND CHECKSUM

Each FIX message received by GPW WATS contains mandatory fields BodyLength (9) and CheckSum (10).

When any of these fields are not present or their value is invalid GPW WATS treats such message as garbled, ignores and does not increment NextNumIn.

6.2.3. HEARTBEAT

Heartbeat (35=0) is single directional message (there are no acknowledgements for this message) designed to maintain a live connection during periods of inactivity (other messages are not sent).

Order Gateway sends Heartbeat messages to Client Application each time a message (from Client Application) is not sent in set heartbeat time interval. The heartbeat time interval is set to 30 seconds. It is expected from Client Application to implement the same heartbeat logic.

When Gateway detects no message (including heartbeat) from Client Application for a period longer than heartbeat interval(30 seconds) it sends TestRequest (1) and starts Grace Period process (see chapter 6.2.3). If Client Application does not respond with Heartbeat(0) message in 90 seconds period Gateway sends Logout and terminates FIX connection.

Heartbeat(0) message sent in response to TestRequest (1) should include TestReqID (112) as received in the last TestRequest (1) message.

6.3. Terminating a Fix Connection

Both the Participant and GPW WATS can, at any time, terminate FIX connection by sending Logout request. For normal termination once Logout request is sent, requestor should wait 3 seconds until Logout acknowledgement and then disconnect TCP connection. It is important that Participant waits before terminating connection – it gives GPW WATS Server an opportunity to perform gap fill operations. When MsgSeqNum (34) in Logout message from Client is higher than expected GPW WATS sends ResendRequest (35=2) to rebuild gap in messages. When Client disconnects without Logout process GPW WATS recognizes it as an inappropriate state and starts Cancel on Disconnect mechanism (if enabled for Participant).

At the end of each trading day, before GPW WATS server shuts down, Participant should terminate fix connection by sending Logout message and wait for GPW WATS Logout Acknowledgement.

6.4. Re-establishing a Fix Session

During trading day FIX connection can be established and terminated many times in scope of the same FIX session. Despite if the Fix connection termination being normal (with valid Logout process) or no new FIX connection is established with incremented MsgSeqNum (34).

6.5. RECOVERY

Due to a reduction to operational risk within GPW WATS, the failover and recovery mechanism has been designed.

Failover and recovery modes:

1. **Failover Mode**: This mode is activated in response to temporary connectivity losses. The system is designed to allow reconnection attempts within a timeframe of several seconds, up to approximately fifteen seconds. If the connection cannot be re-established after multiple attempts, users will be required to transition to Recovery Mode.
2. **Recovery Mode**: This mode is implemented when there is a need to shift sessions to an alternative gateway. An alternative gateway may be located at a different SITE.

Detailed rules for switching to Recovery Mode will be outlined in the Business Continuity Plan (BCP) documentation.

6.5.1. FAILOVER MODE

In the case of temporary and short unavailability of Gateway, Exchange Member should attempt to re-establish the connection and login to Gateway. After successful login, a standard process for verifying a messages gap should be carried out.

When Exchange Member or Gateway discovers MsgSeqNum (34) is higher than expected, it should start gap fill procedure to recover lost messages by sending ResendRequest (35=2). Detailed expected behavior depends on FIX connection state and message type.

When MsgSeqNum (34) is higher than expected in Logon (35=A) message then receiver (either Exchange Member or Gateway) should perform Logon process, send Logon Acknowledgement and then send a ResendRequest (35=2) message.

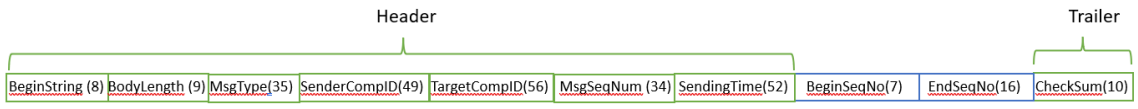
When MsgSeqNum (34) is higher than expected in Logout (35=5) message then receiver (either Exchange Member or Gateway) should send a ResendRequest (35=2) message, receive missing messages and then conclude FIX connection by sending Logout acknowledge message.

When MsgSeqNum (34) is higher than expected in ResendRequest (35=2) message then receiver (either Exchange Member or Gateway) should send a ResendRequest (35=2) message, receive missing messages and then process retransmission of requested messages.

In any other cases receiver (either Exchange Member or Gateway) should send ResendRequest (35=2) and process gap filling procedure.

ResendRequest (35=2) message consists of a Standard Header, BeginSeqNo (7), EndSeqNo (16) and Standard Trailer. When requestor wants one message to be retransmitted EndSeqNo should be equal to BeginSeqNo. When EndSeqNo is set to 0 then all messages since BeginSeqNo should be retransmitted.

Figure 9. – Resend Request message



In response to ResendRequest (35=2) message counterparty should resend messages from requested scope. Each retransmitted message should have:

- PossDupFlag (43) set to Y,
- OrigSendingTime (122) set to the value from the original message (SendingTime (52)),
- SendingTime (52) set to current time from the point of resending message,
- BodyLength (9) recalculated after above changes,
- CheckSum (10) recalculated after above changes.

Each session layer messages (Logon, Logout, Heartbeat, TestRequest, ResentRequest, SequenceReset) should be replaced with SequenceReset (35=4) GapFillFlag (123) set to "Y".

6.5.2. RECOVERY MODE

When the primary Gateway is unavailable for a long period (minimum 15 seconds), the procedure to switch to a secondary could be launched. In such a situation the connection should be established with the same FIX Session (i.e. SenderCompID, TargetCompID and BeginString) parameters but on a new IP address and a port number.

In the recovery mode, it is necessary to fully restore the message state on Exchange Member's side.

The sequence number of the restored messages may differ from those obtained from the previously used gateway. Data consistency on Exchange Member's side should be maintained at the level of business identifiers such as CliOrdId, OrderId, TradeId, TradeReportID, QuoteID.

Exchange Member will set the Logon message with MsgSeqNum and NextExpectedMsgSeqNum set to 1 and begin the login process with a zero state of their received messages storage. The Logon from the gateway side will contain NextExpectedMsgSeqNum set to the expected value. Exchange Member should set their sequence number to the one indicated in the Logon message received from the gateway. This is accomplished by sending a SequenceReset message.

Exchange Member will receive the complete state of messages from the gateway and will be able to continue operations.

In case of further connection issues, please contact the System Operator.

Please note that when using the CoD mechanism and disconnecting without a logout message, which is natural for most failures, orders are cancelled (see section Cancel On Disconnect).

7. Messages

7.1. Standard Header

Tag	Field	Req	Conditional	Description	Type	Length	Values
8	**BeginString**	R		Identifies beginning of new message and protocol version. Aways first field in message. Always unencrypted.	string		FIXT.1.1
9	**BodyLength**	R		Message length in bytes, forward to the CheckSum field. Aways second field in message. Always unencrypted.	length		integer
35	**MsgType**	R		Defines message type. Always third field in message. Always unencrypted.	string		
1128	**ApplVerID**	O		Indicates application version using a service pack identifier. The ApplVerID applies to a specific message occurrence.	string		9 = FIX50SP2
49	**SenderCompID**	R		Assigned value used to identify firm sending message. Always unencrypted.	string	8 chars	alphanumeric, 0-9, A-Z
56	**TargetCompID**	R		Assigned value used to identify receiving firm. Always unencrypted.	string	8 chars	alphanumeric, 0-9, A-Z
115	**OnBehalfOfCompID**	O	Used in Drop Copy service.	Assigned value used to identify firm originating message if the message was delivered by a third party i.e. the third party firm identifier would be delivered in the SenderCompID field and the firm originating the message in this field. Can be embedded within encrypted data section.	string	8 chars	alphanumeric, 0-9, A-Z
34	**MsgSeqNum**	R		Integer message sequence number. Can be embedded within encrypted data section.	seqnum	4 bytes	integer
43	**PossDupFlag**	O	Always required for retransmitted messages, whether prompted by the sending system or as a result of a resend request.	Indicates possible retransmission of message with this sequence number. Can be embedded within encrypted data section.	bool	1 char	N - Original Transmission Y - Possible Duplicate

Tag	Field	Req	Conditional	Description	Type	Length	Values
97	PossResend	O	Required when message may be duplicate of another message sent under a different sequence number.	Indicates that the message may contain information that has been sent under another sequence number. Can be embedded within encrypted data section.	bool	1 char	N - Original Transmission Y - Possible Resend
52	SendingTime	R		Time of message transmission (always expressed in UTC). Can be embedded within encrypted data section.	UTC timestamp		From Gateway: YYYYMMDD-HH:MM:SS.nnnnnnnnn From Exchange Member all listed below formats are accepted: YYYYMMDD-HH:MM:SS.nnnnnnnnn YYYYMMDD-HH:MM:SS.uuuuuu YYYYMMDD-HH:MM:SS.sss
122	OrigSendingTime	O	Required for message resent as a result of a ResendRequest. If data is not available, set to same value as SendingTime.	Original time of message transmission (always expressed in UTC) when transmitting orders as a result of a resend request. Can be embedded within encrypted data section.	UTC timestamp		From Gateway: YYYYMMDD-HH:MM:SS.nnnnnnnnn From Exchange Member all listed below formats are accepted: YYYYMMDD-HH:MM:SS.nnnnnnnnn YYYYMMDD-HH:MM:SS.uuuuuu YYYYMMDD-HH:MM:SS.sss

7.2. Standard Trailer

Tag	Field	Req	Description	Type	Length	Values
10	Checksum	R	Modulo 256 checksum represented on three digits. Always unencrypted.	string	3 chars	

7.3. Logon

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			
98	EncryptMethod	R		Method of encryption.	int	1 byte	0 - None / Other
108	HeartBtInt	R		Heartbeat interval (seconds).	int	8 bytes	30 (seconds)
95	RawDataLength	O		Required for some authentication methods	length		
96	RawData	O		Required for some authentication methods	data		

Tag	Field	Req	Conditional	Description	Type	Length	Values
141	**ResetSeqNumFlag**	O		Indicates that both sides of the FIX session should reset sequence numbers.	bool	1 char	N - No Y - Yes, reset sequence numbers
789	**NextExpectedMsgSeqNum**	O	Optional, alternative via counterparty bi-lateral agreement message gap detection and recovery approach (see "Logon Message NextExpectedMsgSeqNum Processing" section).	Next expected MsgSeqNum value to be received.	seqnum	4 bytes	
1409	**SessionStatus**	O	Session status at time of logon. Field is intended to be used when the logon is sent as an acknowledgement from acceptor of the FIX session.	Status of a FIX session.	int		0 - Session active 4 - Session logout complete 7 - Logons are not allowed at this time 9 - Received MsgSeqNum (34) is too low 10 - Received NextExpectedMsgSeqNum (789) is too high 100 - Unknown user 101 - Invalid user token 102 - Already logged in 103 - Invalid hearbeat interval 104 - Anti-flooding penalty 105 - Connection config changed
1137	**DefaultApplVerID**	R		The default version of FIX messages used in this session.	string		9 = FIX50SP2
58	**Text**	O		Available to provide a response to logon when used as a logon acknowledgement from acceptor back to the logon initiator.	string	18 chars	Free format text string.
C	**StandardTrailer**	R		Trailer used in FIX TP messages			

7.4. Heartbeat (0)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	**StandardHeader**	R		Header used in FIX TP messages			
112	**TestReqID**	C	Required when the heartbeat is the result of a Test Request message.	Identifier provided in Test Request message, returned in resulting heartbeat.	string		
C	**StandardTrailer**	R		Trailer used in FIX TP messages			

7.5. Test Request (1)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			
112	TestReqID	R		Identifier included in Test Request message to be returned in resulting heartbeat.	string		
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.6. Resend Request (2)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			
7	BeginSeqNo	R		Message sequence number of first message in range to be resent.	seqnum	4 bytes	
16	EndSeqNo	R		Message sequence number of last message in range to be resent. If request is for a single message BeginSeqNo (7) - EndSeqNo. If request is for all messages subsequent to a particular message, EndSeqNo = "0" (representing infinity).	seqnum	4 bytes	
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.7. Sequence Reset (4)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			
123	GapFillFlag	O		Indicates if the Sequence Reset message is replacing administrative or application messages (which will not be resent) or reset sequence to a new value.	bool	1 char	N - Sequence Reset, sequence number is being reset to a new value provided in NewSeqNo (36). Y = Gap Fill Message, replacing session layer or application messages which will not be resent when filling gaps in the sequence number.
36	NewSeqNo	R		New sequence number.	seqnum	4 bytes	
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.8. Logout (5)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			
1409	SessionStatus	O	Available only in Logon response (Logon Acknowledge).	Available to provide status of a FIX session in logon acknowledge(response to a logon) message sent by FIX Gateway. Should not be used in Logon send by Client Application initiating FIX connection.	int		0 - Session Active 4 - Session Logout Complete 7 - Logons Are Not Allowed At This Time 9 - Received Msg Seq Num Too Low 10 - Received Next Expected Msg Seq Num Too High 100 - Unknown user 101 - Invalid user token 102 - Already logged in 103 - Invalid hearbeat interval 104 - Anti-flooding penalty 105 - Connection config changed
58	Text	O		Populated with error message when the Logout(5) is sent due to a problem with a FIX connection.	string	18 chars	ASCII 32-126
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.9. Business Message Reject (j)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			
45	RefSeqNum	O		Reference message sequence number.	seqnum	4 vytes	MsgSeqNum of rejected message.
372	RefMsgType	R		The MsgType (35) of the FIX message being referenced.	string	2 chars	
379	BusinessRejectRefID	O	Required unless the corresponding ID field (see list above) was not specified.	The value of the business-level "ID" field on the message being referenced.	string		
380	BusinessRejectReason	R		Code to identify reason for a Business Message Reject message.	int	4 chars	"0 - Other 3 - Unsupported message type 4 - Application not available 5 - Conditionally required field missing 6 - Not authorized 8 - Throttle limit exceeded 9 - Throttle limit exceeded, session will be disconnected 1001 - NoPartyIDs (453) must be equal to 2 or 3 1002 - Duplicated PartyRole (452) value"

Tag	Field	Req	Conditional	Description	Type	Length	Values
58	Text	O		Free format text string, additional description of reject reason.	string	18 chars	Where possible, message to explain reason for rejection.
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.10. Reject (3)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			
45	RefSeqNum	O		Reference message sequence number.	seqnum	4 bytes	MsgSeqNum of rejected message.
371	RefTagID	O		The tag number of the FIX field being referenced.	int		
372	RefMsgType	R		The MsgType (35) of the FIX message being referenced.	string	2 chars	
1130	RefAppVerID	O		Specifies the service pack release being applied to a message at the session level. Enumerated field with values assigned at time of service pack release. Uses same values as ApplVerID.	string		9 = FIX50SP2
373	SessionRejectReason	R		Code to identify reason for a Business Message Reject message.	int	4 chars	"0 - Invalid tag number 1 - Required tag missing 2 - Tag not defined for this message type 3 - Undefined tag 4 - Tag specified without a value 5 - Value is incorrect (out of range) for this tag 6 - Incorrect data format for value 9 - CompID problem 10 - SendingTime accuracy problem 11 - Invalid MsgType 13 - Tag appears more than once 14 - Tag specified out of required order 15 - Repeating group fields out of order 16 - Incorrect NumInGroup count for repeating group 17 - Non ""Data"" value includes field delimiter (<SOH> character) 18 - Invalid / Unsupported application version 99 - Other"
58	Text	O		Free format text string, additional description of reject reason.	string	18 chars	Where possible, message to explain reason for rejection.

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.11. New Order Single (D)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			
11	ClOrdID	R		Unique identifier for an order assigned by Exchange Member. Trading Members that submit GTD / GTC orders must ensure uniqueness across days, for example by embedding a date within the ClOrdID value.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
C	Parties	R					
453	NoPartyIDs	R		Number of repetitions of PartyID (448), PartyIDSource (447), PartyRole (452), PartyRoleQualifier (2376) groupings.	numingroup	1 byte	integer
→448	PartyID	R		Identification of the party. The short code representing the client, execution decision maker or investment decision maker represented by this block (depending on the PartyRole value).	"For PartyIDSource (447) = D: string For PartyIDSource (447) = N: string For PartyIDSource (447) = P: integer"	For PartyIDSource (447) = D and a) PartyRole (452) = 4: 4 bytes b) PartyRole (452) = 33: 8 bytes For PartyIDSource (447) = N: 20 bytes For PartyIDSource (447) = P: 4 bytes	"For PartyIDSource (447) = P: integer 4 - 4.294.967.295 (4 bytes) The following values are reserved for applicable use. Applicable to PartyRole = 3: • 1 = AGGR (an aggregation of multiple client orders) • 2 = PNAL (clients are pending allocation) Applicable to PartyRole = 12: • 3 = NORE (timing and location of the execution determined by the client of the participant)"
→447	PartyIDSource	R		Used to identify classification source.	char	1 char	D - Proprietary / Custom code

Tag	Field	Req	Conditional	Description	Type	Length	Values
							N - Legal Entity Identifier (LEI) P - Short code identifier
→452	PartyRole	R		Identifies the type of PartyID (448). Specifies the role represented by the short code provided in the PartyID tag.	int		"For PartyIDSource (447) = D: • 4 = Clearing Firm • 33 = Interested Party For PartyIDSource (447) = P: • 3 = Client ID • 12 = Executing Trader • 122 = Investment Decision Maker"
→2376	PartyRoleQualifier	C	Field not provided in case of reserved short codes in PartyID (448) field (i.e. 1, 2, 3) or in case of PartyRole (452) = 4 (Clearing Firm) or 33 (Interested Party).	Provides further qualification of the PartyRole (452) value.	int		22 - Algorithm (applicable to PartyRole values 12 or 122) 23 - Firm or legal entity (LEI) (applicable to PartyRole value 3) 24 - Natural person (applicable to PartyRole values 3, 12, 122)
1	Account	O		Account provided by exchange member for clearing purposes.	string	16 chars	alphanumeric, 0-9, A-Z, a-z or ASCII 33-126
581	AccountType	O		Type of account associated with the order.	int		1 - Account is carried on customer side of the books 3 - House trader
18	ExecInst	O		Instructions for order handling on exchange trading floor.	char	1 byte	o = Cancel on connection loss
2362	SelfMatchPreventionID	O		Identifies orders that should not be matched against each other if both of them contain the same SelfMatchPreventionID (2362) for the same instrument and were submitted by the same Exchange Member.	int	1 byte	integer
C	DisplayInstruction						
1138	DisplayQty	C	Conditionally required for Iceberg orders. Only allowed	Initially displayed quantity for Iceberg order in Market Data public order book (Peak Quantity).	qty	4 bytes	integer

Tag	Field	Req	Conditional	Description	Type	Length	Values
			for OrdType (40) = 2 (Limit).	DisplayQty is expressed in number of Lots.			
C	**Instrument**	R					
48	**SecurityID**	R		Security identifier value of SecurityIDSource (22) type (e.g. ISIN, exchange symbol, etc.). Requires SecurityIDSource.	string	4 bytes	integer
22	**SecurityIDSource**	R		Identifies class or source of the SecurityID (48) value.	string	1 char	8 - exchange symbol
54	**Side**	R		Side of order.	char	1 char	1 - Buy 2 - Sell
60	**TransactTime**	R		Time the order submission process was initiated by exchange member. This time should be earlier than message SendingTime (52) provided in the header.	UTC timestamp		The following formats are accepted: YYYYMMDD-HH:MM:SS.nnnnnnnnn YYYYMMDD-HH:MM:SS.uuuuuu YYYYMMDD-HH:MM:SS.sss
C	**OrderQtyData**	R					
38	**OrderQty**	R		Initial total order quantity (expressed in number of Lots).	qty	4 bytes	integer
40	**OrdType**	R		Type of order.	char	1 char	1 - Market 2 - Limit 3 - Stop Loss 4 - Stop Limit K - Market With Left Over as Limit
44	**Price**	C	Conditionally required if OrdType = 2 (Limit) or 4 (Stop Limit).	Price per unit of quantity(smallest portion of instrument)	price	12 digits + decimal point	Floating point number. 6 digits for integer part, decimal point, 6 digits for decimal part. Negative values and 0 allowed.
C	**TriggeringInstruction**	C					
1102	**TriggerPrice**	C	Conditionally required for OrdType (40) = 3 (Stop Loss) or 4 (Stop Limit).	The price at which the order is activated and inserted into the order book.	price	12 digits + decimal point	Floating point number. 6 digits for integer part, decimal point, 6 digits for decimal part. Negative values and 0 allowed.

Tag	Field	Req	Conditional	Description	Type	Length	Values
59	**TimeInForce**	R		Specifies when the order should become expired (Day, Good Till Time, Good Till Date, Good Till Cancel), when it should be activated (At The Opening, At The Closing), or how it should be executed (Immediate Or Cancel, Fill Or Kill).	char	1 byte	0 - Day 1 - Good Till Cancel (GTC) 2 - At The Opening (VFA) 3 - Immediate Or Cancel (IOC) 4 - Fill Or Kill (FOK) 6 - Good Till Date (GTD) 7 - At The Closing (VFC)
432	**ExpireDate**	C	Conditionally required if TimeInForce = 6 and ExpireTime (126) is not specified.	Date of order expiry (last day the order can trade), always expressed in terms of the local market date. ExpireDate and ExpireTime tags are mutually exclusive.	localmktdate		The following formats are accepted: YYYYMMDD (local)
126	**ExpireTime**	C	Conditionally required if TimeInForce = 6 and ExpireDate (432) is not specified.	Time of order expiry (always expressed in UTC (Universal Time Coordinated, also known as "GMT"). It always refers to the current trading day. ExpireDate and ExpireTime tags are mutually exclusive.	UTC timestamp		The following formats are accepted: YYYYMMDD-HH:MM:SS (UTC)
528	**OrderCapacity**	R		Designates the capacity of the Trading Firm placing the order.	char	1 char	A - Agency (mapped to AOTC) P - Principal (mapped to DEAL) R - Riskless Principal (mapped to MTCH)
529	**OrderRestrictions**	O		Designation required for regular Liquidity Providers (as opposite to Issuer Liquidity Providers). When an Issuer Liquidity Provider orders tag should be omitted.	char	1 char	5 - Acting as Market Maker or Specialist in the security
C	**OrderAttributeGrp**						
2593	**NoOrderAttributes**	R		Number of order attribute entries.	int	1 byte	1
→2594	**OrderAttributeType**	R		The type of order attribute.	int	1 byte	2 - Liquidity provision activity order
→2595	**OrderAttributeValue**	R		The value associated with the order attribute type specified in OrderAttributeType (2594). As opposed to OrderRestriction tag specifies whether the order poses any type of liquidity provision activity (either regular or issuer).	string	1 char	Y - Yes N - No
58	**Text**	O		Free format text string for Exchange Member use.	string	18 chars	alphanumeric, 0-9, A-Z, a-z

Tag	Field	Req	Conditional	Description	Type	Length	Values
							or ASCII 33-126
1724	OrderOrigination	O		Designation required for orders submitted via Direct Electronic Access (both DMA or Sponsored Access). In case of regular trading access this tag should be omitted.	int		5 - Order received from a direct access or sponsored access customer
20011	FeeStructureID	O		Optional identifier of a fee scheme for billing purposes.	int	1 byte	1 to 99 - for internal purposes for Members 100 to 255 - for internal purposes for WSE (GPW)
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.12. Execution Report (8)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			Header used in FIX TP messages
37	OrderID	R		Unique identifier for an order assigned by Trading System. OrderID is unique across all trading days, all order books, and all exchange members, and it is persistent for the entire life of an order (i.e. it does not change after order modification). OrderID is generated for each order as a concatenation of connection ID + sequential number of a trading day (starting from 1) + sequence number of the message.	string	8 bytes	numeric
278	MDEntryID	R		Identifier for an order assigned by Trading System for Market Data purposes. Corresponds to the publicOrderId field in Market Data messages. MDEntryID is unique across all order books and all exchange members, but only within a single trading day. On the next trading day the numbering is reset and starts afresh (i.e. MDEntryID is not persistent for the entire life of an order). Moreover it also	string	8 bytes	numeric

Tag	Field	Req	Conditional	Description	Type	Length	Values
				changes with each refill of an Iceberg order.			
11	**ClOrdID**	R		Unique identifier for an order assigned by Exchange Member. Exchange Members who submit GTD / GTC orders must ensure uniqueness across days, for example by embedding a date within the ClOrdID value.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
41	**OrigClOrdID**	O	Provided in response to a Cancel Request or Cancel/Replace Request (ExecType (150) = 4 or 5).	OrigClOrdID is the last ClOrdID in the modification chain for an order.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
C	**Parties**	R					
453	**NoPartyIDs**	R		Number of repetitions of PartyID (448), PartyIDSource (447), PartyRole (452), PartyRoleQualifier (2376) groupings.	numingroup	1 byte	integer
→448	**PartyID**	R		Identification of the party. The short code representing the client, execution decision maker or investment decision maker represented by this block (depending on the PartyRole value).	For PartyIDSource (447) = D: string For PartyIDSource (447) = N: string For PartyIDSource (447) = P: integer	For PartyIDSource (447) = D and a) PartyRole (452) = 4: 4 bytes b) PartyRole (452) = 33: 8 bytes For PartyIDSource (447) = N: 20 bytes For PartyIDSource (447) = P: 4 bytes	integer 4 - 4.294.967.295 (4 bajts) The following values are reserved for applicable use. Applicable to PartyRole = 3: 1 = AGGR (an aggregation of multiple client orders) 2 = PNAL (clients are pending allocation) Applicable to PartyRole = 12: 3 = NORE (timing and location of the execution determined by the client of the participant)
→447	**PartyIDSource**	R		Used to identify classification source.	char	1 char	D - Proprietary / Custom code P - Short code identifier

Tag	Field	Req	Conditional	Description	Type	Length	Values
→452	PartyRole	R		Identifies the type of PartyID (448). Specifies the role represented by the short code provided in the PartyID tag.	int		For PartyIDSource (447) - D: • 4 - Clearing Firm • 33 – Interested Party For PartyIDSource (447) - P: • 3 - Client ID • 12 - Executing Trader • 122 - Investment Decision Maker
→2376	PartyRoleQualifier	CR	Provided for PartyIDSource (447) = P (Short code identifier) and PartyID>=4(.its not present reserved values 1-3)	Provides further qualification of the PartyRole (452) value.			22 - Algorithm (applicable to PartyRole values 12 or 122) 23 - Firm or legal entity (LEI) (applicable to PartyRole value 3) 24 - Natural person (applicable to PartyRole values 3, 12, 122)
880	TrdMatchID	C	Provided for trades (ExecType (150) = F or H).	Identifier assigned to a trade by a matching system.	string	4 bytes	numeric
17	ExecID	R		Unique identifier of Execution Report (8) message.	string	29 chars	alphanumeric, 0-9, A-Z, a-z
19	ExecRefID	C	Provided for ExecType (150) = H (Trade Cancel).	ExecID (17) value of Execution Report (8) message informing about a trade that is being cancelled.	string	18 chars	alphanumeric, 0-9, A-Z, a-z
150	ExecType	R		Describes the type of Execution Report (8) message, while OrderStatus (39) provides the current status of an order in Trading System.	char	1 char	0 = New 4 = Canceled 5 = Replaced 8 = Rejected C = Expired D = Restated F = Trade H = Trade Cancel L = Triggered
2431	ExecTypeReason	C	Provided in case of ExecType (150) = 4 (Cancelled) or D (Restated) or F (Trade)	Provides detailed information on ExecType (150) in case of order restatement, Iceberg order refill, unsolicited order cancellation, first trade resulting from aggressive order matching.	char	1 char	100 - GT order restatement 101 - Iceberg order refill 102 - Cancel IOC/FOK order 103 - Cancel by STP 104 - Cancel by Market Operations 105 - Cancel on Trading Halt 106 - Cancel on Disconnect 107 - Cancel by Corporate Action 108 – Cancel by Mass Cancel 110 - First trade on aggressive order

Tag	Field	Req	Conditional	Description	Type	Length	Values
39	**OrdStatus**	R		Provides the current status of an order in Trading System.	char	1 char	0 = New 1 = Partially Filled 2 = Filled 4 = Canceled 8 = Rejected C = Expired
103	**OrdRejReason**	C	Provided for ExecType (150) = 8 (Rejected).	Code to identify reason for order rejection.	int	4 chars	Values for OrdRejReason are presented in the *GPW WATS 2.03 Rejection Codes* document.
2667	**AlgorithmicTradeIndicator**	C	Provided for ExecType (150) = F (Trade) or H (Trade Cancel).	Specifies that the trade originates from a computer program or algorithm requiring little-to-no human intervention. In the context of ESMA MiFID II, a trade has to be flagged as "algorithmic" if at least one of the matched orders was submitted by a trading algorithm.	int		0 = Non-algorithmic trade 1 = Algorithmic trade
1	**Account**	O		Account provided by exchange member for clearing purposes.	string	16 chars	alphanumeric, 0-9, A-Z, a-z or ASCII 33-126
581	**AccountType**	O		Type of account associated with the order.	int		1 = Account is carried on customer side of the books 3 = House trader
C	**Instrument**	R					
48	**SecurityID**	R		Security identifier value of SecurityIDSource (22). Numerical (non-ISIN) instrument identifier provided on a daily basis via Market Data reference services.	string	4 bytes	integer
22	**SecurityIDSource**	R		Identifies class or source of the SecurityID (48) value.	string	1 char	8 = exchange symbol
54	**Side**	R		Side of order.	char	1 char	1 = Buy 2 = Sell
C	**OrderQtyData**	R					
38	**OrderQty**	R		Total order quantity (expressed in number of Lots).	int	4 bytes	integer

Tag	Field	Req	Conditional	Description	Type	Length	Values
40	OrdType	R		Type of order.	char	1 char	1 = Market 2 = Limit 3 = Stop Loss 4 = Stop Limit K = Market With Left Over as Limit
44	Price	N	Provided for OrdType (40) = 2 (Limit) or 4 (Stop Limit).	Price per unit of quantity(smallest portion of instrument)	price	12 digits + decimal point	Floating point number. 6 digits for integer part, decimal point, 6 digits for decimal part. Negative values and 0 allowed.
C	TriggeringInstruction						
1102	TriggerPrice	C	Provided for OrdType (40) = 3 (Stop Loss) or 4 (Stop Limit).	The price at which the order is activated and inserted into the order book.	price	12 digits + decimal point	Floating point number. 6 digits for integer part, decimal point, 6 digits for decimal part. Negative values and 0 allowed.
1823	Triggered	C	Provided for TimeInForce (59) = 2 (VFA) or 7 (VFC), and OrdType (40) = 3 (Stop Loss) or 4 (Stop Limit).	Indicates whether an order has been already triggered or not. When triggered VFA/VFC orders value 1, for triggered Stop orders value 2.	int	1 char	0 = Not triggered (default) 1 = Triggered 2 = Stop order triggered
15	Currency	R		Identifies currency used for price (i.e. instrument trading currency).	currency	3 bytes	
59	TimeInForce	R		Specifies when the order should become expired (Day, Good Till Time, Good Till Date, Good Till Cancel), when it should be activated (At The Opening, At The Closing), or how it should be executed (Immediate Or Cancel, Fill Or Kill).	char	1 char	0 = Day 1 = Good Till Cancel (GTC) 2 = At The Opening (VFA) 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FOK) 6 = Good Till Date (GTD) 7 = At The Closing (VFC)
432	ExpireDate	C	Conditionally required if TimeInForce = 6 and ExpireDate (432) is not specified.	Date of order expiry (last day the order can trade), always expressed in terms of the local market date. ExpireDate and ExpireTime tags are mutually exclusive.	localmktdate		YYYYMMDD (local)
126	ExpireTime	C	Conditionally required if TimeInForce = 6 and ExpireDate (432) is not specified.	Time of order expiry (always expressed in UTC (Universal Time Coordinated, also known as "GMT"). It always refers to the current trading day. ExpireDate and ExpireTime tags are mutually exclusive.	UTC timestamp		YYYYMMDD-HH:MM:SS (UTC)

Tag	Field	Req	Conditional	Description	Type	Length	Values
528	**OrderCapacity**	R		Designates the capacity of the Trading Firm placing the order.	char	1 char	A = Agency (mapped to AOTC) P = Principal (mapped to DEAL) R = Riskless Principal (mapped to MTCH)
529	**OrderRestrictions**	O		Designation required for regular Liquidity Providers (as opposite to Issuer Liquidity Providers). When Issuer Liquidity Provider orders tag will not be provided.	char	1 char	5 = Acting as Market Maker or Specialist in the security
C	**OrderAttributeGrp**						
2593	**NoOrderAttributes**	R		Number of order attribute entries.	int	1 byte	1
→2594	**OrderAttributeType**	R		The type of order attribute.	int	1 byte	2 = Liquidity provision activity order
→2595	**OrderAttributeValue**	R		The value associated with the order attribute type specified in OrderAttributeType (2594). As opposite to OrderRestriction tag specifies whether the order poses any type of liquidity provision activity (either regular or issuer).	string	1 char	Y = Yes N = No
32	**LastQty**	C	Provided for ExecType (150) = F (Trade) or H (Trade Cancel).	Quantity of last execution (expressed in number of Lots).	qty	4 bytes	integer
31	**LastPx**	C	Provided for ExecType (150) = F (Trade) or H (Trade Cancel).	Price of last execution per unit of quantity(smallest portion of instrument).	price	12 digits + decimal point	Floating point number. 6 digits for integer part, decimal point, 6 digits for decimal part. Negative values and 0 allowed.
151	**LeavesQty**	R		Quantity remaining for further execution (expressed in number of Lots). When an order that is no longer active, i.e. OrdStatus (39) = 4 (Canceled), C (Expired), 8 (Rejected), LeavesQty (151) is set to 0, otherwise LeavesQty (151) = OrderQty (38) - CumQty (14).	qty	4 bytes	integer
14	**CumQty**	R		Total executed order quantity (expressed in number of Lots).	qty	4 bytes	integer
60	**TransactTime**	R		Timestamp when the business event represented by the message occurred in Trading System. This time should be earlier than message SendingTime (52) provided in the header.	UTC timestamp		The following format is returned from Gateway: YYYYMMDD-HH:MM:SS.nnnnnnnnn

Tag	Field	Req	Conditional	Description	Type	Length	Values
2362	**SelfMatchPreventionID**	O		Identifies orders that should not be matched against each other if both of them contain the same SelfMatchPreventionID (2362) for the same instrument and were submitted by the same Exchange Member.	int	1 byte	
C	**DisplayInstruction**						
1138	**DisplayQty**	O		Currently displayed quantity for Iceberg order in Market Data public order book (as opposed to initially displayed quantity). DisplayQty is expressed in number of Lots.	qty	4 bytes	integer
1608	**InitialDisplayQty**	O		Initially display quantity requested for Iceberg order in New Order Single (D) or Order Cancel/Replace Request (G) messages. InitialDisplayQty is expressed in number of Lots.	qty	4 bytes	integer
58	**Text**	O		Free format text string provided by Exchange member in New Order Single (D) message.	string	18 chars	alphanumeric, 0-9, A-Z, a-z or ASCII 33-126
638	**PriorityIndicator**	O		Required if Cancel/Replace accepted.	int		0 = Priority unchanged 1 = Lost priority as a result of order change
851	**LastLiquidityInd**	C	Provided for ExecType (150) = F (Trade).	Indicates whether the order added liquidity or removed liquidity from the order book (looking from the perspective of the current execution).	int		1 = Added liquidity 2 = Removed liquidity 4 = Auction execution
797	**CopyMsgIndicator**	O		Specifies the leg executions of a multi-leg order or quote.	bool	1 char	
1724	**OrderOrigination**	O		Designation required for orders submitted via Direct Electronic Access (both DMA or Sponsored Access). When regular trading access this tag should be omitted.	int		5 = Order received from a direct access or sponsored access customer
18	**ExecInst**	O		Instructions for order handling on exchange trading floor.	multiplecharvalue		O = Cancel on connection loss
20011	**FeeStructureID**	O		Optional identifier of a fee scheme for billing purposes.	num	1 byte	1 to 99 - for internal purposes for Members

Tag	Field	Req	Conditional	Description	Type	Length	Values
							100 to 255 - for internal purposes for WSE (GPW)
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.13. Order Cancel Replace Request (G)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			Header used in FIX TP messages
37	OrderID	O		Unique identifier for an order assigned by the trading system. OrderID is unique across all trading days, all order books, and all Exchange Members, and it is persistent for the entire life of an order (i.e. it does not change after order modification). OrderID is generated for each order as a concatenation of connection ID + sequential number of a trading day (starting from 1) + sequence number of the message.	string	8 bytes	numeric
41	OrigClOrdID	O		ClOrdID (11) of the previous order (not the initial order of the day) as assigned by the institution, used to identify the previous order in cancel and cancel/replace requests.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
11	ClOrdID	R		Unique identifier for an order assigned by the Exchange Member. Exchange Members that submit GTD / GTC orders must ensure uniqueness across days, for example by embedding a date within the ClOrdID value.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
C	Parties	R					
453	NoPartyIDs	R		Number of repetitions of PartyID (448), PartyIDSource (447), PartyRole (452), PartyRoleQualifier (2376) groupings.	numingroup	1 byte	integer
→448	PartyID	R		Identification of the party. The short code representing the client, execution decision maker or investment decision maker represented by this block (depending on the PartyRole value).	int	4 bytes	integer 4 - 4.294.967.295 (4 bytes) The following values are reserved for applicable use. Applicable to PartyRole = 3: • 1 = AGGR (an aggregation of multiple client orders)

Tag	Field	Req	Conditional	Description	Type	Length	Values
							• 2 = PNAL (clients are pending allocation) Applicable to PartyRole = 12: • 3 = NORE (timing and location of the execution determined by the client of the participant)
→447	PartyIDSource	R		Used to identify classification source.	char	1 char	P = Short code identifierf
→452	PartyRole	R		Identifies the type of PartyID (448). Specifies the role represented by the short code provided in the PartyID tag.	int		For PartyIDSource (447) = P: • 3 = Client ID • 12 = Executing Trader • 122 = Investment Decision Maker
→2376	PartyRoleQualifier		Conditionally required for PartyIDSource (447) = P (Short code identifier) and PartyID >=4(Its not present for reserved values: 1,2,3).	Provides further qualification of the PartyRole (452) value.	int		22 = Algorithm (applicable to PartyRole values 12 or 122) 23 = Firm or legal entity (LEI) (applicable to PartyRole value 3) 24 = Natural person (applicable to PartyRole values 3, 12, 122)
C	DisplayInstruction						
1138	DisplayQty	C	Required for Iceberg order modification.	The quantity to be displayed. Required for Iceberg orders. On orders specifies the quantity to be displayed, on execution reports the currently displayed quantity. DisplayQty is expressed in number of Lots.	qty	4 bytes	integer
C	Instrument	R					
48	SecurityID	R		Security identifier value of SecurityIDSource (22). Numerical (non-ISIN) instrument identifier provided on a daily basis via Market Data reference services.	string	4 bytes	integer
22	SecurityIDSource	R		Identifies class or source of the SecurityID (48) value.	string	1 char	1 8 = Exchange symbol
54	Side	R		Side of order.	char	1 char	1 = Buy 2 = Sell
60	TransactTime	R		Time the order modification process was initiated by the Trading Member. This time should be earlier than message SendingTime (52) provided in the header.	UTC timestamp		The following formats are accepted: YYYYMMDD-HH:MM:SS.nnnnnnnnn (UTC) YYYYMMDD-HH:MM:SS.uuuuuu (UTC) YYYYMMDD-HH:MM:SS.mmm (UTC)
C	OrderQtyData	R					

Tag	Field	Req	Conditional	Description	Type	Length	Values
38	OrderQty	R		Initial total order quantity (expressed in number of Lots)..	qty	4 bytes	integer
40	OrdType	R		Type of the order.	char	1 char	1 = Market 2 = Limit 3 = Stop Loss 4 = Stop Limit K = Market With Left Over as Limit
44	Price	C	Conditionally required if OrdType = 2 (Limit) and OrdType = 4 (Stop Limit), otherwise forbidden.	Price per unit of quantity (smallest portion of instrument)	price	12 digits + decimal point	Floating point number. 6 digits for integer part, decimal point, 6 digits for decimal part. Negative values and 0 allowed.
C	TriggeringInstruction	C					
1102	TriggerPrice	C	Provided in case of OrdType (40) = 3 (Stop Loss) or 4 (Stop Limit).	The price at which the order is activated and inserted into the order book.	price	12 digits + decimal point	Floating point number. 6 digits for integer part, decimal point, 6 digits for decimal part. Negative values and 0 allowed.
432	Expire Date	C	Conditionally required if TimeInForce = 6 and ExpireDate (432) was set in the ordiginal order.	Date of order expiration (last day the order can trade), always expressed in terms of the local market date. ExpireDate and ExpireTime tags are mutually exclusive.	lokalmkdate		The following formats are accepted: YYYYMMDD (local)
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.14. Order Cancel Reject (9)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			Header used in FIX TP messages
37	OrderID	R		Unique identifier for an order assigned by the trading system. OrderID is unique across all trading days, all order books, and all trading members, and it is persistent for the entire life of an order (i.e. it does not change after order modification). OrderID is generated for each order as a concatenation of	string	8 bytes	numeric

Tag	Field	Req	Conditional	Description	Type	Length	Values
				connection ID + sequential number of a trading day (starting from 1) + sequence number of the message.			
11	**ClOrdID**	R		Unique identifier for an order assigned by the trading member. Exchange Members that submit GTD / GTC orders must ensure uniqueness across days, for example by embedding a date within the ClOrdID value.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
41	**OrigClOrdID**	O		ClOrdID (11) of the previous order (not the initial order of the day) as assigned by the institution, used to identify the previous order in cancel and cancel/replace requests.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
39	**OrdStatus**	R		OrdStatus value after this cancel reject is applied.	char	1 char	0 = New 1 = Partially Filled 2 = Filled 4 = Canceled 8 = Rejected C = Expired
60	**TransactTime**	R		Timestamp when the business event represented by the message occurred in the trading system. This time should be earlier than message SendingTime (52) provided in the header.	UTC timestamp		The following format is returned by Gateway: YYYYMMDD-HH:MM:SS.nnnnnnnnn (UTC)
434	**CxlRejResponseTo**	R		Identifies the type of request that a Cancel Reject is in response to.	char	1 char	1 = Order Cancel Request 2 = Order Cancel/Replace Request
102	**CxlRejReason**	R		Code to identify reason for Cancel or Cancel/Replace rejection.	int	4 bytes	Please see FIX Rejection Codes document for the list of available codes with their descriptions.
C	**Parties**	R					

Tag	Field	Req	Conditional	Description	Type	Length	Values
453	**NoPartyIDs**	R		Number of repetitions of PartyID (448), PartyIDSource (447), PartyRole (452), PartyRoleQualifier (2376) groupings.	numingroup	1 byte	integer
→448	**PartyID**	R		Identification of the party. The short code representing the client, execution decision maker or investment decision maker represented by this block (depending on the PartyRole value).	"For PartyIDSource (447) = D : string For PartyIDSource (447) = N : string For PartyIDSource (447) = P : integer"	"For PartyIDSource (447) = D : 4 bytes For PartyIDSource (447) = N : 20 bytes For PartyIDSource (447) = P : 4 bytes"	"integer 4 - 4.294.967.295 (4 bajts) The following values are reserved for applicable use. Applicable to PartyRole = 3 : 1 = AGGR (an aggregation of multiple client orders) 2 = PNAL (clients are pending allocation) Applicable to PartyRole = 12 : 3 = NORE (timing and location of the execution determined by • the client of the participant)"
→447	**PartyIDSource**	R		Used to identify classification source.	char	1 char	D = Proprietary / Custom code N = Legal Entity Identifier (LEI) for exchange member clients. P = Short code identifier
→452	**PartyRole**	R		Identifies the type of PartyID (448). Specifies the role represented by the short code provided in the PartyID tag.	int		For PartyIDSource (447) = D: • 4 = Clearing Firm For PartyIDSource (447) = P: • 3 = Client ID • 12 = Executing Trader • 122 = Investment Decision Maker
→2376	**PartyRoleQualifier**		Conditionally required for PartyIDSource (447) = P (Short code identifier) and PartyID >=4(Its	Provides further qualification of the PartyRole (452) value.	int		22 = Algorithm (applicable to PartyRole values 12 or 122) 23 = Firm or legal entity (LEI) (applicable to PartyRole value 3) 24 = Natural person (applicable to PartyRole values 3, 12, 122)

Tag	Field	Req	Conditional	Description	Type	Length	Values
			not present for reserved values: 1,2,3).				
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.15. Order Cancel Request (F)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			
41	OrigClOrdID	O	Provided in the response to a Cancel Request or Cancel/Replace Request (ExecType (150) = 4 or 5).	ClOrdID (11) of the previous order (not the initial order of the day) as assigned by the institution, used to identify the previous order in cancel and cancel/replace requests.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
37	OrderID	O		Unique identifier for an order assigned by the trading system. OrderID is unique across all trading days, all order books, and all trading members, and it is persistent for the entire life of an order (i.e. it does not change after order modification). OrderID is generated for each order as a concatenation of connection ID + sequential number of a trading day (starting from 1) + sequence number of the message.	int	8 bytes	numeric
11	ClOrdID	R		Unique identifier for an order assigned by the Exchange Member. Exchange Members that submit GTD / GTC orders must ensure uniqueness across days, for example by embedding a date within the ClOrdID value.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
C	Parties	R					

Tag	Field	Req	Conditional	Description	Type	Length	Values
453	**NoPartyIDs**	R		Number of repetitions of PartyID (448), PartyIDSource (447), PartyRole (452), PartyRoleQualifier (2376) groupings.	numingroup	1 byte	Integer
→448	**PartyID**	R		Identification of the party. The short code representing the client, execution decision maker or investment decision maker represented by this block (depending on the PartyRole value).	num	4 bytes	integer 4 - 4.294.967.295 (4 bytes) The following values are reserved for applicable use. Applicable to PartyRole = 3: • 1 = AGGR (an aggregation of multiple client orders) • 2 = PNAL (clients are pending allocation) Applicable to PartyRole = 12: • 3 = NORE (timing and location of the execution determined by the client of the participant)
→447	**PartyIDSource**	R		Used to identify classification source.	char	1	P = Short code identifier
→452	**PartyRole**	R		Identifies the type of PartyID (448). Specifies the role represented by the short code provided in the PartyID tag.		1 byte	For PartyIDSource (447) = P: • 3 = Client ID • 12 = Executing Trader • 122 = Investment Decision Maker
→2376	**PartyRoleQualifier**		Conditionally required for PartyIDSource (447) = P (Short code identifier) and PartyID >=4(Its not present for reserved values: 1,2,3).	Provides further qualification of the PartyRole (452) value.	int	1 byte	22 = Algorithm (applicable to PartyRole values 12 or 122) 23 = Firm or legal entity (LEI) (applicable to PartyRole value 3) 24 = Natural person (applicable to PartyRole values 3, 12, 122)
C	**Instrument**	R					
48	**SecurityID**	R		Security identifier value of SecurityIDSource (22). Numerical (non-ISIN) instrument identifier provided on a daily basis via Market Data reference services.	string	4 chars	integer
22	**SecurityIDSource**	R		Identifies class or source of the SecurityID (48) value.	string	1 char	8 = Exchange symbol
54	**Side**	R		Side of order.	char	1 char	1 = Buy

Tag	Field	Req	Conditional	Description	Type	Length	Values
							2 = Sell
60	**TransactTime**	R		Time the request was initiated by the Exchange Member. This time should be earlier than message SendingTime (52) provided in the header.	UTC timestamp		The following formats are accepted: YYYYMMDD-HH:MM:SS.nnnnnnnnn (UTC) YYYYMMDD-HH:MM:SS.uuuuuu (UTC) YYYYMMDD-HH:MM:SS.mmm (UTC)
C	**StandardTrailer**	R		Trailer used in FIX TP messages			

7.16. Trade Capture Report (AE)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	**StandardHeader**	R		Header used in FIX TP messages			
571	**TradeReportID**	R		Unique identifier of Trade Capture Report. Last character is reserved for exchange-assigned identifiers.	string	21 chars	alphanumeric, 0-9, A-Z, a-z
1003	**TradeID**	O	Provided in case of matched trades.	Identifier assigned to a trade by a matching system.	string	4 bytes	numeric
487	**TradeReportTransType**	R		Identifies Trade Report message transaction type.	int		0 = New 1 = Cancel 2 = Replace
856	**TradeReportType**	R		Type of Trade Report.	int		0 = Submit 1 =Alleged 2 = Accept 3 = Decline 6 = Trade Report Cancel 7 = Trade Break
828	**TrdType**	R		Type of trade.	int		22 = Privately negotiated trade (Cross trade) 38 = Block trade
2667	**AlgorithmicTradeIndicator**	C	Provided in case of ExecType (150) = F (Trade).	Indicates that the trade originates from a computer program or algorithm requiring little-to-no human intervention. In the context of ESMA MiFID II, a trade has to be flagged as "algorithmic" if at least one of the matched orders was submitted by a trading algorithm.	int		0 = Non-algorithmic trade 1 = Algorithmic trade

Tag	Field	Req	Conditional	Description	Type	Length	Values
1123	**TradeHandlingInstr**	R		Specifies how the Trade Capture Report (35=AE) should be handled by the respondent.	char	1 byte	0 = Trade confirmation 1 = Two-party report 3 = One-party report for pass-through
150	**ExecType**	C	Not present for incoming TCR . Present for outgoing TCR's in case it notifies of Trade/Trade Cancel or Trade first leg expiration.	Type of execution being reported.	char	1 byte	C =Expired F = Trade H= Trade Cancel
572	**TradeReportRefID**	C		Identifier used to reference leg for cancellation or in case of reporting matched trades.	string	21 chars	alphanumeric, 0-9, A-Z, a-z
818	**SecondaryTradeReportID**	C	Ignored in incoming first leg of single-sided declaration and in incoming dual-sided declaration.	Secondary identifier of trade capture report assigned by the trading system.	string	21 chars	alphanumeric, 0-9, A-Z, a-z
C	**Instrument**	R					
48	**SecurityID**	R		Security identifier value of SecurityIDSource (22). Numerical (non-ISIN) instrument identifier provided on a daily basis via Market Data reference services.	string	4 bytes	integer
22	**SecurityIDSource**	R		Identifies class or source of the SecurityID (48) value.	string	1 char	8 = exchange symbol
32	**LastQty**	R		Quantity (expressed in number of Lots).	qty	4 bytes	integer
31	**LastPx**	R		Price per unit of quantity (smallest portion of instrument)	price	12 digits + decimal point	Floating point number. 6 digits for integer part, decimal point, 6 digits for decimal part. Negative values and 0 allowed.
60	**TransactTime**	R		Timestamp when the business transaction represented by the message occurred. Time the transaction represented by when this TradeCaptureReport (35=AE) occurred. Execution time of trade. Also describes the time of block trades.	UTC timestamp		The following formats are accepted: YYYYMMDD-HH:MM:SS.nnnnnnnnn (UTC) YYYYMMDD-HH:MM:SS.uuuuuu (UTC) YYYYMMDD-HH:MM:SS.mmm (UTC)
64	**SettlDate**	C	Required in case of block trades for all asset classes except for derivatives (where it is ignored).	Specific date of trade settlement (SettlementDate) in YYYYMMDD format, in line with applicable settlement calendar.	localmktdate		YYYYMMDD (local) In case of block trades (for all asset classes except for derivatives)

Tag	Field	Req	Conditional	Description	Type	Length	Values
							values from T+0 to T+30 are allowed.
C	TrdCapRptSideGrp	R					
552	NoSides	R		Number of Side repeating group instances.	numingroup	1 byte	1 = One Side 2 = Both Sides
→54	Side	R	Required when NoSides (552) > 0.	Side of order.	char	1 char	1 = Buy 2 = Sell
→C	Parties	R					
→453	NoPartyIDs	R		Number of repetitions of PartyID (448), PartyIDSource (447), PartyRole (452), PartyRoleQualifier (2376) groupings.	numingroup	1 byte	integer
→→448	PartyID	R	Required if NoPartyIDs (453) > 0	Identification of the party.	"For PartyIDSource (447) = D : string For PartyIDSource (447) = N : string For PartyIDSource (447) = P : integer"	"For PartyIDSource (447) = D : string For PartyIDSource (447) = N : string For PartyIDSource (447) = P : integer"	"integer For clients, the following values are reserved for applicable use: Applicable to PartyRole value 3: 1 = AGGR (an aggregation of multiple client orders) 2 = PNAL (clients are pending allocation) Applicable to PartyRole value 12: 3 = NORE (timing and location of the execution determined by • the client of the participant)"
→→447	PartyIDSource	R	Required if NoPartyIDs (453) > 0	Used to identify classification source.	char	1 char	D = Proprietary / Custom code N = Legal Entity Identifier (LEI) P = Short code identifier

Tag	Field	Req	Conditional	Description	Type	Length	Values
→→452	PartyRole	R	Required if NoPartyIDs (453) > 0	Identifies the type of PartyID (448). Specifies the role represented by the short code provided in the PartyID tag.	int		"For PartyIDSource (447) = D : 1 = Executing Firm 4 = Clearing Firm 17 = Contra Firm 33 = Interested Party For PartyIDSource (447) = P : 3 = Client ID 12 = Executing Trader • 122 = Investment Decision Maker"
→→2376	PartyRoleQualifier	C	Required if NoPartyIDs (453) > 0 and PartyID >=4(Its not present for reserved values: 1,2,3).	Provides further qualification of the PartyRole (452) value.	int		22 = Algorithm (applicable to PartyRole values 12 or 122) 23 = Firm or legal entity (LEI) (applicable to PartyRole value 3) 24 = Natural person (applicable to PartyRole values 3, 12, 122)
→1	Account	O		Account mnemonic as agreed between buy and sell sides, e.g. broker and institution or investor/intermediary and fund manager.	string	12 or 16 chars	alphanumeric, 0-9, A-Z, a-z or ASCII 33-126
→581	AccountType	O		Type of account associated with the order.	int		1 = Account is carried on customer side of the books 3 = House trader
→58	Text	O		Free format text string for Exchange Member use.	string	18 chars	alphanumeric, 0-9, A-Z, a-z or ASCII 33-126
→20011	FeeStructureID	O		Optional identifier of a fee scheme for billing purposes.	int	1 byte	1 to 99 - for internal purposes for Members 100 to 255 - for internal purposes for WSE (GPW)
→C	TradeReportOrderDetail	O					
→528	OrderCapacity	R		Designates the capacity of the firm placing the order.	char	1 char	A = Agency (mapped to AOTC) P = Principal (mapped to DEAL)

Tag	Field	Req	Conditional	Description	Type	Length	Values
							R = Riskless Principal (mapped to MTCH)
→529	OrderRestrictions	O		Designation required for regular Liquidity Providers (as opposite to Issuer Liquidity Providers). In case of Issuer Liquidity Provider orders tag should be omitted.	char	1 char	5 = Acting as Market Maker or Specialist in the security
→1724	OrderOrigination	O		Designation required for orders submitted via Direct Electronic Access (both DMA or Sponsored Access). In case of regular trading access this tag should be omitted.	int		5 = Order received from a direct access or sponsored access customer
→C	OrderAttributeGrp	O					
→2593	NoOrderAttributes	R		Number of order attribute entries.	Int	1 byte	1
→→2594	OrderAttributeType	R		The type of order attribute.	int	1 byte	2 = Liquidity provision activity order
→→2595	OrderAttributeValue	R		The value associated with the order attribute type specified in OrderAttributeType (2594). As opposite to OrderRestriction tag specifies whether the order poses any type of liquidity provision activity (either regular or issuer).	string	1 char	Y = Yes N = No
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.17. Trade Capture Report Ack (AR)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			
571	TradeReportID	R		Unique identifier of Trade Capture Report. Last character is reserved for exchange-assigned identifiers.	string	21 chars	alphanumeric, 0-9, A-Z, a-z
1003	TradeID	O	Provided in case of matched trades.	Identifier assigned to a trade by a matching system.	string	4 bytes	numeric
487	TradeReportTransType	R		Identifies trade report message transaction type.	int		0 = New 1 = Cancel 2 = Replace
856	TradeReportType	R		Type of trade report.	int		0 = Submit 1 = Alleged 2 = Accept

Tag	Field	Req	Conditional	Description	Type	Length	Values
							3 = Decline 6 = Trade Report Cancel 14 = Alleged Trade Report Cancel
828	**TrdType**	R		Type of trade.	int		22 = Privately negotiated trade (Cross trade) 38 = Block trade
1123	**TradeHandlingInstr**	R		Specifies how the Trade Capture Report (35=AE) should be handled by the respondent.	char	1 char	0 = Trade confirmation 1 = Two-party report 3 = One-party report for pass-through
572	**TradeReportRefID**	O		Reference identifier used with Cancel and Replace transaction types. The TradeReportID (571) that is being referenced for trade correction or cancelation.	string	21 chars	alphanumeric, 0-9, A-Z, a-z
939	**TrdRptStatus**	R		Status of trade report.	int		0 = Accepted 1 = Rejected
751	**TradeReportRejectReason**	C	Provided in case of TrdRptStatus (939) = 1 (Rejected).	Code to identify reason for trade report rejection.	int		Please see FIX Rejection Codes document for the list of available codes with their descriptions.
818	**SecondaryTradeReportID**	C	Required for tradeReportType= Alleged or Submit	Unique system identifier of the trade.	string	20 chars	numeric
32	**LastQty**	R		Quantity (expressed in number of Lots).	qty	4 bytes	integer
31	**LastPx**	R		Price per unit of quantity (smallest portion of instrument)	price	12 digits + decimal point	Floating point number. 6 digits for integer part, decimal point, 6 digits for decimal part. Negative values and 0 allowed.
C	**Instrument**	R					
48	**SecurityID**	R		Security identifier value of SecurityIDSource (22). Numerical (non-ISIN) instrument identifier provided on a daily basis via Market Data reference services.	string	4 bytes	integer
22	**SecurityIDSource**	R		Identifies class or source of the SecurityID (48) value.	string	1 char	8 = exchange symbol
15	**Currency**	R	It is recommended	Identifies currency used for price (i.e. instrument trading currency). Absence of this	currency	3 bytes	

Tag	Field	Req	Conditional	Description	Type	Length	Values
			that systems provide the currency value whenever possible.	field is interpreted as the default for the security.			
60	**TransactTime**	R		Timestamp when the business event represented by the message occurred. This time should be earlier than message SendingTime (52) provided in the header.	UTC timestamp		YYYYMMDD-HH:MM:SS.nnnnnnnnn (UTC)
C	**TrdCapRptAckSideGrp**	R					
552	**NoSides**	R		Number of Side repeating group instances.	numingroup	1 byte	1 = One Side 2 = Both Sides
→54	**Side**	R		Side.	char	1 char	1 = Buy 2 = Sell
→C	**Parties**	R					
→453	**NoPartyIDs**	R		Number of repetitions of PartyID (448), PartyIDSource (447), PartyRole (452), PartyRoleQualifier (2376) groupings.	numingroup	1 byte	integer
→→448	**PartyID**	R	Required if NoPartyIDs (453) > 0.	Identification of the party. The short code representing the client, execution decision maker or investment decision maker represented by this block (depending on the PartyRole value).	"For PartyIDSource (447) = D : string For PartyIDSource (447) = N : string For PartyIDSource (447) = P : integer"	"For PartyIDSource (447) = D and a) PartyRole (452) = 1, 4, 17 : 4 bytes b) PartyRole (452) = 33 : 8 bytes For PartyIDSource (447) = N : 20 bytes For PartyIDSource (447) = P : 4 bytes"	"integer For clients, the following values are reserved for applicable use: Applicable to PartyRole value 3: 1 = AGGR (an aggregation of multiple client orders) 2 = PNAL (clients are pending allocation) Applicable to PartyRole value 12: 3 = NORE (timing and location of the execution determined by • the client of the participant)"

Tag	Field	Req	Conditional	Description	Type	Length	Values
→→447	PartyIDSource	R	Required if NoPartyIDs (453) > 0.	Used to identify classification source.	char	1 byte	D = Proprietary / Custom code N = Legal Entity Identifier (LEI) P = Short code identifier
→→452	PartyRole	R	Required if NoPartyIDs (453) > 0.	Identifies the type of PartyID (448). Specifies the role represented by the short code provided in the PartyID tag.	int		For PartyIDSource (447) = D: • 1 = Executing Firm • 4 = Clearing Firm • 17 = Contra Firm • 33 = Interested Party For PartyIDSource (447) = P: • 3 = Client ID • 12 = Executing Trader • 122 = Investment Decision Maker
→→2376	PartyRoleQualifier	R	Required if NoPartyIDs (453) > 0. Conditionally required for PartyIDSource (447) = P (Short code identifier) and PartyID >=4(Its not present for reserved values: 1,2,3).	Provides further qualification of the PartyRole (452) value.	int		22 = Algorithm (applicable to PartyRole values 12 or 122) 23 = Firm or legal entity (LEI) (applicable to PartyRole value 3) 24 = Natural person (applicable to PartyRole values 3, 12, 122)
→1	Account	O		Account provided by the Exchange Member for clearing purposes.	string	12 or 16 chars	alphanumeric, 0-9, A-Z, a-z or ASCII 33-126
→581	AccountType	O		Type of account associated with the order.	int		1 = Account is carried on customer side of the books 3 = House trader
→58	Text	O		Free format text string.	string	18 chars	alphanumeric, 0-9, A-Z, a-z or ASCII 33-126
→20011	FeeStructureID	O		Optional identifier of a fee scheme for billing purposes.	int	1 byte	1 to 99 - for internal purposes for Members 100 to 255 - for internal purposes for WSE (GPW)
→C	TradeReportOrderDetail	R					

Tag	Field	Req	Conditional	Description	Type	Length	Values
→528	OrderCapacity	R		Designates the capacity of the firm placing the order.	char	1 char	A = Agency (mapped to AOTC) P = Principal (mapped to DEAL) R = Riskless Principal (mapped to MTCH)
→529	OrderRestrictions	O		Designation required for regular Liquidity Providers (as opposite to Issuer Liquidity Providers). In case of Issuer Liquidity Provider orders tag should be omitted.	char	1 char	5 - Acting as Market Maker or Specialist in the security
→1724	OrderOrigination	O		Designation required for orders submitted via Direct Electronic Access (both DMA or Sponsored Access). In case of regular trading access this tag should be omitted.	int		5 - Order received from a direct access or sponsored access customer
→C	OrderAttributeGrp	O					
→2593	NoOrderAttributes	R		Number of order attribute entries.	int	1 byte	
→→2594	OrderAttributeType	R		The type of order attribute.	int	1 byte	2 - Liquidity provision activity order
→→2595	OrderAttributeValue	R		The value associated with the order attribute type specified in OrderAttributeType (2594). As opposite to OrderRestriction tag specifies whether the order poses any type of liquidity provision activity (either regular or issuer).	string	1 char	Y = Yes N = No
64	SettlDate	C	Ignored in case of privately negotiated trades (i.e. cross trades) for all asset classes. Required in case of block trades for all asset classes except for derivatives (where it is ignored).	Specific date of trade settlement (SettlDate) in YYYYMMDD format, in line with applicable settlement calendar.	localmktdate		YYYYMMDD
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.18. Mass Quote (i)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			
117	QuoteID	R		Unique identifier for quote.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
C	Parties	R					
453	NoPartyIDs	R		Number of PartyID (448), PartyIDSource (447), PartyRole (452) and PartyRoleQualifier (2376) entries.	numingroup	1 byte	integer
→448	PartyID	R	Required if NoPartyIDs (453) > 0	Identification of the party.	"For PartyIDSource (447) = D : string For PartyIDSource (447) = N : string For PartyIDSource (447) = P : integer"	"For PartyIDSource (447) = D and a) PartyRole (452) = 4 : 4 bytes b) PartyRole (452) = 33 : 8 bytes For PartyIDSource (447) = N : 20 bytes For PartyIDSource (447) = P : 4 bytes"	"integer For clients, the following values are reserved for applicable use: Applicable to PartyRole value 3: 1 = AGGR (an aggregation of multiple client orders) 2 = PNAL (clients are pending allocation) Applicable to PartyRole value 12: 3 = NORE (timing and location of the execution determined by • the client of the participant)"
→447	PartyIDSource	R	Required if NoPartyIDs (453) > 0	Used to identify classification source.	char	1 char	D - Proprietary / Custom code N - Legal Entity Identifier (LEI) P - Short code identifier
→452	PartyRole	R	Required if NoPartyIDs (453) > 0	Identifies the type of PartyID (448).	int		For PartyIDSource (447) - D: • 4 - Clearing Firm • 33 = Interested Party For PartyIDSource (447) - P:

Tag	Field	Req	Conditional	Description	Type	Length	Values
							• 3 - Client ID • 12 - Executing Trader • 122 - Investment Decision Maker
→2376	PartyRoleQualifier	R	Required if NoPartyIDs (453) > 0 and PartyID >=4(Its not present for reserved values: 1,2,3).	Used to further qualify the value of PartyRole (452).			For PartIDSource (447) - P: 22 - Algorithm (applicable to PartyRole values 12 or 122) 23 - Firm or legal entity (LEI) (applicable to PartyRole value 3) 24 - Natural person (applicable to PartyRole values 3, 12, 122)
1	Account	O		Account mnemonic as agreed between buy and sell sides, e.g. broker and institution or investor/intermediary and fund manager.	string	16 chars	alphanumeric, 0-9, A-Z, a-z or ASCII 33-126
581	AccountType	O		Type of account associated with the order.	int		1 - Account is carried on customer side of the books 3 - House trader
528	OrderCapacity	R		Designates the capacity of the firm placing the order.	char	1 char	A - Agency (mapped to AOTC) P - Principal (mapped to DEAL) R - Riskless Principal (mapped to MTCH)
529	OrderRestrictions	O		Restrictions associated with an order.	char	1 char	5 - Acting as Market Maker or Specialist in the security
1724	OrderOrigination	O		Identifies the origin of the order.	int		5 - Order received from a direct access or sponsored access customer
60	TransactTime	R		Timestamp when the business transaction represented by the message occurred.	UTC timestamp		For incoming messages all listed below formats are accepted: YYYYMMDD-HH:MM:SS.nnnnnnnnn YYYYMMDD-HH:MM:SS.uuuuuu YYYYMMDD-HH:MM:SS.sss
58	Text	O		Free format text string.	string	18 chars	alphanumeric, 0-9, A-Z, a-z or ASCII 33-126
20011	FeeStructureID	O		Optional identifier of a fee scheme for billing purposes.	int	1 byte	1 to 99 - for internal purposes for Members 100 to 255 - for internal purposes for WSE (GPW)
C	QuoteSetGrp	R					

Tag	Field	Req	Conditional	Description	Type	Length	Values
296	NoQuoteSets	R		The number of sets of quotes in the message.	int	1 byte	Currently always equal to 1.
→302	QuoteSetID	R	Required if NoQuoteSets > 0	Sequential number for the Quote Set. For a given QuoteID - assumed to start at 1. Must be the first field in the repeating group.	string	1 byte	Currently always equal to 1.
→304	TotNoQuoteEntries	R		Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries in each message that has repeating quotes that are part of the same quote set.	int	1 byte	1-30
→C	QuoteEntryGrp	R					
295	NoQuoteEntries	R		The number of quotes for this Symbol (instrument) (QuoteSet) that follow in this message.	int	1 byte	1-30
→→299	QuoteEntryID	R		Unique identifier for a quote. The QuoteEntryID stays with the quote as a static identifier even if the quote is updated.	string	1 byte	1-30
→→C	Instrument	R					
→→48	SecurityID	R		Security identifier value of SecurityIDSource (22) type (e.g. ISIN, exchange symbol, etc). Requires SecurityIDSource.	num	4 bytes	integer
→→22	SecurityIDSource	R		Identifies class or source of the SecurityID (48) value.	char	1 char	8 - exchange symbol
→→132	BidPx	R		Bid price per unit of quantity (smallest portion of instrument)	int	12 digits + decimal point	Floating point number. 6 digits for integer part, decimal point, 6 digits for decimal part. Negative values and 0 allowed.
→→133	OfferPx	R		Offer price per unit of quantity (smallest portion of instrument).	int	12 digits + decimal point	Floating point number. 6 digits for integer part, decimal point, 6 digits for decimal part. Negative values and 0 allowed.

Tag	Field	Req	Conditional	Description	Type	Length	Values
→→134	BidSize	R	0 for quote cancellation.	Quantity of bid.	int	4 bytes	integer
→→135	OfferSize	R	0 for quote cancellation.	Quantity of offer.	int	4 bytes	integer
2362	SelfMatchPreventionID	O		Identifies orders/quotes that should not be matched against each other if both of them contain the same SelfMatchPreventionID (2362) for the same instrument and were submitted by the same Trading Member. The only option available in Mass Quote message is "cancel passive order".	int	1 byte	integer
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.19. Mass Quote Ack(B)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			
117	QuoteID	O	Required when acknowledgment is in response to a Mass Quote message.	Unique identifier for quote. Maps to QuoteID (117) of a Mass Quote.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
297	QuoteStatus	R		Status of the mass quote acknowledgement.	int		0 - Accepted 5 - Rejected
300	QuoteRejectReason	O		Reason Quote was rejected.	int		2 - Exchange (security) closed 4 - Too late to enter 6 - Duplicate quote 99 - Other
C	Parties	R					
453	NoPartyIDs	R		Number of PartyID (448), PartyIDSource (447), PartyRole (452) and PartyRoleQualifier (2376) entries.	numingroup	1 byte	integer

Tag	Field	Req	Conditional	Description	Type	Length	Values
→448	PartyID	R	Required if NoPartyIDs (453) > 0	Identification of the party.	"For PartyIDSource (447) = D : string For PartyIDSource (447) = N : string For PartyIDSource (447) = P : integer"	"For PartyIDSource (447) = D and a) PartyRole (452) = 4 : 4 bytes b) PartyRole (452) = 33 : 8 bytes For PartyIDSource (447) = N : 20 bytes For PartyIDSource (447) = P : 4 bytes"	"integer For clients, the following values are reserved for applicable use: Applicable to PartyRole value 3: 1 = AGGR (an aggregation of multiple client orders) 2 = PNAL (clients are pending allocation) Applicable to PartyRole value 12: 3 = NORE (timing and location of the execution determined by • the client of the participant)"
→447	PartyIDSource	R	Required if NoPartyIDs (453) > 0	Used to identify classification source.	char	1 char	D - Proprietary / Custom code N - Legal Entity Identifier (LEI) P - Short code identifier
→452	PartyRole	R	Required if NoPartyIDs (453) > 0	Identifies the type of PartyID (448).	int		For PartyIDSource (447) - D: • 4 - Clearing Firm • 33 = Interested PArty For PartyIDSource (447) - P: • 3 - Client ID • 12 - Executing Trader • 122 - Investment Decision Maker
→2376	PartyRoleQualifier	R	Required if NoPartyIDs (453) > 0	Used to further qualify the value of PartyRole (452) and PartyID >=4(Its not present for reserved values: 1,2,3).			"For PartIDSource (447) = P : 0 = None (only valid for reserved short codes) 22 = Algorithm (applicable to PartyRole values 12 or 122) 23 = Firm or legal entity (LEI) (applicable to PartyRole value 3) • 24 = Natural person (applicable to PartyRole values 3, 12, 122)"

Tag	Field	Req	Conditional	Description	Type	Length	Values
1	Account	O		Account mnemonic as agreed between buy and sell sides, e.g. broker and institution or investor/intermediary and fund manager.	string	16 chars	alphanumeric, 0-9, A-Z, a-z or ASCII 33-126
581	AccountType	O		Type of account associated with the order.	int		1 - Account is carried on customer side of the books 3 - House trader
60	TransactTime	R		Timestamp when the business transaction represented by the message occurred.	UTC timestamp		For outcoming messages: TransactTime format string to YYYYMMDD-HH:MM:SS.nnnnnnnnn (UTC)
528	OrderCapacity	R		Designates the capacity of the firm placing the order.	char	1 char	A - Agency (mapped to AOTC) P - Principal (mapped to DEAL) R - Riskless Principal (mapped to MTCH)
529	OrderRestrictions	O		Restrictions associated with an order.	char	1 char	5 - Acting as Market Maker or Specialist in the security
1724	OrderOrigination	O		Identifies the origin of the order.	int		5 - Order received from a direct access or sponsored access customer
58	Text	O		Free format text string.	string	18 chars	alphanumeric, 0-9, A-Z, a-z or ASCII 33-126
20011	FeeStructureID	O		Optional identifier of a fee scheme for billing purposes.	int	1 byte	1 to 99 - for internal purposes for Members 100 to 255 - for internal purposes for WSE (GPW)
C	QuoteSetAckGrp	R					
296	NoQuoteSets	R		The number of sets of quotes in the message.	int	1 byte	Currently always equal to 1.
→302	QuoteSetID	R	Required if NoQuoteSets > 0	Sequential number for the Quote Set. For a given QuoteID - assumed to start at 1. Must be the first field in the repeating group.	int	1 byte	Currently always equal to 1.
→304	TotNoQuoteEntries	R	Required if NoQuoteEntries > 0	Total number of quotes for the quote set across all messages. Should be the sum of all NoQuoteEntries in each message that has repeating quotes that are part of the same quote set.	int	1 byte	
→1168	TotNoClxdQuotes	R		Total number of quotes canceled for the quote set across all messages.	int	1 byte	
→1169	TotNoAccQuotes	R		Total number of quotes accepted for the quote set across all messages.	int	1 byte	

Tag	Field	Req	Conditional	Description	Type	Length	Values
→1170	TotNoRejQuotes	R		Total number of quotes rejected for the quote set across all messages.	int	1 byte	
→C	QuoteEntryAckGrp	O					
295	NoQuoteEntries	R		The number of quotes for this Symbol (instrument) (QuoteSet) that follow in this message.	int	1 byte	
→→299	QuoteEntryID	R	Required if NoQuoteEntries > 0.	Uniquely identifies the quote across the complete set of all quotes for a given quote provider. First field in repeating group.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
→→C	Instrument	R					
→→48	SecurityID	R		Security identifier value of SecurityIDSource (22) type (e.g. ISIN, exchange symbol, etc). Requires SecurityIDSource.	string	4 bytes	integer
→→22	SecurityIDSource	R		Identifies class or source of the SecurityID (48) value.	char	1 char	4 - Exchange symbol
→→1747	BidQuoteID	O		Marketplace assigned quote identifier for the bid side. Can be used to indicate priority.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
→→1748	OfferQuoteID	O		Marketplace assigned quote identifier for the offer side. Can be used to indicate priority.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
→→132	BidPx	R		Bid price per unit of quantity (smallest portion of instrument).	int	12 digits + decimal points	Floating point number. 6 digits for integer part, decimal point, 6 digits for decimal part. Negative values and 0 allowed.
→→133	OfferPx	R		Offer price per unit of quantity (smallest portion of instrument).	int	12 digits + decimal point	Floating point number. 6 digits for integer part, decimal point, 6 digits for decimal part. Negative values and 0 allowed.
→→134	BidSize	R		Bid quantity (expressed in number of Lots).	int	4 bytes	integer
→→135	OfferSize	R		Offer quantity (expressed in number of Lots).	int	4 bytes	integer
→→1167	QuoteEntryStatus	O		Identifies the status of an individual quote. See also QuoteStatus (297) which is used for single Quotes.	int		0 - Accepted 5 - Rejected
→→368	QuoteEntryRejectReason	O		Reason quote entry was rejected.	int		18 - Invalid price increment 1025 - Order quantity must be greater than minimum quantity 1026 - Order quantity must be lower than maximum quantity 1027 - Order price must be greater than minimum price 1028 - Order price must be lower than maximum price

Tag	Field	Req	Conditional	Description	Type	Length	Values
							1029 - Order price must be nonzero 1030 - Order value must be greater than minimum value 1031 - Order value must be lower than maximum value 1037 - Price below low collar 1038 - Price above high collar 1057 - Firm not authorized to buy and sell the instrument 1058 - Firm not authorized to buy the instrument 1059 - Firm not authorized to sell the instrument 1201 - Unknown symbol 1203 - Instrument closed 1208 - Invalid bid 1209 - Not authorized to quote security 99 - Other
2362	SelfMatchPreventionID	O		Identifies orders/quotes that should not be matched against each other if both of them contain the same SelfMatchPreventionID (2362) for the same instrument and were submitted by the same Trading Member. The only option available in Mass Quote message is "cancel passive order".	int	1 byte	integer
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.20. Request For Execution (UE)

Tag	Field	Req	Description	Type	Length	Values
C	StandardHeader	R	Header used in FIX TP messages			
C	Instrument	R				
48	SecurityID	R	Security identifier value of SecurityIDSource (22) type (e.g. ISIN, exchange symbol, etc). Requires SecurityIDSource.	int	4 bytes	integer
22	SecurityIDSource	R	Identifies class or source of the SecurityID (48) value.	char	1 char	8 - exchange symbol
60	TransactTime	R	Timestamp when the business action represented by the message occurred.	UTC timestamp		For outcoming messages: TransactTime format string to YYYYMMDD-HH:MM:SS.nnnnnnnnn (UTC)
20002	RequestForExecutionReason	R	Specifies whether Request For Execution message was triggered by the incoming client order hitting passive MM quote, or the aggressive MM quote hitting a resting client order.	int		1 - passive MM quote 2 - aggressive MM quote

Tag	Field	Req	Description	Type	Length	Values
C	StandardTrailer	R	Trailer used in FIX TP messages			

7.21. Market Maker Command (UB)

Tag	Field	Req	Description	Type	Length	Values
C	StandardHeader	R	Header used in FIX TP messages			
20003	MarketMakerCommandID	R	Unique identifier for Market Maker Command request assigned by the market maker.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
C	Instrument	R				
48	SecurityID	R	Security identifier value of SecurityIDSource (22) type (e.g. ISIN, exchange symbol, etc.). Requires SecurityIDSource.	int	4 bytes	integer
22	SecurityIDSource	R	Identifies class or source of the SecurityID (48) value.	char	1 char	8 - exchange symbol
60	TransactTime	R	Timestamp when the business action represented by the message occurred.	UTC timestamp		For incoming messages all listed below formats are accepted: YYYYMMDD-HH:MM:SS.nnnnnnnnn YYYYMMDD-HH:MM:SS.uuuuuu YYYYMMDD-HH:MM:SS.sss
20009	MarketMakerCommandAction	R	Specifies the action on the instrument to trigger.	int	1 byte	1 - Buy Only State 2 - Knock-Out 3 - Revoke Knock-Out
C	StandardTrailer	R	Trailer used in FIX TP messages			

7.22. Market Maker Command Ack (UC)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			
20003	MarketMakerCommandID	R		Unique identifier for Market Maker Command request assigned by the market maker.	String	20 chars	alphanumeric, 0-9, A-Z, a-z
C	Instrument	R					

Tag	Field	Req	Conditional	Description	Type	Length	Values
48	SecurityID	R		Security identifier value of SecurityIDSource (22) type (e.g. ISIN, exchange symbol, etc). Requires SecurityIDSource.	int	4 bytes	Integer
22	SecurityIDSource	R		Identifies class or source of the SecurityID (48) value.	char	1 char	8 - exchange symbol
60	TransactTime	R		Timestamp when the business action represented by the message occurred.	UTC timestamp		For outcoming messages: TransactTime format string to YYYYMMDD-HH:MM:SS.nnnnnnnnn (UTC)
20004	MarketMakerCommandResult	C	Provided in case of MMCommandStatus (20004) - 2 (Rejected).	Informs whether Market Maker Command was successfully completed or not.	char	1 char	1 -Success 2 - FailureRejected
20005	MarketMakerCommandRejectionCode	C	Provided in case of MMCommandAction (20003) - 2 (Knock-Out) or 3 (Revoke Knock-Out).	Reason for rejecting Market Maker Command message.	int	4 bytes	99 - Other 1001 - UnknownInstrument 3002 - ExchangeClosed 3006 - DuplicateRequestId 3009 - FirmNotAuthorizedToQuoteInstrument 3020 - CommandNotAllowedInCurrentState
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.23. Order Mass Cancel Report (R)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			
11	ClOrdID	R		Unique identifier for an Order Mass Cancel Request assigned by the Trading Member.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
1369	MassActionReportID	R		Unique Identifier for the Order Mass Cancel Report assigned by the recipient of the Order Mass Cancel Request.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
530	MassCancelRequestType	R		Order Mass Cancel Request Type accepted by System.	char	1 char	1 - Cancel orders for a security 7 - Cancel all orders 9 - Cancel orders for a market segment
531	MassCancelResponse	R		Specifies the action taken by counterparty order handling system as a result of the Order Mass Cancel Request.	char	1 char	0 - Cancel request rejected 1 - Cancel orders for a security 7 - Cancel all orders

Tag	Field	Req	Conditional	Description	Type	Length	Values
							9 - Cancel orders for a market segment
532	MassCancelRejectReason	C	Required if MassCancelResponse (531) - 0.	Indicates why Order Mass Cancel Request was rejected.	int	1 byte	1 - Invalid or unknown security 8 - Invalid or unknown market segment 99 - Other
C	Parties	R		Total number of orders affected by the Order Mass Cancel Request (q).	int	4 bytes	
453	NoPartyIDs	R		Number of repetitions of PartyID (448), PartyIDSource (447), PartyRole (452), PartyRoleQualifier (2376) groupings.	int		always 1
→→448	PartyID	R	Required if NoPartyIDs (453) > 0	Identification of the party. The short code representing the execution decision maker represented by this block.	int	4 bytes	integer 4 - 4.294.967.295 (4 bytes)
→→447	PartyIDSource	R	Required if NoPartyIDs (453) > 0	Used to identify classification source.			P - Short code identifier
→→452	PartyRole	R	Required if NoPartyIDs (453) > 0	Identifies the type of PartyID (448). Specifies the role represented by the short code provided in the PartyID tag.			For PartyIDSource (447) - P: 12 - Executing Trader
→→2376	PartyRoleQualifier	R	Required if NoPartyIDs (453) > 0	Provides further qualification of the PartyRole (452) value.			22 - Algorithm (applicable to PartyRole value 12) 24 - Natural person (applicable to PartyRole value 12)
C	TargetParties	O					
1461	NoTargetPartyIDs	O		Number of TargetPartyID (1462), TargetPartyIDSource (1463), TargetPartyRole (1464) and TargetPartyRoleQualifier (1818) entries.	numin group	1 byte	always 1
→→1462	TargetPartyID	C	Required if NoTargetPartyIDs (1461) > 0	Used to identify the party targeted for the action specified in the message.	int	4 bytes	integer
→→1463	TargetPartyIDSource	C	Required if NoTargetPartyIDs (1461) > 0	PartyIDSource value within an target party repeating group. Same values as PartyIDSource (447).	char	1 char	D - Proprietary / Custom code P - Short code identifier
→→1464	TargetPartyRole	C	Required if NoTargetPartyIDs (1461) > 0	PartyRole value within an target party repeating group. Same values as PartyRole (452).	int		For PartyIDSource (447) - D: • 35 - Liquidity provider • 54 - Sender Location • 66 - Market Maker For PartyIDSource (447) - P:

Tag	Field	Req	Conditional	Description	Type	Length	Values
							• 3 - Client ID • 12 - Executing Trader • 122 - Investment Decision Maker
C	Instrument	C					
48	SecurityID	C	Required for MassCancelRequestType - 1 (Cancel orders for a security).	Security identifier value of SecurityIDSource (22) type (e.g. ISIN, exchange symbol, etc). Requires SecurityIDSource.	int	4 bytes	integer
22	SecurityIDSource	C	Required for MassCancelRequestType - 1 (Cancel orders for a security).	Identifies class or source of the SecurityID (48) value.	char	1 char	8 - exchange symbol
1300	MarketSegmentID	C	Required for MassCancelRequestType - 9 (Cancel orders for a market segment).	Identifies the market segment.	string	2 bytes	alphanumeric, 0-9, A-Z
60	TransactTime	R		Time this order request was initiated/released by the trader, trading system, or intermediary.	UTC timestamp	For outcoming messages: TransactTime format string to YYYYMMDD-HH:MM:SS.nnnnnnnnn (UTC)	
C	StandardTrailer	R		Trailer used in FIX TP messages			

7.24. Order Mass Cancel Request (q)

Tag	Field	Req	Conditional	Description	Type	Length	Values
C	StandardHeader	R		Header used in FIX TP messages			

Tag	Field	Req	Conditional	Description	Type	Length	Values
11	ClOrdID	R		Unique identifier for an Order Mass Cancel Request assigned by Trading Member.	string	20 chars	alphanumeric, 0-9, A-Z, a-z
530	MassCancelRequestType	R		Specifies scope of Order Mass Cancel Request.	char	1 char	1 - Cancel orders for a security 7 - Cancel all orders 9 - Cancel orders for a market segment
C	Parties	R					
453	NoPartyIDs	R		Number of repetitions of PartyID (448), PartyIDSource (447), PartyRole (452), PartyRoleQualifier (2376) groupings.	int		always 1
→448	PartyID	R	Required if NoPartyIDs (453) > 0	Identification of the party. The short code representing the execution decision maker represented by this block.	int	4 bytes	integer 4 - 4.294.967.295 (4 bytes)
→447	PartyIDSource	R	Required if NoPartyIDs (453) > 0	Used to identify classification source.			P - Short code identifier
→452	PartyRole	R	Required if NoPartyIDs (453) > 0	Identifies the type of PartyID (448). Specifies the role represented by the short code provided in the PartyID tag.			For PartyIDSource (447) - P: • 12 - Executing Trader
→2376	PartyRoleQualifier	R	Required if NoPartyIDs (453) > 0	Provides further qualification of the PartyRole (452) value.			22 - Algorithm (applicable to PartyRole value 12) 24 - Natural person (applicable to PartyRole value 12)
C	TargetParties	O					
1461	NoTargetPartyIDs	O		Number of TargetPartyID (1462), TargetPartyIDSource (1463), TargetPartyRole (1464) and TargetPartyRoleQualifier (1818) entries.	int		always 1
→1462	TargetPartyID	C	Required if NoTargetPartyIDs (1461) > 0	Used to identify the party targeted for the action specified in the message.	int	4 bytes	Always 1
→1463	TargetPartyIDSource	C	Required if NoTargetPartyIDs (1461) > 0	PartyIDSource value within an target party repeating group. Same values as PartyIDSource (447).	int	4 bytes	D = Proprietary / Custom code P = Short code identifier
→1464	TargetPartyRole	C	Required if NoTargetPartyIDs (1461) > 0	PartyRole value within an target party repeating group.			For PartyIDSource (447) - D: • 35 - Liquidity Provider • 54 - Sender Location • 66 - Market Maker For PartyIDSource (447) - P: • 3 - Client ID

Tag	Field	Req	Conditional	Description	Type	Length	Values
							• 12 - Executing Trader • 122 - Investment Decision Maker
C	**Instrument**	C					
48	**SecurityID**	C	Required for MassCancelRequestType - 1 (Cancel orders for a security).	Security identifier value of SecurityIDSource (22) type (e.g. ISIN, exchange symbol, etc). Requires SecurityIDSource.	int	4 bytes	
22	**SecurityIDSource**	C	Required for MassCancelRequestType - 1 (Cancel orders for a security).	Identifies class or source of the SecurityID (48) value.	char	1 char	
1300	**MarketSegmentID**	C	Required for MassCancelRequestType - 9 (Cancel orders for a market segment).	Identifies the Market Segment.	string	2 bytes	
60	**TransactTime**	R		Time this order request was initiated/released by the trader, trading system, or intermediary.	UTC timestamp		For incoming messages all listed below formats are accepted: YYYYMMDD-HH:MM:SS.nnnnnnnnn YYYYMMDD-HH:MM:SS.uuuuuu YYYYMMDD-HH:MM:SS.sss
C	**SecurityID**	C	Required for MassCancelRequestType - 1 (Cancel orders for a security).	Security identifier value of SecurityIDSource (22) type (e.g. ISIN, exchange symbol, etc). Requires SecurityIDSource.	int	4 bytes	

7.25. BidOfferUpdate (UU)

Tag	Field	Req	Description	Type	Length	Values
C	**StandardHeader**	R	Header used in FIX TP messages			
48	**SecurityID**	R	Security identifier value of SecurityIDSource (22) type (e.g. ISIN, exchange symbol, etc). Requires SecurityIDSource.	int	4 bytes	
22	**SecurityIDSource**	R	Identifies class or source of the SecurityID (48) value.	char	1 char	8 - exchange symbol
3005	**UpdateType**	R	Identifies the a type of the BidOfferUpdate message. UU messages are sent to SEL site only for IPO and to BUY site only for Tender offer.	int	1 bytes	1 - IPO 2 - Tender offer
1749	**TotalBidSize**	R	Specifies the total bid size.	qty	4 bytes	integer

Tag	Field	Req	Description	Type	Length	Values
1750	**TotalOfferSize**	R	Specifies the total offer size.	qty	4 bytes	integer
3003	**BidOrders**	R	Number of bid orders.	int	4 bytes	integer
3004	**OfferOrders**	R	Number of offer orders.	int	4 bytes	integer
C	**StandardTrailer**	R	Trailer used in FIX TP messages			

8. Message Kinematics

Scope:

The purpose of this appendix is to present the message flow within the FIX communication protocol, as used in GPW WATS. The appendix does not include all available message flows in GPW WATS, but only those selected for presentation purposes.

Remarks:

Due to ensuring transparency of message flow diagrams, the scope of presented fields is limited to basic components and does not include the complete set of all fields that a given message may contain. The scope of the fields for the same messages may also vary depending on the specific example described in the diagram. The complete range of fields in the messages is available in the description of each individual message within the protocol.

Actors:

For the purpose of the presentations, the following actors have been utilized in the context of kinematics:

FIX Trading Client Application – Exchange Member authorized to submit buy and sell orders, communicating with the system represented by the FIX Order Gateway.

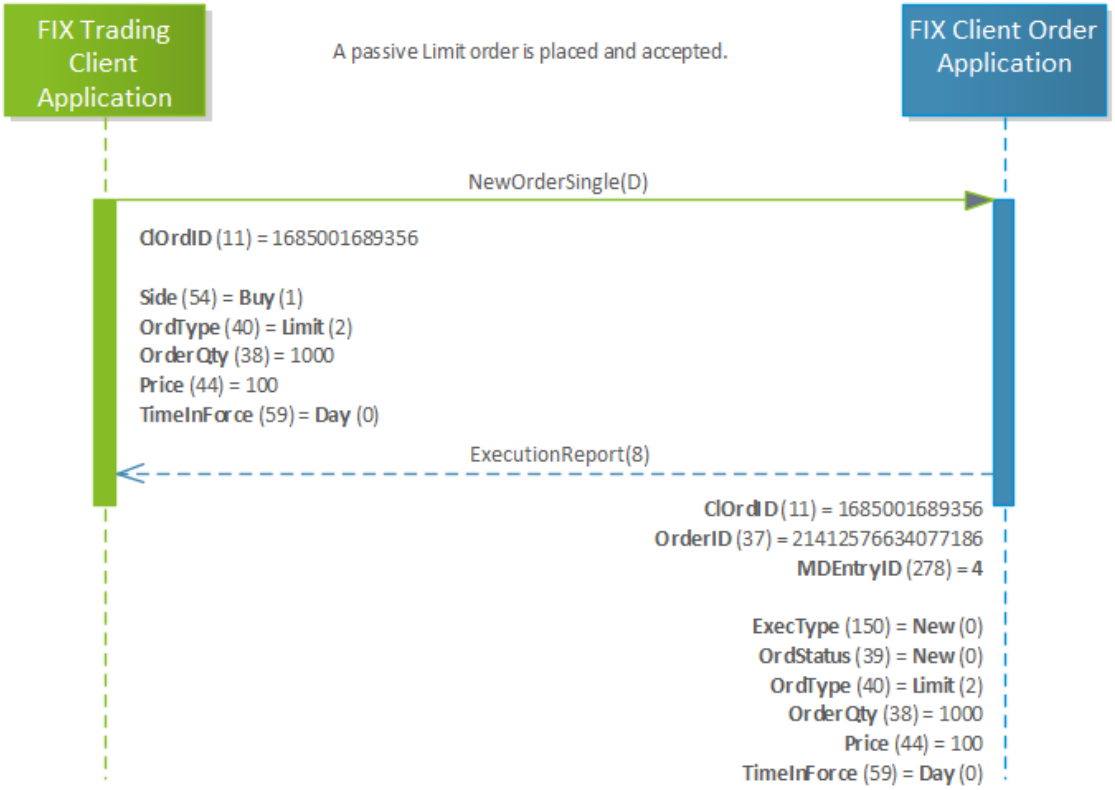
FIX Order Gateway – System that communicates with the FIX Protocol Client Application.

Note:

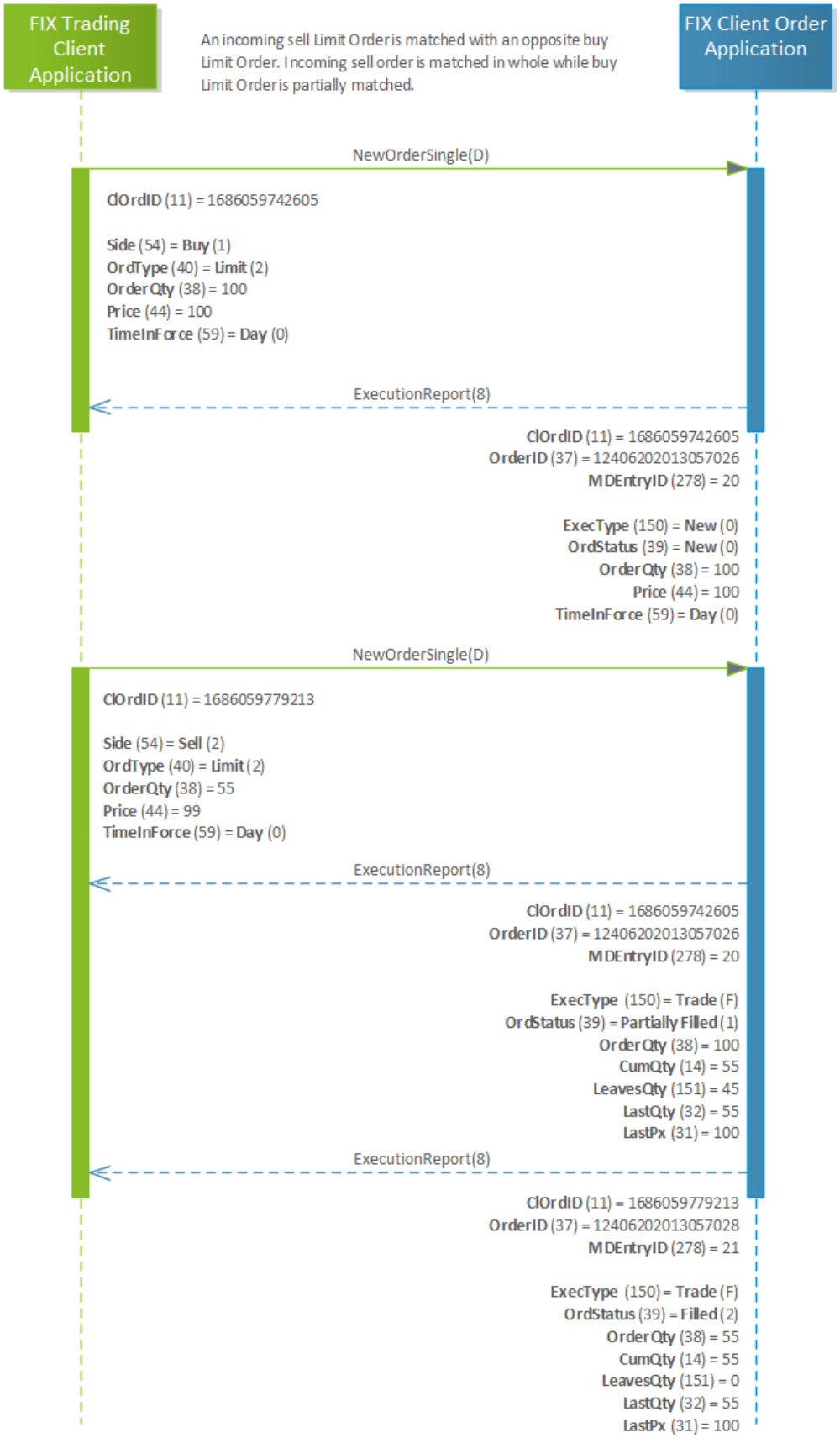
Presentation of simultaneous communication between multiple clients and Gateway was omitted. As a result, the FIX Trading Client Application can either represent a single Exchange Member or simulate the exchange of messages between two different Exchange Members including buying, selling and conducting transactions.

8.1. CLOB

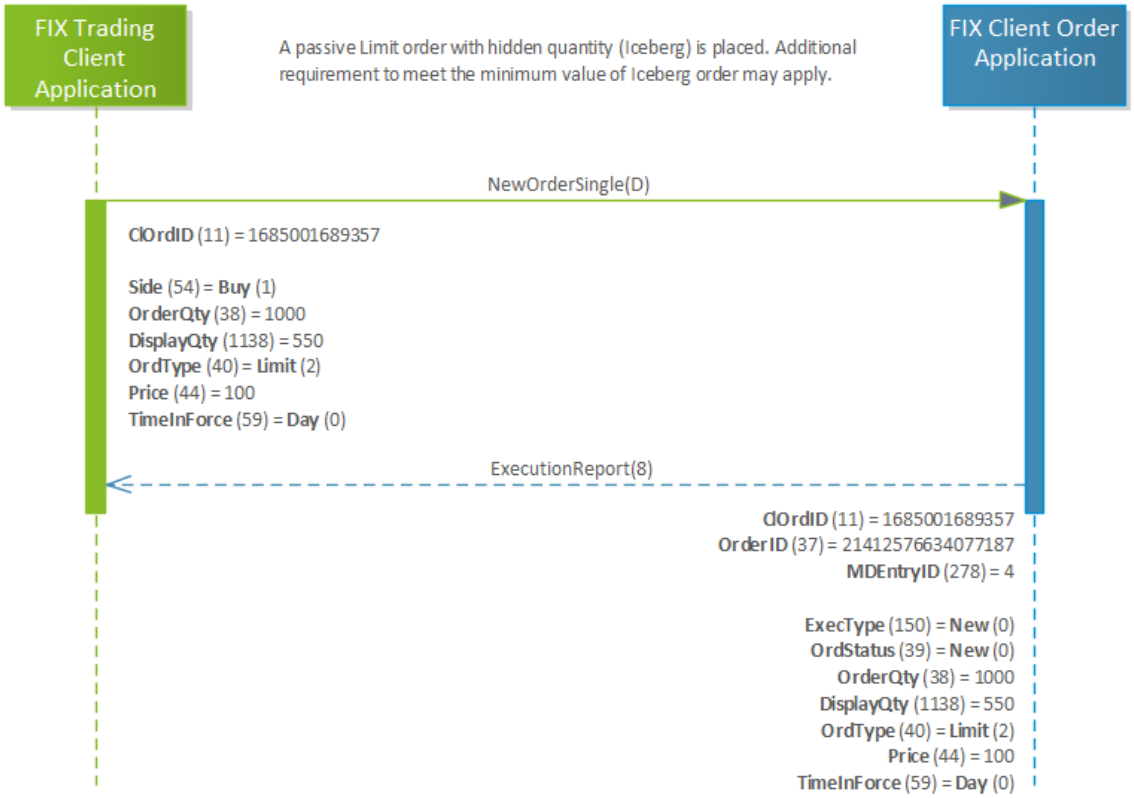
Order Accepted



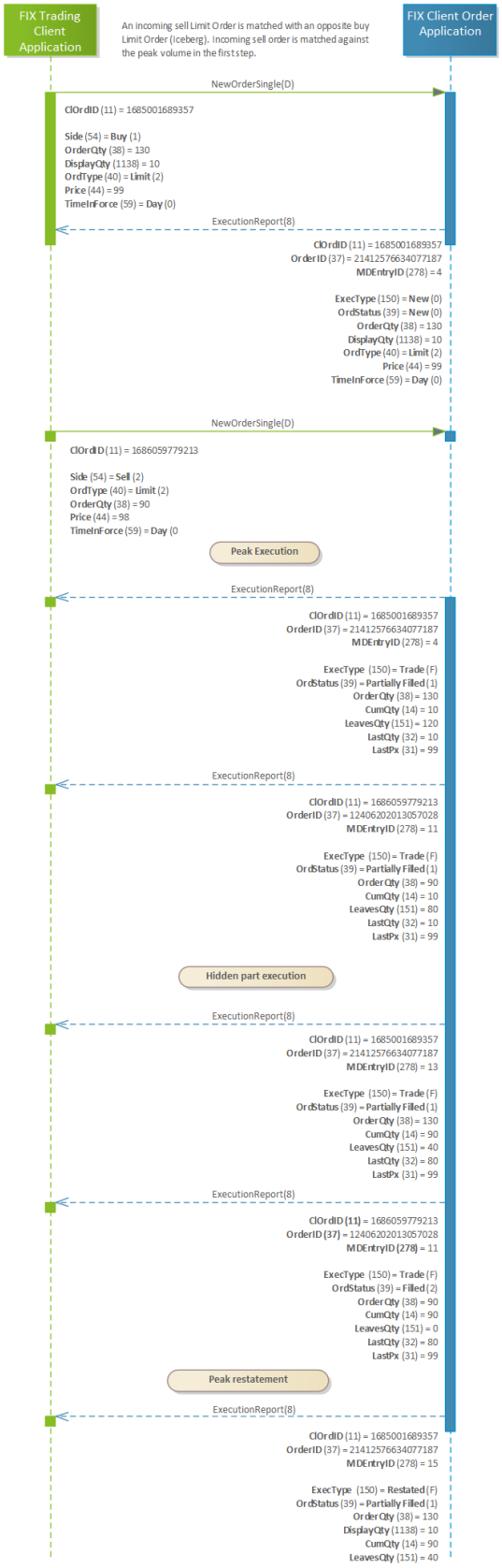
Order – Accepted and Filled – Full and Partially



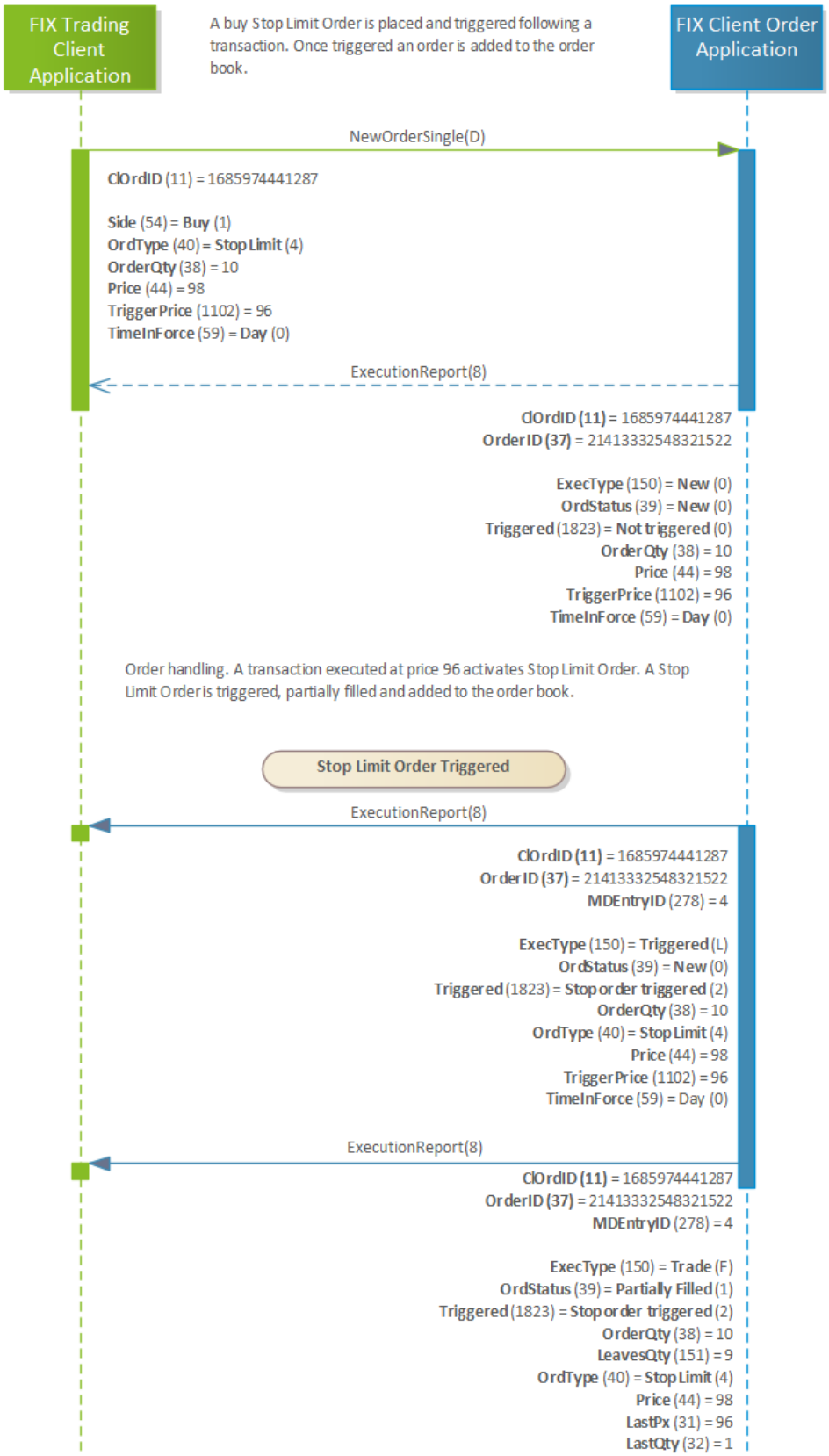
Iceberg Order - Accepted



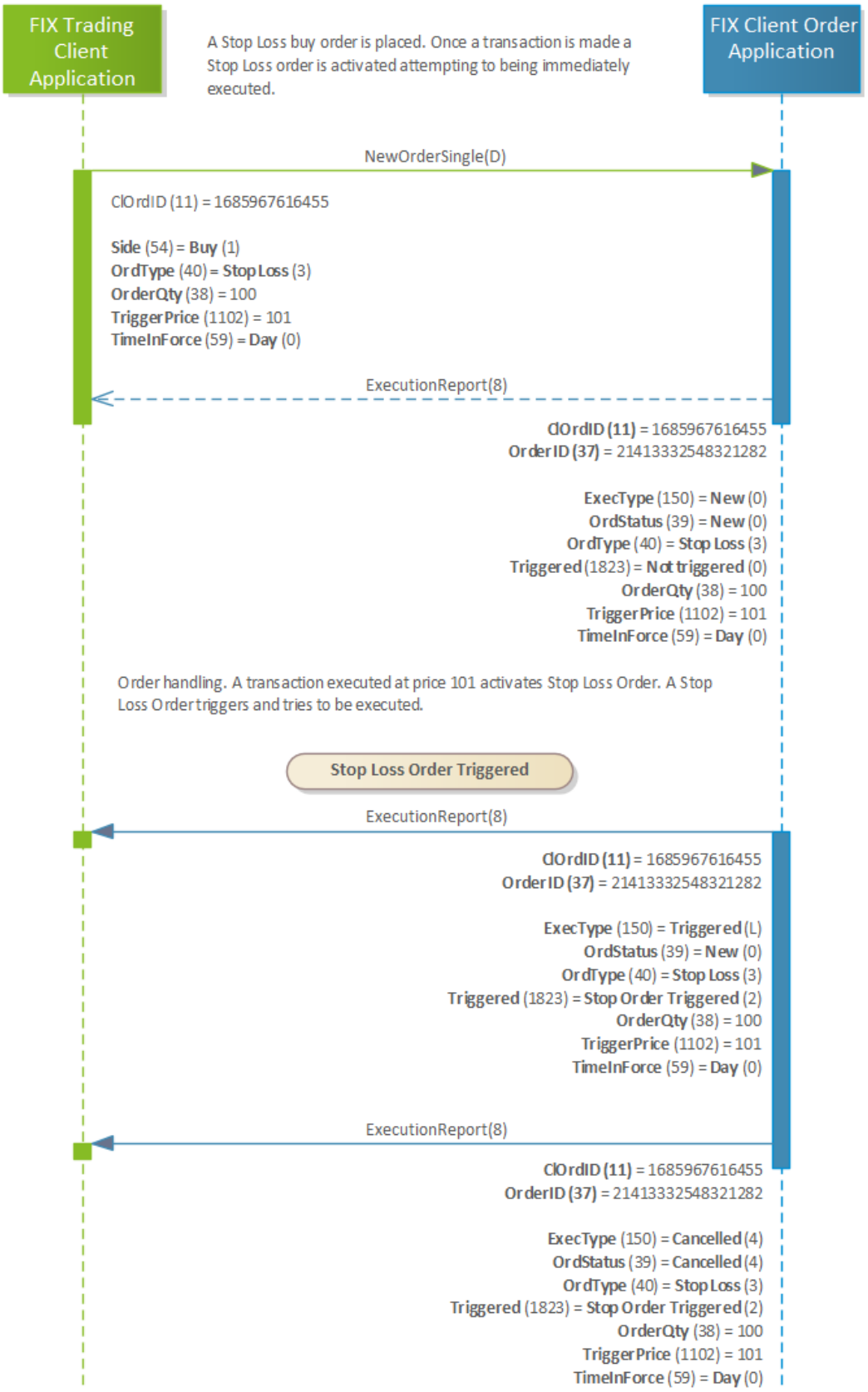
Iceberg Order – Accepted, Filled Partially and Restated



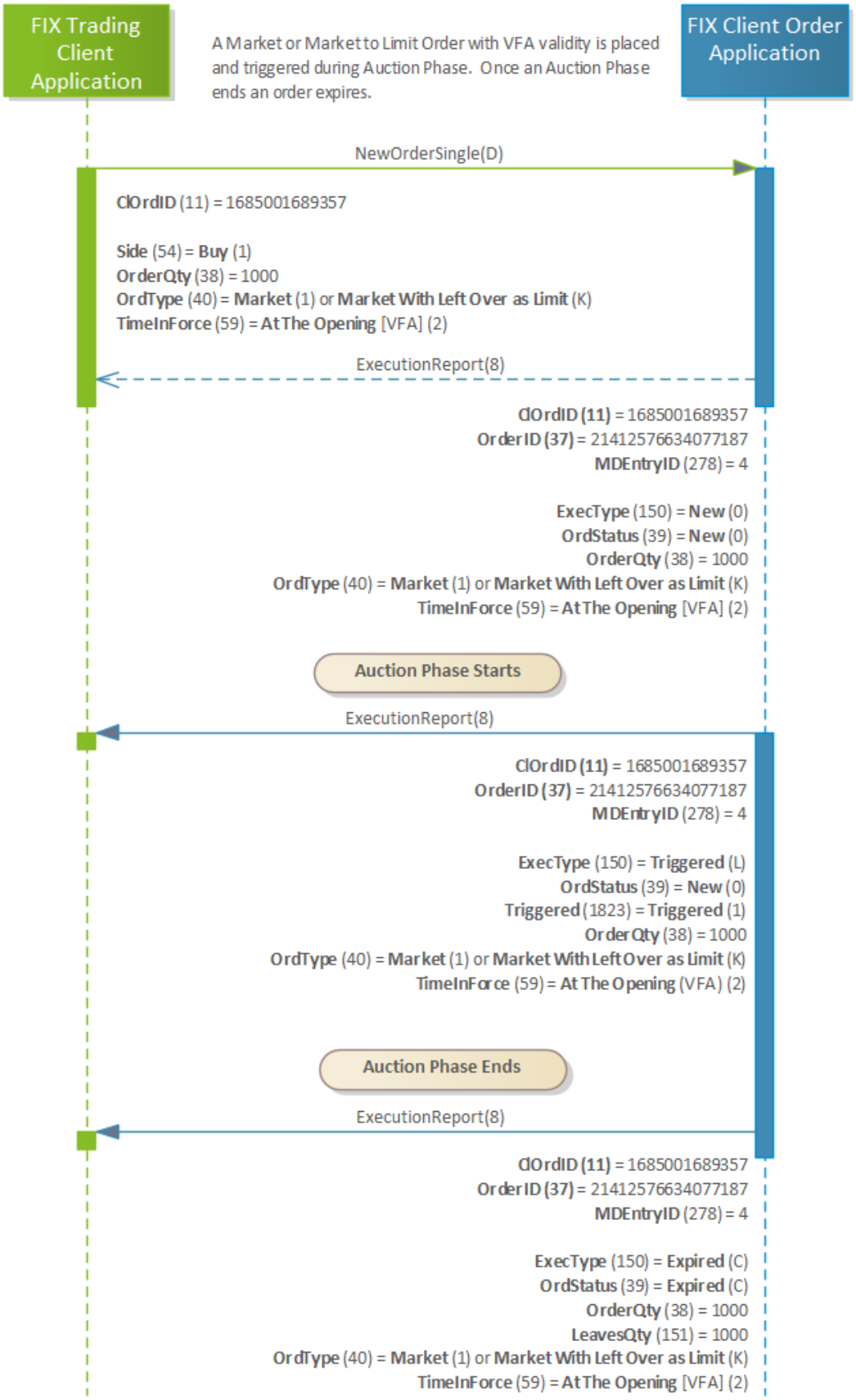
Stop Limit Order – Accepted and Triggered



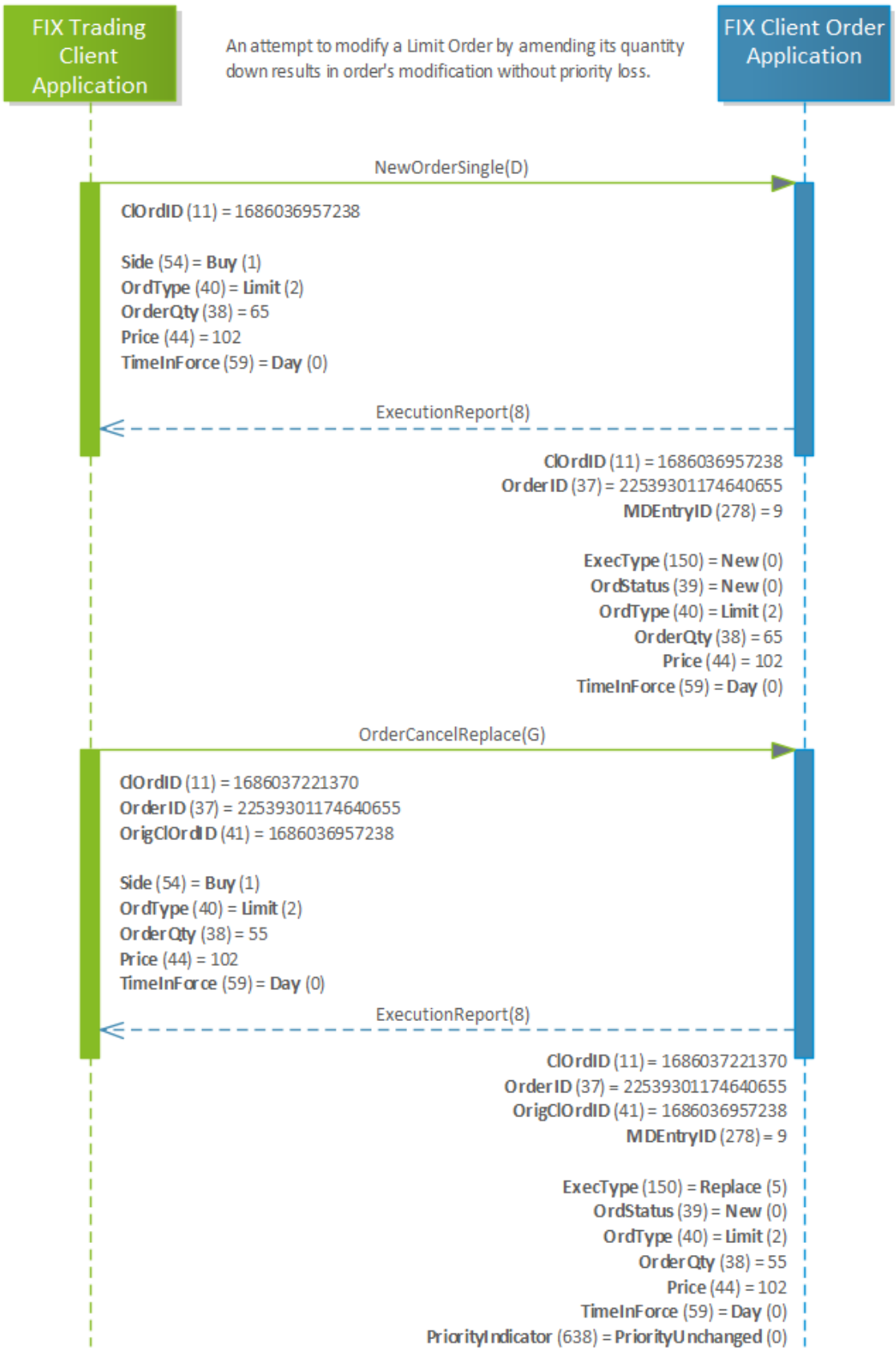
Stop Loss Order – Accepted, Triggered and Cancelled



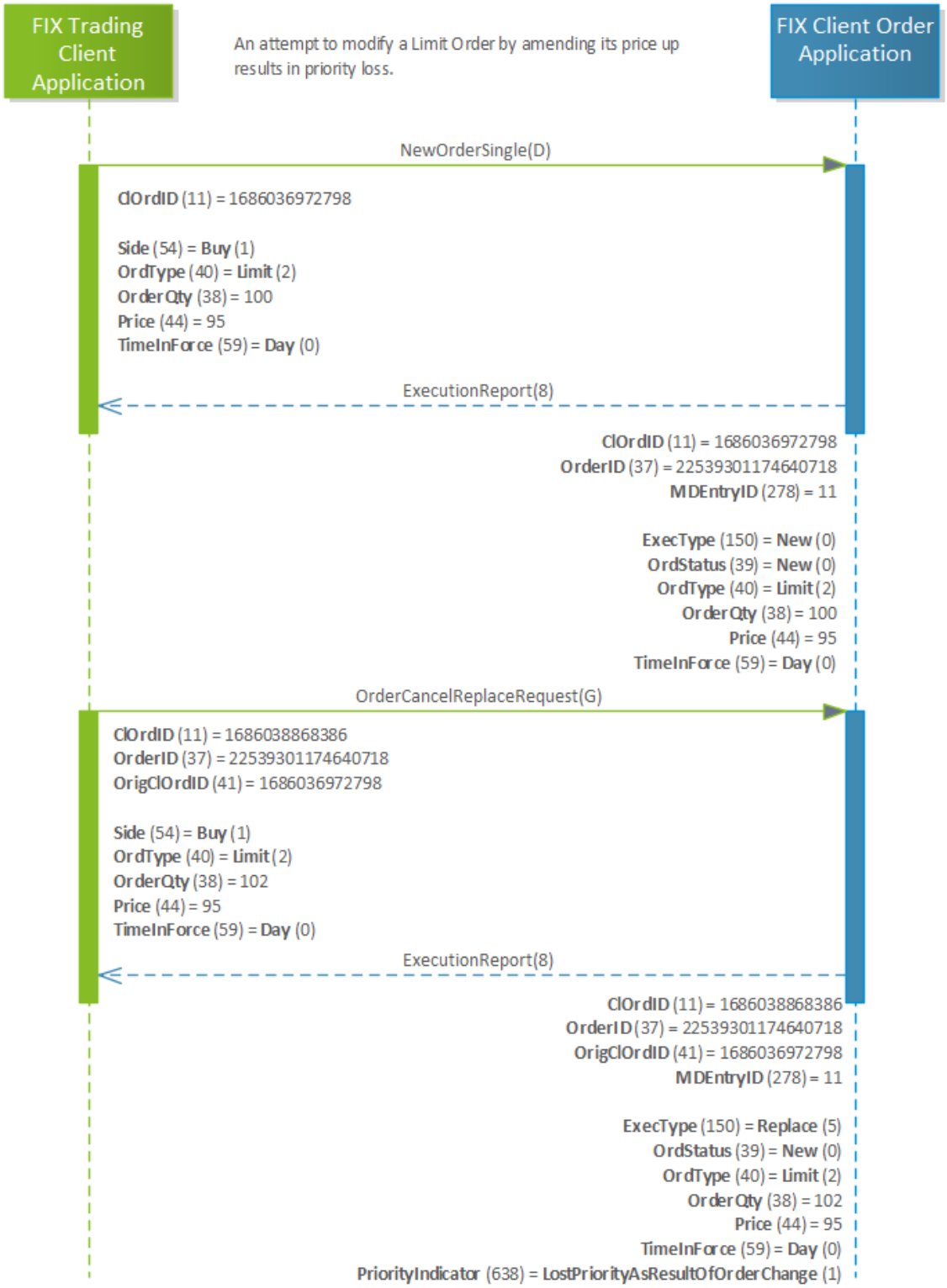
Market Order and Market to Limit Order with TIF VFA – Triggered during Auction



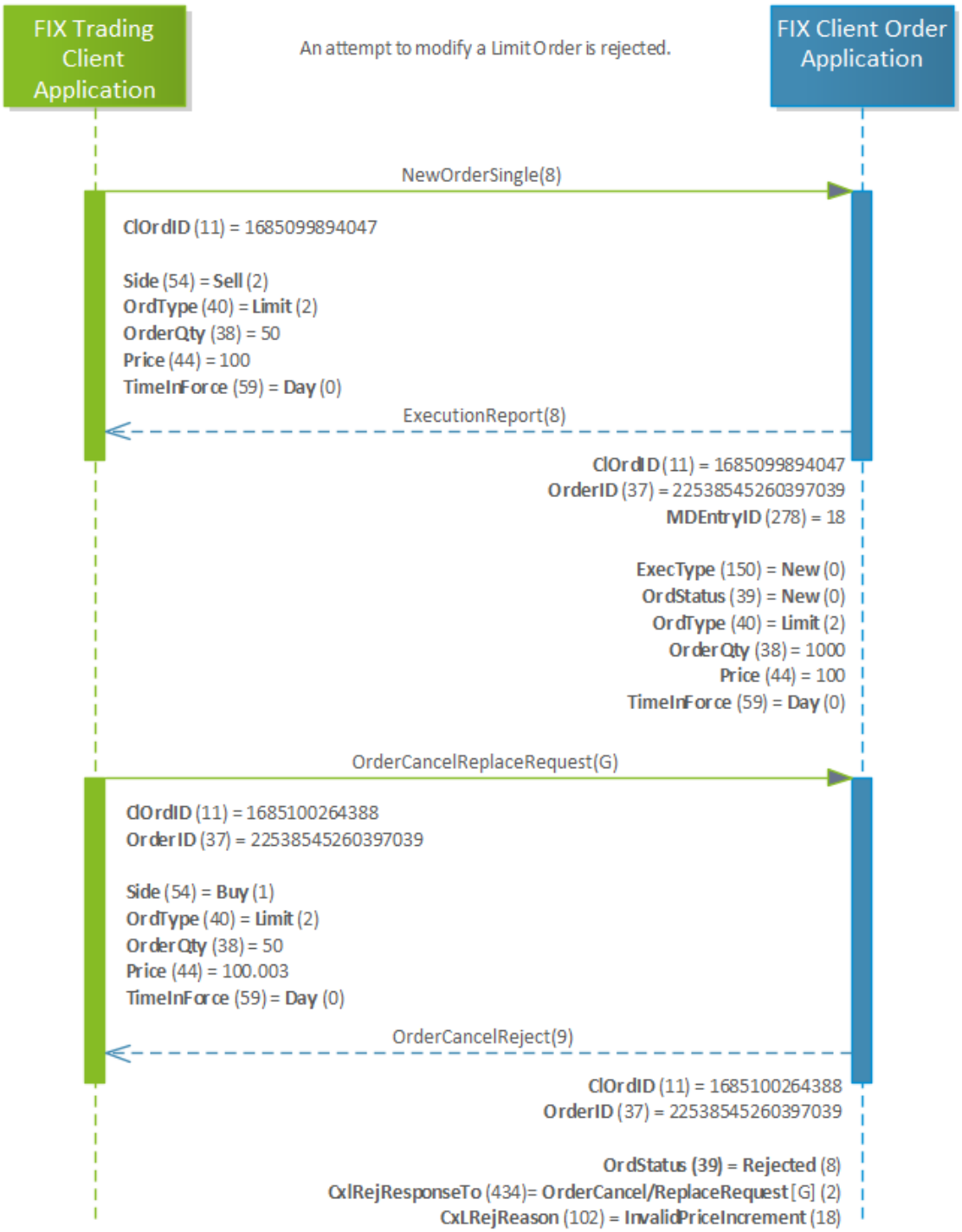
Order Modification without priority loss



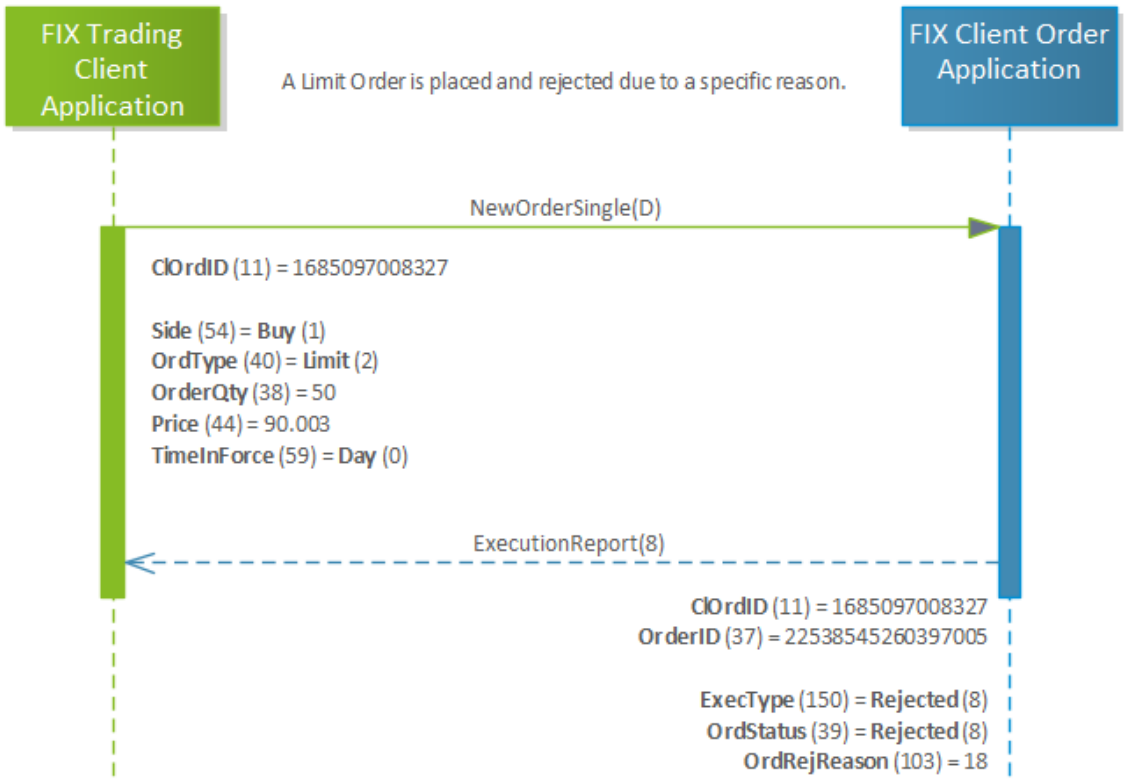
Order Modification with priority loss



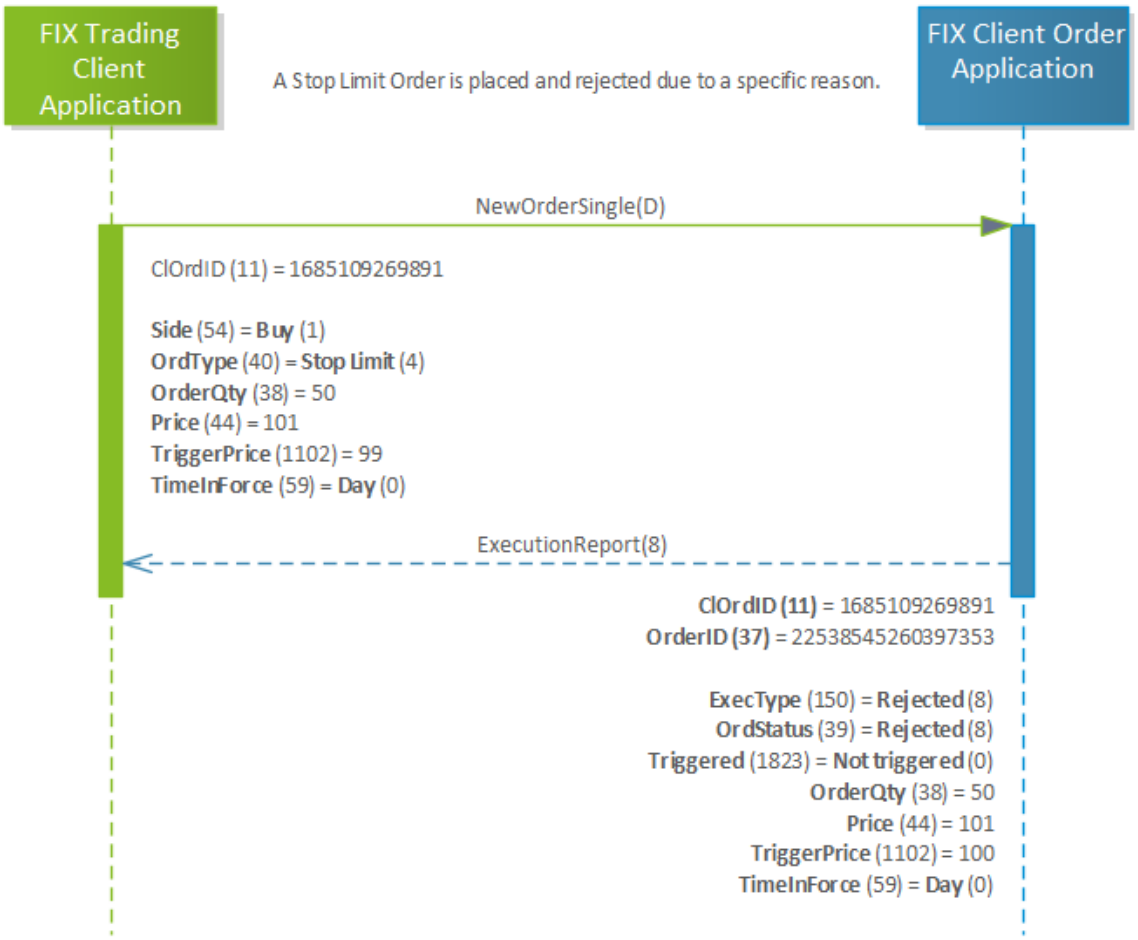
Order Modification - Rejected



Order - Rejected

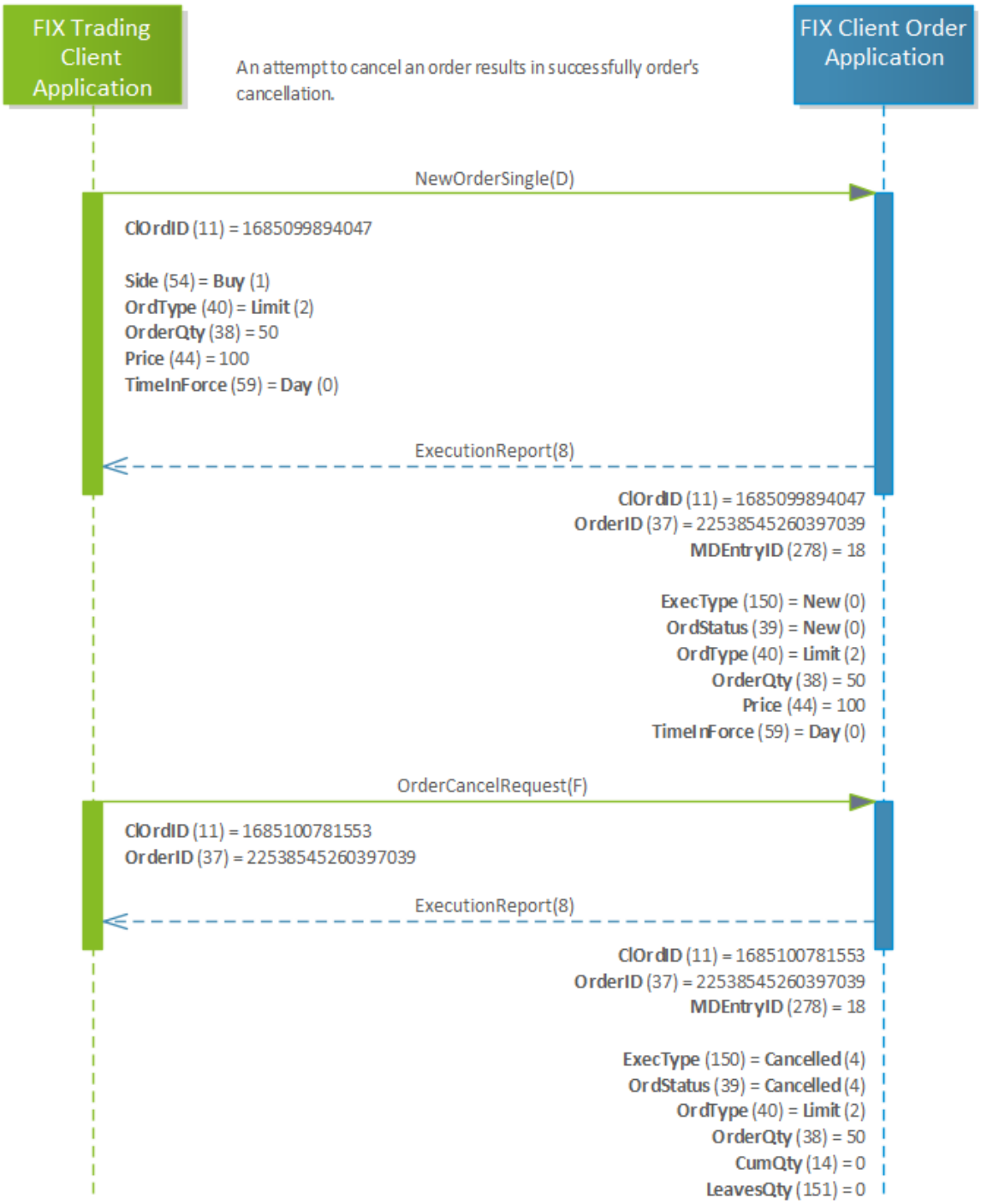


Stop Limit Order - Rejected

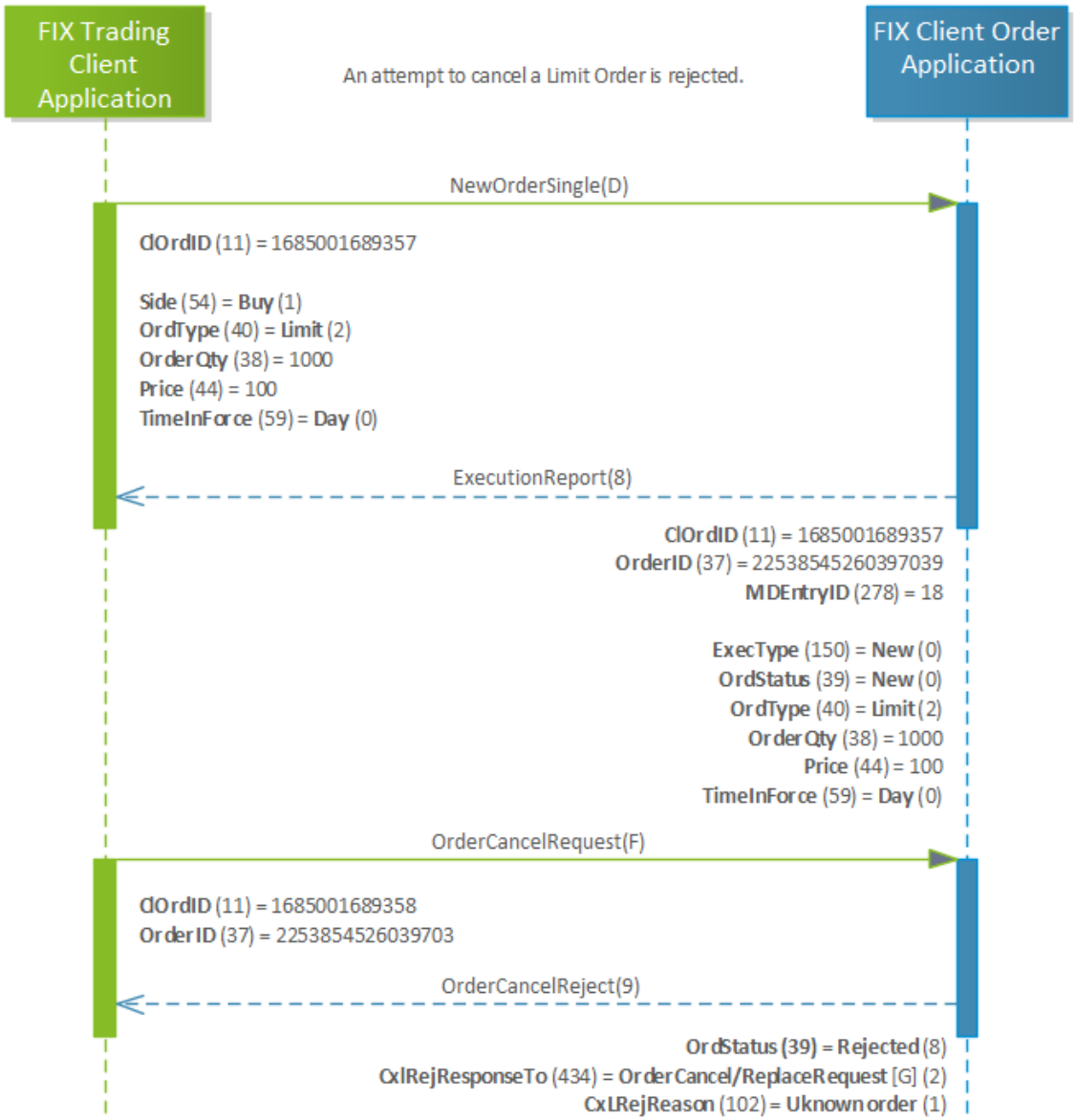


Market Order and Market to Limit Order IOC/FOK – Rejected during Auction

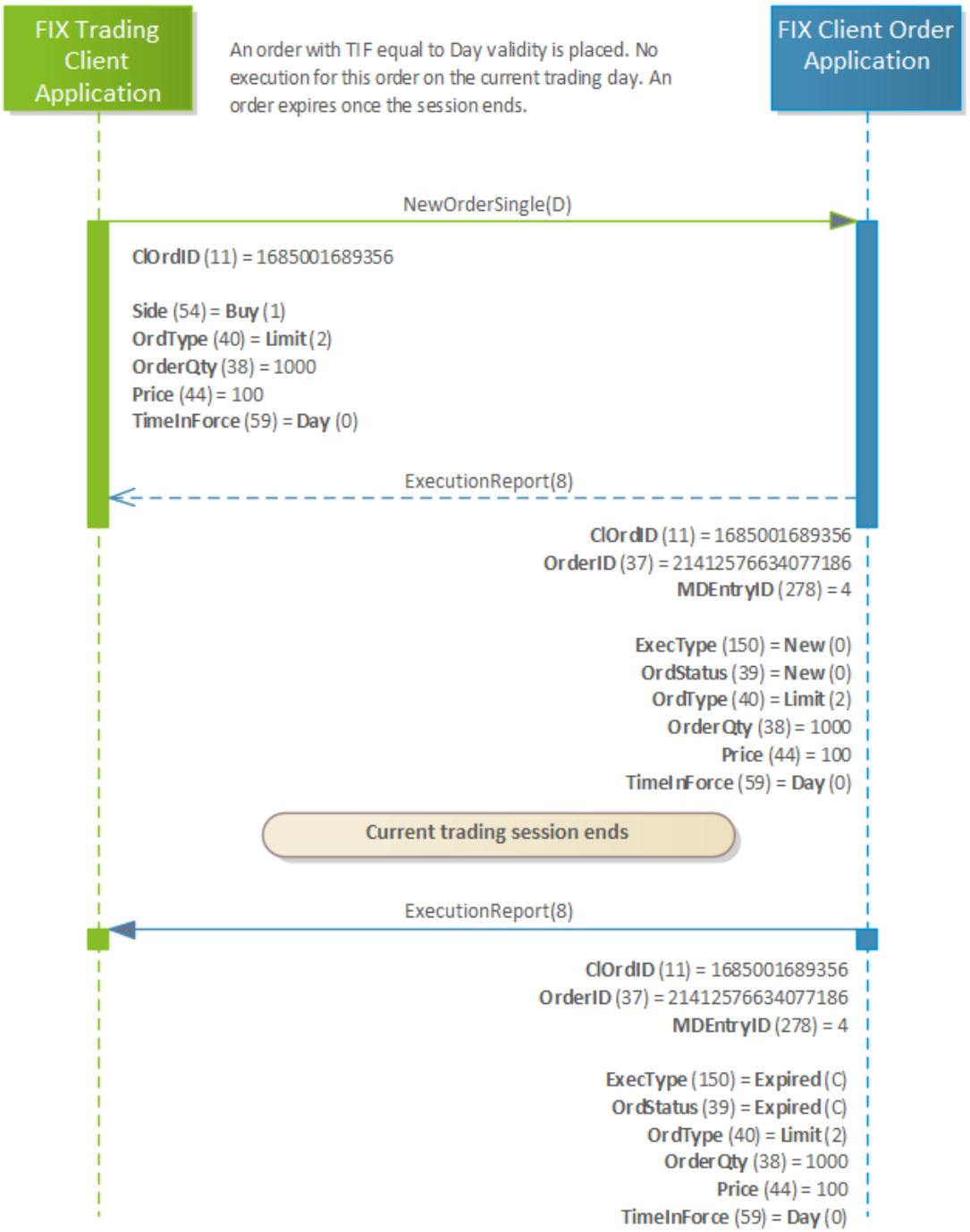
Order Cancellation - Accepted



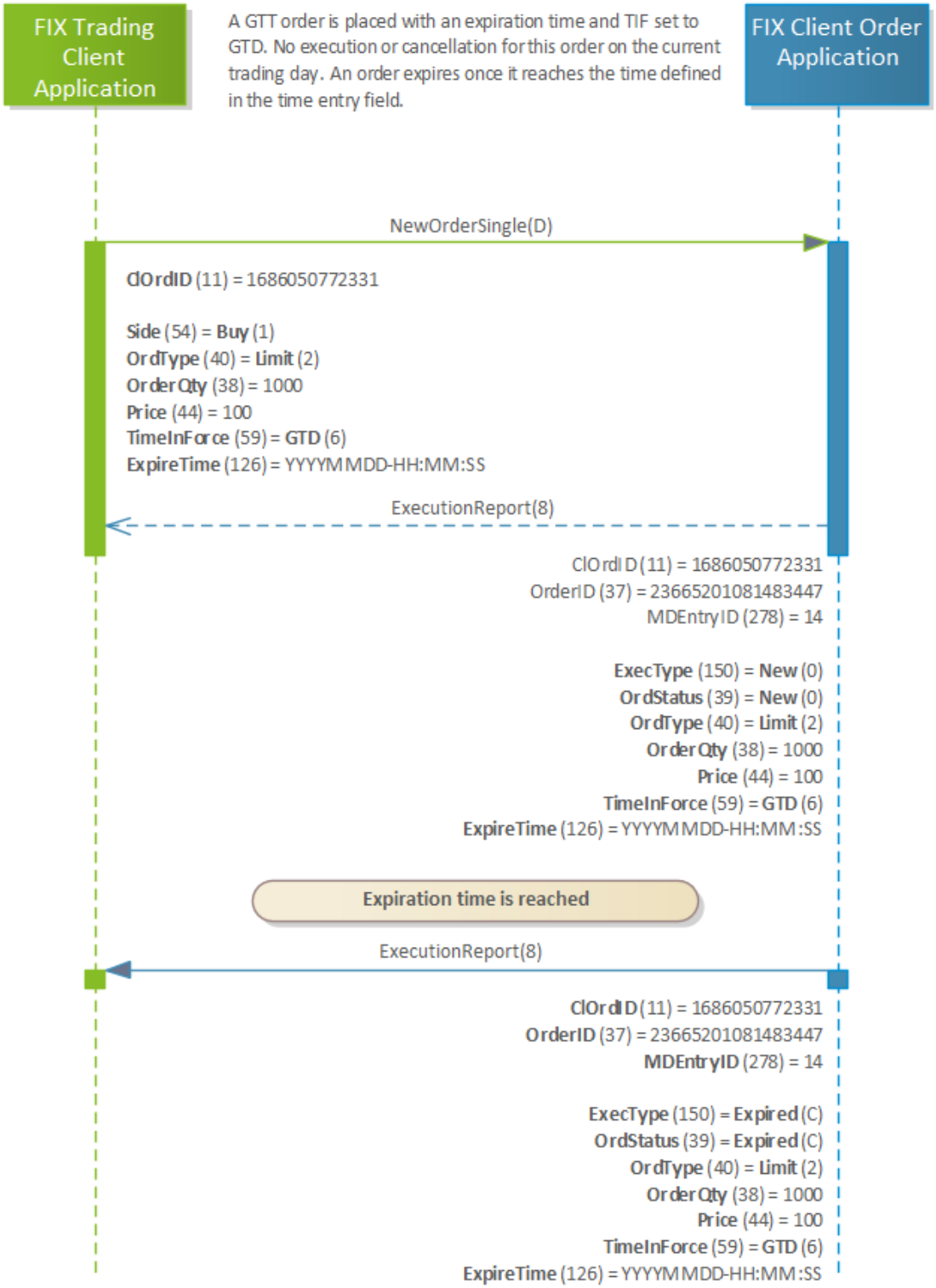
Order Cancellation - Rejected



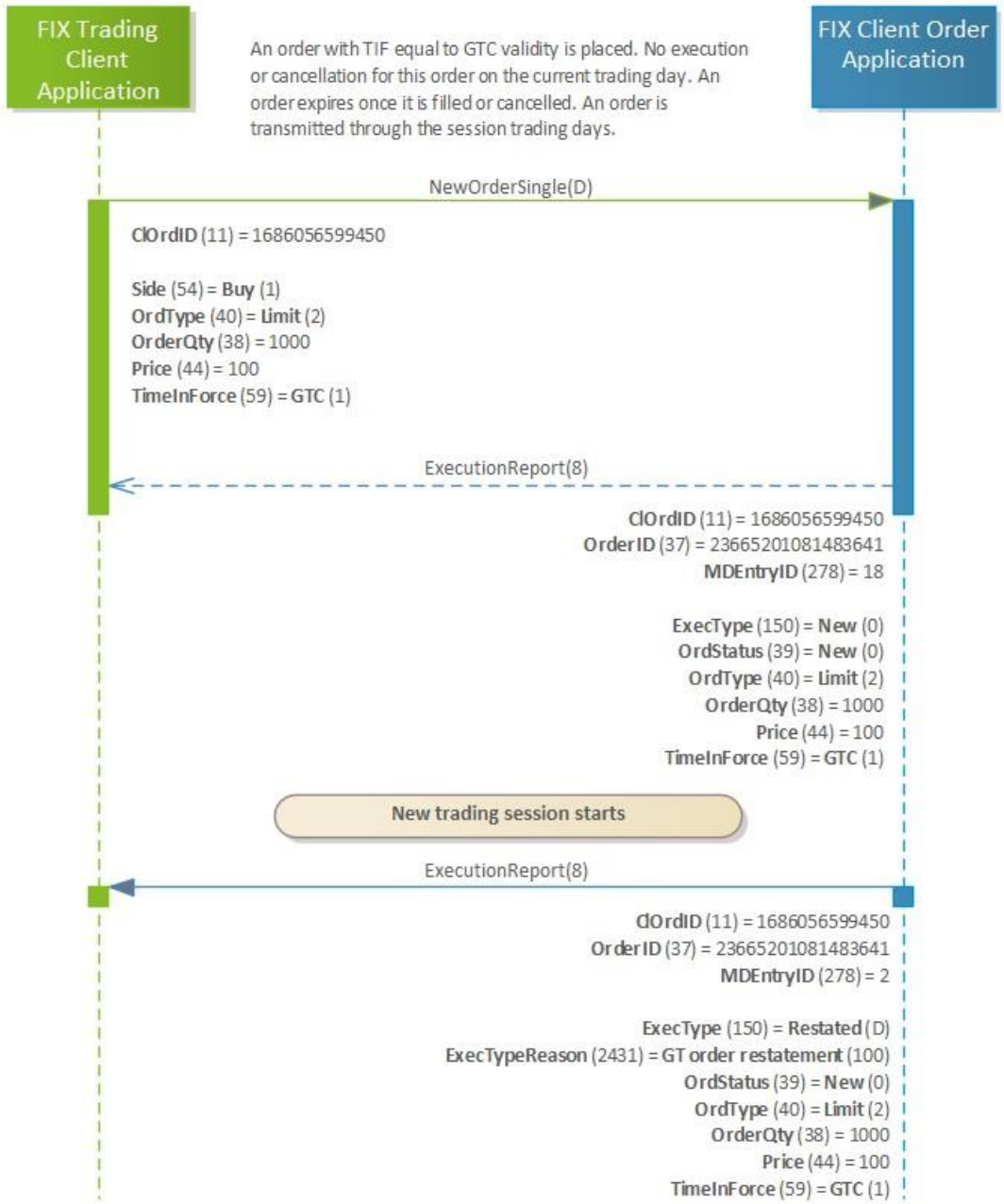
Day Order - Expired



GTT Order - Expired

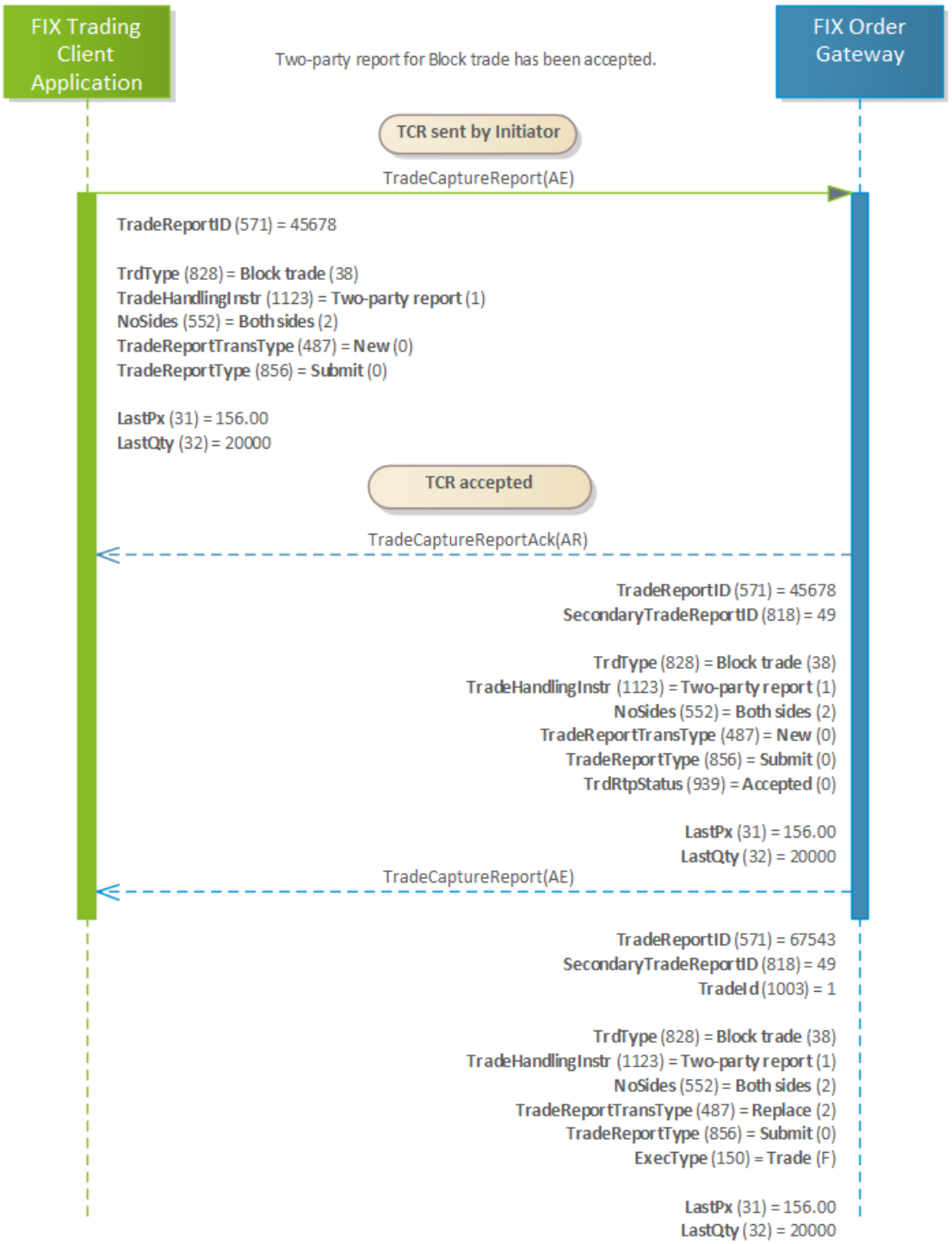


GTC Order - Restatement

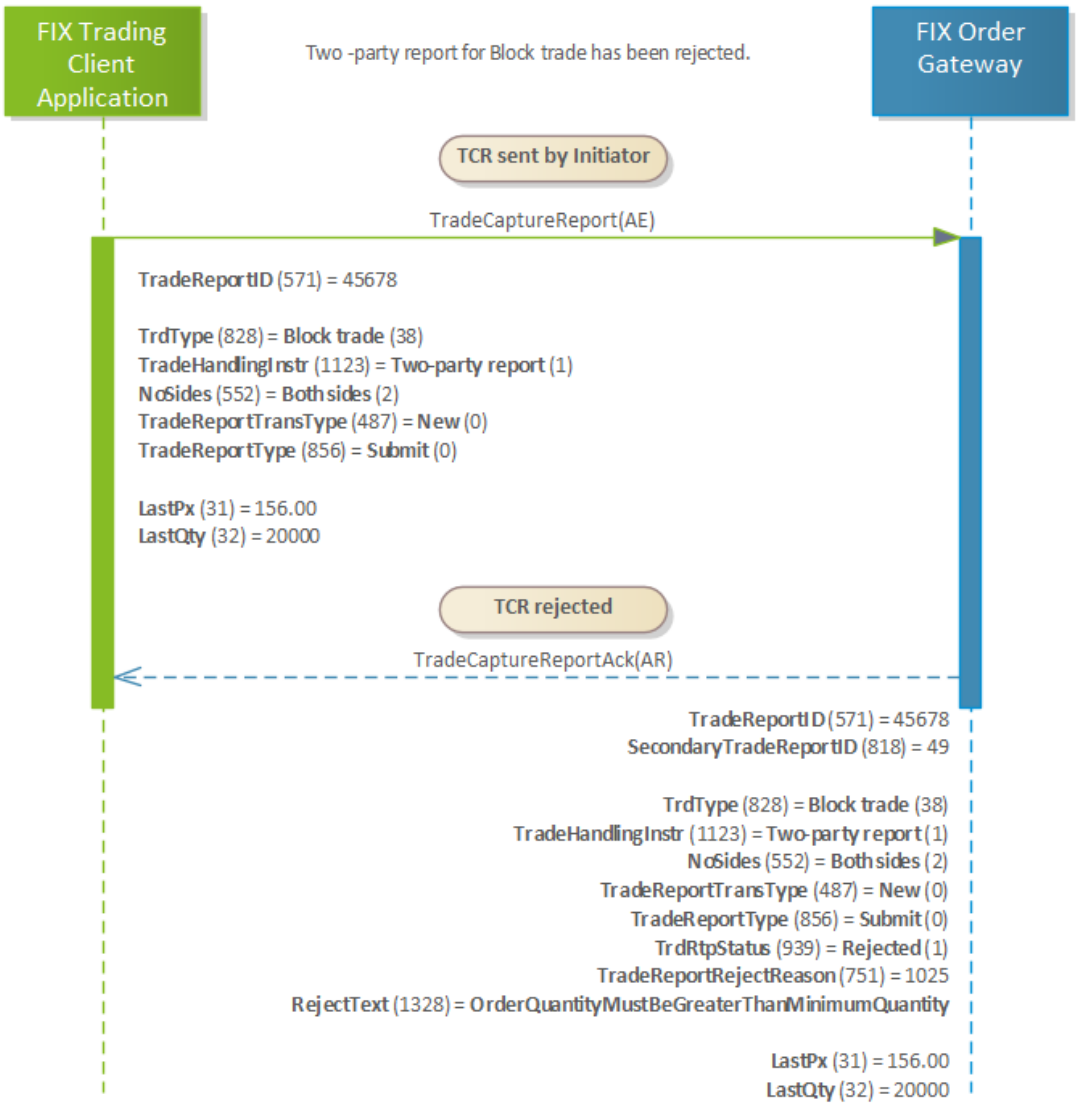


8.2. Off-Book (Block or Cross)

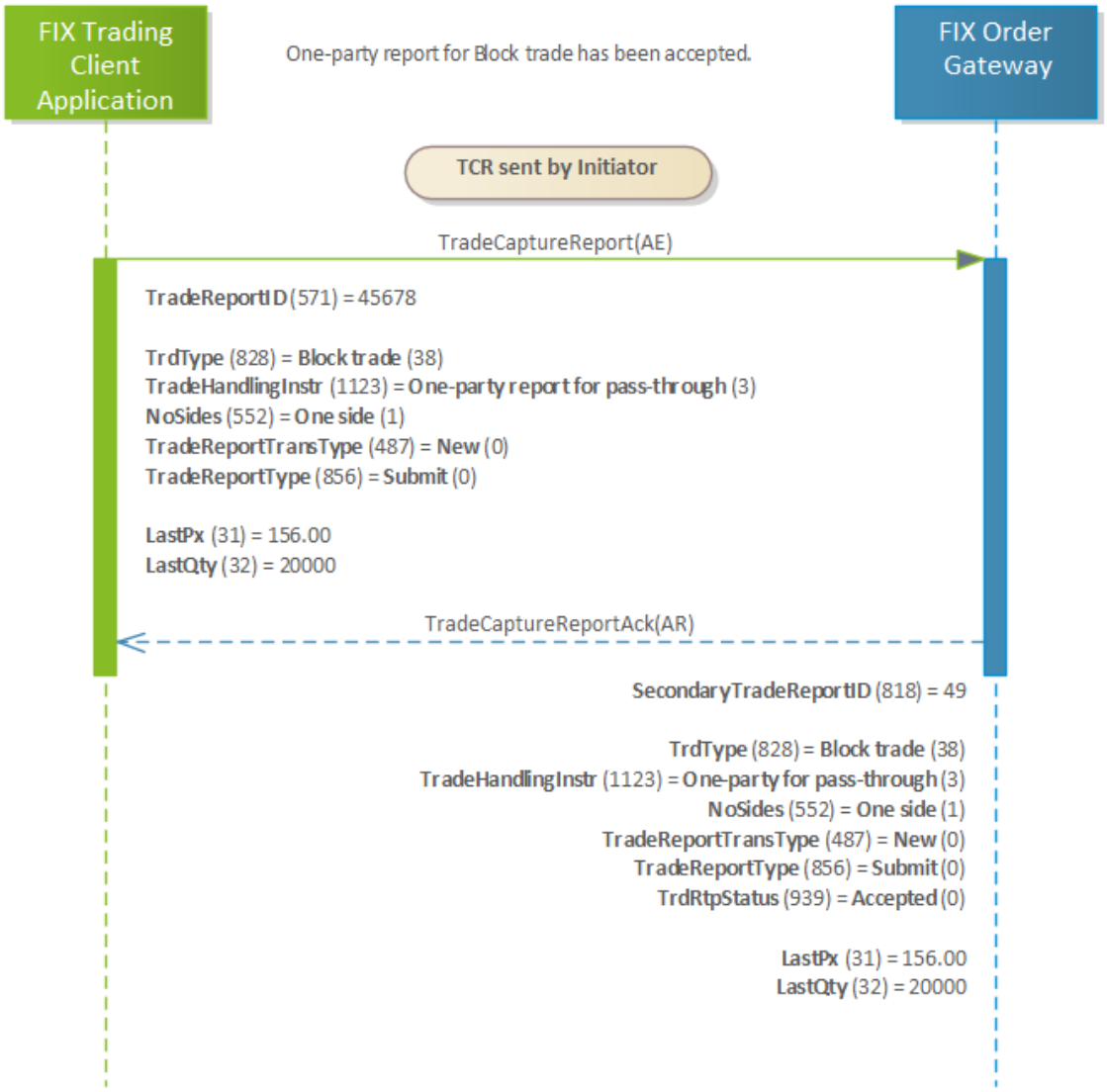
Two-party – Accepted



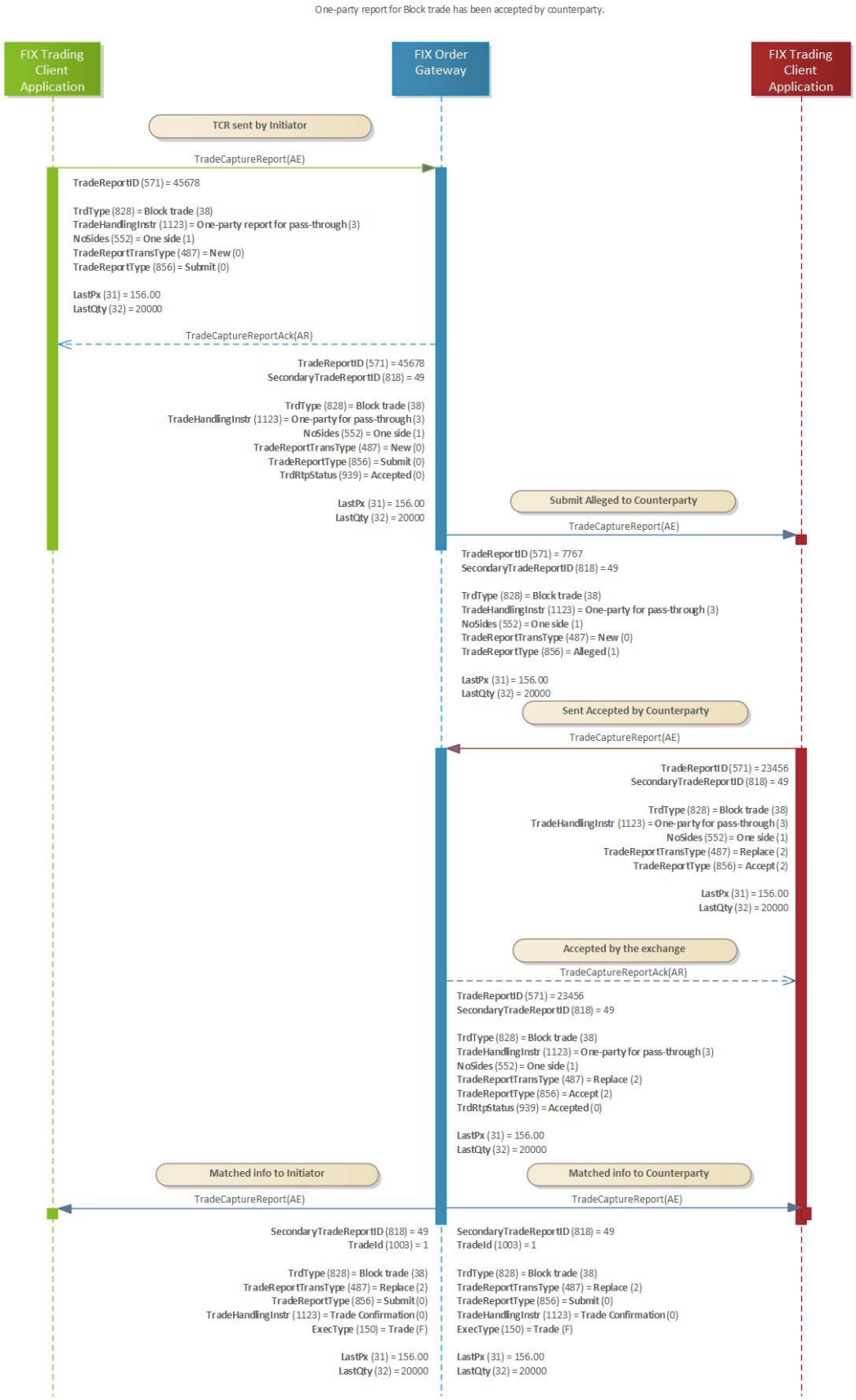
Two-Party – Rejected



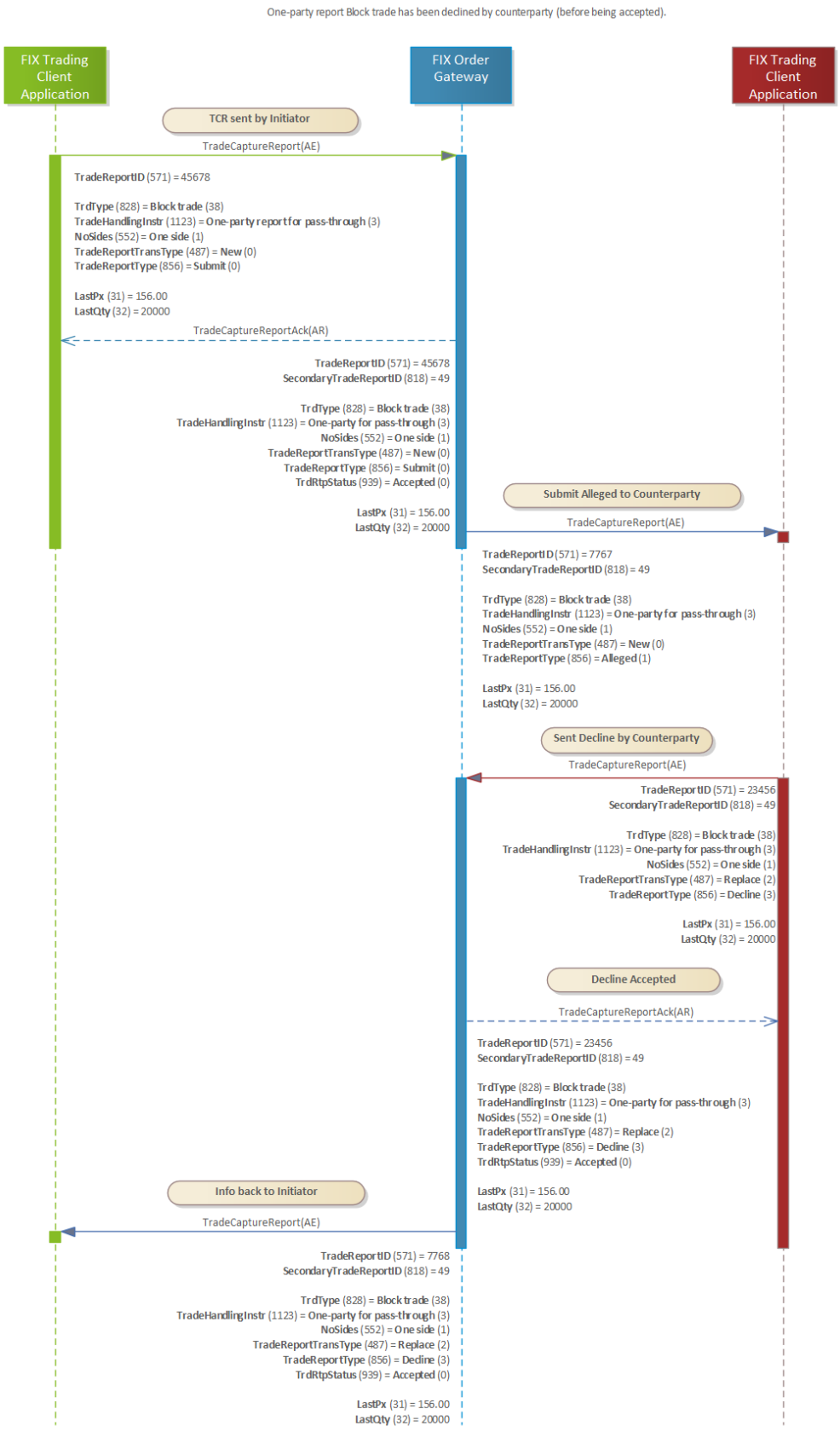
One-party - Accepted



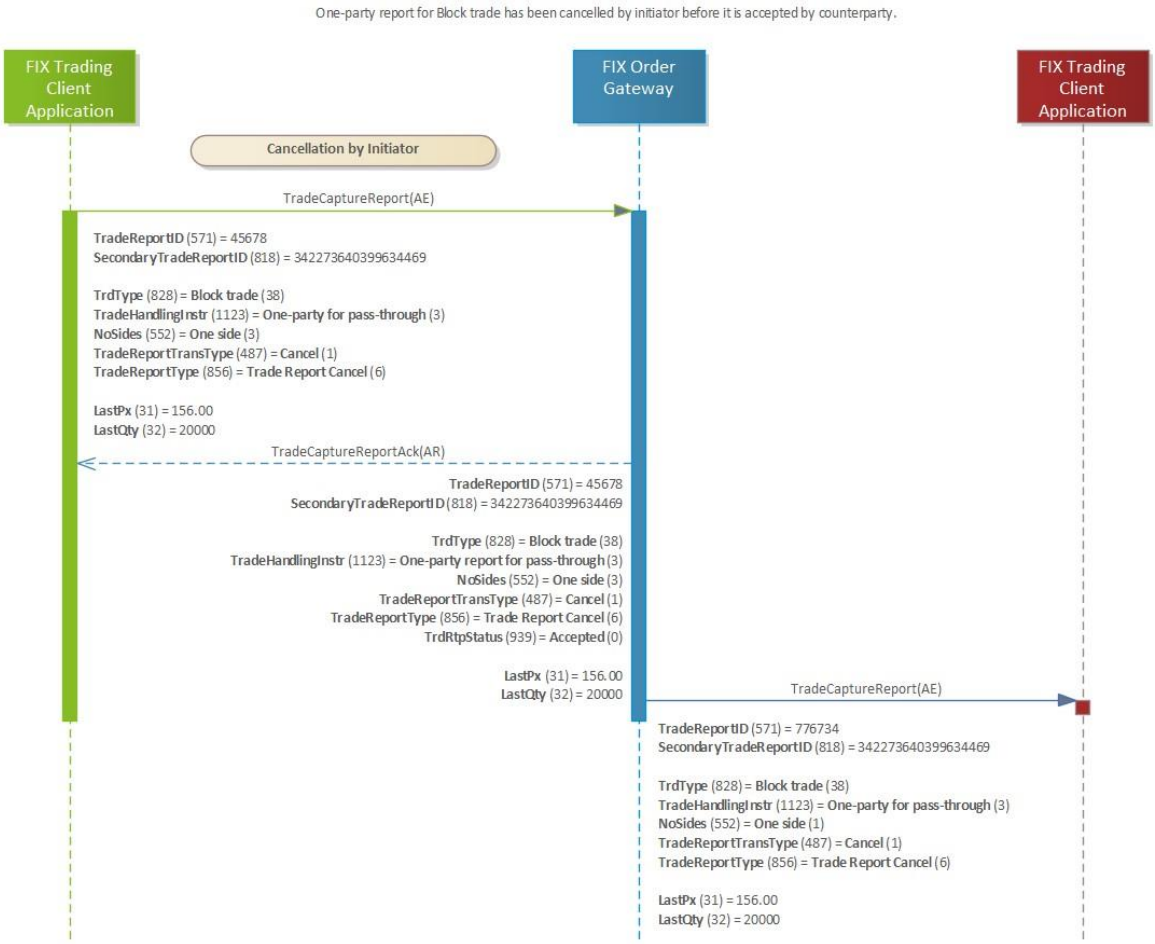
One-party – Accepted by Counterparty



One-party – Denied by Counterparty



One-party – Cancelled by Initiator



One-party – Rejected