FIX for Genium INET for NASDAQ OMX Nordic 1.0

Revision 1.35

2014-02-26 | Ulf Ahlenius

Confidentiality/Disclaimer

This draft specification is being forwarded to you strictly for informational purposes and solely for the purpose of developing or operating systems for your use that interact with systems of NASDAQ OMX and its affiliates (collectively, NASDAQ OMX). This specification is proprietary to NASDAQ OMX.

NASDAQ OMX reserves the right to withdraw, modify, or replace this specification at any time, without prior notice. No obligation is made by NASDAQ OMX regarding the level, scope or timing of NASDAQ OMX’s implementation of the functions or features discussed in this specification. The specification is provided “AS IS,” “WITH ALL FAULTS”. NASDAQ OMX makes no warranties to this specification or its accuracy, and disclaims all warranties, whether express, implied, or statutory related to the specification or its accuracy. This document is not intended to represent an offer of any terms by NASDAQ OMX. Whilst all reasonable care has been taken to ensure that the details contained herein are true and not misleading at the time of publication, no liability whatsoever is assumed by NASDAQ OMX for any incompleteness or inaccuracies. By using this specification you agree that you will not, without prior written permission from NASDAQ OMX, copy or reproduce the information in this specification except for the purposes noted above. You further agree that you will not, without prior written permission from NASDAQ OMX, store the information contained in this specification in a retrieval system, or transmit it in any form or by any means, whether electronic, mechanical, or otherwise except for the purposes noted above. In addition you agree that you will not, without prior written permission from NASDAQ OMX, permit access to the information contained herein except to those with a need-to-know for the purposes noted above.

Contents

1 References

[1]
FIX 4.4 Protocol Specification
http://www.fixprotocol.org/specifications/fix4.4spec

[2]
FIX 5.0 SP2 Protocol Specification
http://fixprotocol.org/specifications/fix5.0sp2spec

[3]
FIX Protocol Limited, Market Data Optimization Working Group:
Recommended Practices for Book Management, Version 2.00, 2007
http ://fixprotocol.org/documents/2518/MDOWG_Book_Mgt v20.doc

[4]
RFC 2045, Base64 encoding for MIME
http ://www.ietf.org/rfc/rfc2045.txt

2 Overview

This document contains the specification for the FIX interface to NASDAQ OMX Nordic Genium INET, the Derivatives, Fixed Income and Commodities Market trading system. The interface is based on the FIX Protocol 4.4 standard (Financial Information exchange). More detailed information about the standard can be found in FIX specification document see [1].

The interface implemented by NASDAQ OMX follows the FIX specifications as far as possible. In the majority of cases the structure and semantics of the messages are identical to the standard.
In some cases, the protocol has been extended to cover functions not considered by the standard. These extensions are clearly detailed in the document. In other cases, the standard is ambiguous or indicates that the details should be bilaterally agreed between the parties. In these cases this manual provides a detailed description to avoid any possible ambiguity.

To avoid possible duplication in the sources of information, this document does not include explanations of those matters that comply exactly with the standard. Therefore, the standard documentation should be considered as the main source of information for any matter that is not explicitly covered in this manual.

This specification tries not to repeat what is specified in the FIX standard. In many cases however, the FIX standard is, by necessity, more generic than that required for a specific marketplace. In other cases NASDAQ OMX has found reason to clarify matters. NASDAQ OMX tries to be explicit on deviations from the FIX standard specification in order to avoid confusion.

2.1 Supported messages

2.1.1 Administrative messages

Logon
Logout
Sequence Reset

Resend Request
Reject
Heartbeat
Test Request

2.1.2 Inbound Application messages

User Request
New Order Single
Order Cancel Replace Request
Order Cancel Request
Mass Quote
Trade Capture Report
Security Definition Request
New Order List
List Cancel Request
One Sided Auction Request
One Sided Cancel Auction Request
Allocation Instruction

2.1.3 Outbound Application messages

User Response
User Notification
Execution Report
Order Cancel Reject
Business Message Reject
Mass Quote AcknowledgementTrade Capture Report
Trade Capture Report Ack
Security Definition (TMC/REPO ack/reject)
Security Definition Update Report (TMC/REPO)
List Status
One-Sided Auction Request Ack
Allocation Report

2.2 NASDAQ OMX Extensions

In order to support specific functionality of the back-end not covered by the FIX 4.4 standard protocol, a number of extensions have been made. A few messages and a number of fields have been added. Whenever an entity needed to be added, NASDAQ OMX has tried to make use of entities from later FIX versions (FIX 5.0 SP2 in particular). Inventing new fields or messages have only been done when no other alternative existed.

Throughout this specification all deviations from the standard FIX protocol has been marked with “NASDAQ OMX Extension”.

If a field marked as an extension has a tag number lower than 1500, it is taken from the FIX 5.0 SP2 specification. If the added field has a tag number higher than 20000, it is a new field invented by NASDAQ OMX.

2.3 The NASDAQ OMX Repository

This specification defines the full set of messages, fields and enumerated values that can be used. As with most FIX implementations, this only supports a small subset of all available messages, components, fields and enumerated values defined in FIX 4.4. An FPL-formatted repository corresponding to this specification is delivered separately.

NOTE: Inbound messages not conforming to this spec, will be rejected with a session-level Reject message.

3 The FIX Session

The session layer conforms to the standard FIX session. Please see the standard FIX specification for additional details.

3.1 CompIDs

The Sender- and TargetCompID uniquely define the FIX session. A session can only be active (established) between two hosts simultaneously. Any attempts to establish a second FIX session using the same CompIDs (for instance to a backup gateway) in parallel will be rejected.

- The TargetCompID for transactions sent *inbound* to the Exchange will be "GENIUM" for production and "GENIUM_TEST" for test systems.
- The SenderCompID for transactions sent *outbound* from the Exchange will be "GENIUM" for production and "GENIUM_TEST" for test systems.

3.2 SenderSubID

Each inbound business transaction must have the SenderSubID (tag 50) field set to an authenticated user. One user can be authenticated by setting the Username and Password field in the Logon message. Additional users can be authenticated using the User Request message. See chapter 4 for a description on how to authenticate additional users.
The SenderSubID on incoming transactions will be echoed back in TargetSubID (tag 57) on outbound transactions.
NOTE: On the Logon or User Request, the SenderSubID must be set to the user id the client intends to log on.

3.3 User Authentication

Each incoming business transaction must have a username set in the SenderSubID field. The user needs must be authenticated for the transaction to be accepted. A user is authenticated by setting the Username (553) and Password (554) fields in the Logon message.

3.3.1 Renewal of passwords

A new password may be set by setting the NewPassword (925) field along with the current password in the Password (554) field. This can be done either with the Logon message or the User Request message. The SessionStatus (1409) field of the Logon returned to the client can be checked to see if the new password was accepted.

3.3.2 Expired passwords

If the password has expired when a client tries to log in, the system will respond with a Logout message with SessionStatus set to 8 – Password expired. To gain access, the client must issue a new Logon message with NewPassword set (along with the expired password in Password). If the new password is not valid, the system will respond with another Logout message. SessionStatus will be set to 3 – New session password does not comply with policy. The client will be able to log in again with another new password.

3.4 Logon

At Logon, clients are identified by:

- CompIDs
- IP Address

The Logon Username and Password fields are used to authenticate the client. When the client is authenticated, the system responds with a Logon message to the client.

3.5 Heartbeat intervals

Heartbeat intervals are negotiated at Logon using the HeartBtInt (108) field. The system allows heartbeat intervals greater than 10 seconds. **Recommended heartbeat interval is 30 s.** A heartbeat interval set lower than 10 seconds will result in a Logout response.

3.6 Encryption

The system does not support encryption.

3.7 Datatypes and required fields

This specification does not change the datatype on any fields defined in the standard FIX specification. There may be places where this specification restricts the value range of a field further than specified in standard FIX. This will be clearly marked in the spec.
All fields listed in this specification that are marked as required in the standard specification, are required also in this specification. This document specifies additional fields as required. These fields are marked with a ‘Q’ in the required column of the message listings.

3.8 Character encoding

In order to support the Scandinavian characters present in the back-end system, such as å, ä and ö, the FIX gateway will use the 8-bit standard ISO-8859-1 encoding, often called Latin-1. The lower 7 bits are compatible with the standard 7-bit ASCII character encoding.

3.9 Session lifetime

The FIX session lifetime is restricted to one trading day. The session lifetime is not ended at connectivity loss or even Logouts. The sequence numbers are reset to one each morning.

3.10 Failover and message recovery

At reconnect and Logon standard FIX message recovery is performed. All FIX sessions have at least one primary and one secondary gateway to which the session states are fully replicated. This means that regardless to which gateway a client connects, full message recovery is provided.
A client cannot have the same FIX session active towards multiple gateway instances simultaneously.

3.10.1 Order Suspension/inactivation at connection loss

A FIX session can be configured by the marketplace to automatically suspend all outstanding orders at FIX connection loss. At reconnection the FIX client will be able to cancel the suspended orders.

3.11 FIX Session Level Test Cases

This implementation is fully compliant with the session-level test cases specified in the standard FIX 4.4 Specification, Volume 2, section “FIX Session-level Test Cases and Expected Behaviors”. The only exception is the encryption test cases.

3.12 Drop Copy Sessions

Drop Copy Sessions, or Drops, can be set up to mirror outbound traffic per FIX session(s) or participant(s) outbound traffic. The following business-level messages can be seen on a Drop session:

- Execution Reports
- Trade Capture Reports
- Trade Capture Report Acks

NOTE: Rejects (on orders, cancels, cancel replaces and trade reports) will **not** be seen on the Drop.
Drop Copy Sessions are authenticated just like regular sessions using the Logon message with a username and password supplied. Be aware that since a drop may be configured to receive updates from multiple users, the TargetSubID of the received messages may be different than the authenticated user.
All copied messages will have the CopyMsgIndicator (797) tag set to “Y”.

3.12.1 Drop Party Identifiers

To be able to identify the legal owner of an order or trade, all business-level messages on a drop will have the Parties block (Execution Reports) or RootParties block (Trade Capture Reports and Trade Capture Report Acks). The Parties/RootParties block will contain:

453	NoPartyIDs			Set to 1 or 2
→	448	PartyID		Trader id or participant id.
→	447	PartyIDSource		Valid values: D = Proprietary /Custom code
→	452	PartyRole		Identifies the type or role of the PartyID (448) specified. Valid values: 1 = Executing Firm 12 = Executing Trader

On a regular FIX session these identifiers are given by the TargetCompID (participant) and TargetSubID (trader), but on a drop these fields identify the drop session client.

3.12.2 Non-FIX Drops

Drop Copy Sessions can be configured to send updates for orders and trades that were not entered via FIX.
For example, a Drop session is configured to send updates on all orders and trades for one or more participants. When an order is entered via OMnet, an Execution Report – Order Ack would be sent on the Drop session. A regular FIX session for the same participant in contrast, would **not** get order or trade acks for orders entered via Omnet.

3.13 The Standard Header

All FIX messages contain a Standard Header. The header contains important information such as session identifiers (CompIDs), sequence numbers and message type and length etc.

Tag num	FIX Field name	Req’d	Comment
8	BeginString	Y	Identifies beginning of new message and protocol version. ALWAYS FIRST FIELD IN MESSAGE. Valid values: FIX.4.4
9	BodyLength	Y	Message length, in bytes, forward to the CheckSum field. ALWAYS SECOND FIELD IN MESSAGE.
35	MsgType	Y	Defines message type ALWAYS THIRD FIELD IN MESSAGE.
49	SenderCompID	Y	As specified in separate agreement

50	SenderSubID		Required on inbound transactions. Must be set to a valid authenticated user.
56	TargetCompID	Y	As specified in separate agreement
57	TargetSubID		Should not be populated on inbound transactions. Will contain the value of incoming SenderSubID on outbound transactions. In some cases, such as in unsolicited cancels, TargetSubID will not be set.
34	MsgSeqNum	Y	Integer message sequence number.
43	PossDupFlag		Indicates possible retransmission of message with this sequence number. Always required for retransmitted messages
97	PossResend		Indicates that message may contain information that has been sent under another sequence number. Required when message may be duplicate of another message sent under a different sequence number.
52	SendingTime	Y	Time of message transmission (always expressed in UTC (Universal Time Coordinated, also known as "GMT")
122	OrigSendingTime		Original time of message transmission (always expressed in UTC (Universal Time Coordinated, also known as "GMT"). Required for message resent as a result of a ResendRequest.

3.13.1 Possible Duplicate vs. Possible Resend

The two FIX fields PossDupFlag (43) and PossResend (97) of the Standard Header have different purposes. The PossDupFlag is set on messages retransmitted as a result of a Resend Request. These messages have the original sequence numbers (MsgSeqNum).
PossResend is set on messages resent with a new sequence number. This may be used to resend an order which no response has been received. The gateway will check whether the client identifier (such as the ClOrdID, TradeReportID etc) in the message has been received before. If the client identifier has been seen before, the message will be dropped.

3.14 The Standard Trailer

All FIX messages end with a Standard Trailer. The trailer only includes a simple checksum field. The details on how to calculate the checksum can be found in the standard FIX specification.

Tag num	FIX Field name	Req'd	Comment
10	CheckSum	Y	

3.15 Message Details

3.15.1 How to interpret the Required (Req'd) column

A 'Y' marks the field as required in standard FIX (and of course also in this implementation). A 'Q' means that the field is required in this implementation although it is not required in standard FIX. No entry at all means the field is optional.

3.15.2 Repeating groups

The fields in a FIX Repeating group are marked in the message listings with an arrow. Example (Parties block):

453	NoPartyIDs				Optional repeating group only used for on behalf of transactions.
→	448	PartyID		Q	Party identifier.
→	447	PartyIDSource		Q	Valid values: D = Proprietary/Custom code
→	452	PartyRole		Q	Identifies the type of role for the PartyID specified.
→	802	NoPartySubIDs			Number of PartySubIDs present. Only used for PartyRole=Executing Firm. Will always be 1.
→	→	523	PartySubID	Q	Sub-identifier of party. Here Exchange code of the party.
→	→	803	PartySubIDType	Q	Type of PartySubID (523) value

In the above example nested repeating groups can also be seen.
Also notice that the req'd flag on the NumInGroup field (NoPartyIDs, NoPartySubIDs). If it is present (either Y or Q), it means that the *whole repeating group will always be present*.
A Q or Y set on an individual field in a repeating group means that *it will always be present if the repeating group is present*.

3.15.3 Logon – inbound to Marketplace

The response to a logon is either a Logon, which denotes a successful logon, or a Logout.
A client must be prepared to handle failure scenarios including (but not limited to):
A Logon attempt may fail or be rejected for several reasons. The FIX gateway will react differently depending on the kind of failure. The two different actions it may take are:

Silently ignore the Logon.

- If authentication fails (for security reasons).
- If the wrong Sender or Target CompID is specified.
- For other reasons specified in the standard FIX specifications.
- If the FIX gateway has no connection with the back-end system.

Respond with a Logout.

- Logon failure for other reasons than authentication/security.

The Logout response to a Logon will always contain a note on why in the Text (58) field.

Tag num	FIX Field name	Req'd	Comment
	Standard Header	Y	MsgType = A
98	EncryptMethod	Y	Encryption not supported. Valid values: Valid values: 0 = None / Other
141	ResetSeqNumFlag		Indicates that both sides of a FIX session should reset sequence numbers. **NOTE:** Resetting the sequence numbers will result in all prior messaging being lost. Valid values: Y = Yes
108	HeartBtInt	Y	Heartbeat interval. Any value greater than 10 s is accepted. A lower value will result in a

			Logout response.
553	Username	Q	User name **NOTE:** Must be in CAPTIAL LETTERS.
554	Password	Q	password (unencrypted)
925	NewPassword		*NASDAQ OMX Extension:* Specifies a new password for the FIX Logon. The new password is used for subsequent logons.
	Standard Trailer	Y	

3.15.4 Logon – outbound from Marketplace

Tag num	FIX Field name	Req'd	Comment
	Standard Header	Y	MsgType = A
98	EncryptMethod	Y	Encryption not supported. Valid values: 0 = None / Other
141	ResetSeqNumFlag		Indicates that both sides of a FIX session should reset sequence numbers. Will only be set as a response to an inbound Logon with this flag set. Valid values: Y = Yes
108	HeartBtInt	Y	As specified in inbound Logon. Valid range: Greater than 10 s
1409	SessionStatus	Q	NASDAQ OMX Extension: Status of the FIX session. Valid values: 0 = Session Active 1 = Session password changed 3 = New session password does not comply with policy
	Standard Trailer	Y	

3.15.5 Logout (in/out)

The Logout message is used to gracefully disconnect a FIX session. When receiving a Logout, the counterparty should respond with a Logout. A Logout can also be the response to an unsuccessful Logon attempt.

SessionStatus = 100 means that a critical formatting error has been detected in an inbound transaction. The gateway is unable to reliably continue parsing further messages on the session. The connection is closed and can only be enabled by manual intervention.

Tag num	FIX Field name	Req'd	Comment
	Standard Header	Y	MsgType = 5
1409	SessionStatus		Status of the FIX session. Only set on outbound Logouts. Valid values: 3 = New session password does not comply with policy 4 = Session logout complete 8 = Password expired 100 = NASDAQ OMX Extension: Invalid body length in received message, session suspended 101 = NASDAQ OMX Extension: Heartbeat interval too low.
58	Text		Free text

	Standard Trailer	Y	

3.15.6 Sequence Reset (in/out)

This message has two uses. The common usage is with GapFillFlag set to 'Y', which is used in a response to a Resend Request to indicate that a range of messages will not be resent. This is commonly used to avoid resending administrative messages like Heartbeats.
The other (very rare) usage is to reset the sequence numbers to a higher number to get out of a deadlock. This is only triggered by manual intervention.

Tag num	FIX Field name	Req'd	Comment
	Standard Header	Y	MsgType = 4
123	GapFillFlag		
36	NewSeqNo	Y	
	Standard Trailer	Y	

3.15.7 Resend Request (in/out)

Resend Request is used to recover messages when a sequence number gap has been detected.

Tag num	FIX Field name	Req'd	Comment
	Standard Header	Y	MsgType = 2
7	BeginSeqNo	Y	
16	EndSeqNo	Y	
	Standard Trailer	Y	

3.15.8 Reject (out)

The Reject, or session-level reject, message is sent whenever the FIX gateway is able to at least partially parse the message, but the message does not adhere to the specification and cannot be delivered to the back-end system.

Tag num	FIX Field name	Req'd	Comment
	Standard Header	Y	MsgType = 3
45	RefSeqNum	Y	
371	RefTagID		
372	RefMsgType		
373	SessionRejectReason	Q	Valid values: 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 15 = Repeating group fields out of order 16 = Incorrect NumInGroup count for repeating group 99 = Other

58	Text		
	Standard Trailer	Y	

3.15.9 Heartbeat (in/out)

A heartbeat message is sent at the interval set at Logon. It is also the response to a Test Request message.

Tag num	FIX Field name	Req'd	Comment
	Standard Header	Y	MsgType = 0
112	TestReqID		Identifier included in Test Request message to be returned in resulting Heartbeat. Required when the heartbeat is the result of a Test Request message.
	Standard Trailer	Y	

3.15.10 Test Request (in/out)

Test Request is used to "ping" the counterparty whenever a heartbeat has not arrived at the negotiated heartbeat interval.

Tag num	FIX Field name	Req'd	Comment
	Standard Header	Y	MsgType = 1
112	TestReqID	Y	Identifier included in Test Request message to be returned in resulting Heartbeat
	Standard Trailer	Y	

4 User Authentication

Each incoming business transaction must have a username set in the SenderSubID field. The user needs must be authenticated for the transaction to be accepted. There are two ways to authenticate a user:

- Using the username and password in the Logon message.
- Using the User Request message to authenticate additional users.

A valid username and password is required in the Logon message, so one authenticated user is always available after Logon. Additional User Request messages can be issued to authenticate additional users on the same session.

The SenderSubID field on each incoming business message must be set to an authenticated user.

NOTE: The FIX session must be specifically configured to allow multiple users on the same FIX session. Please contact the marketplace to request such configuration.

NOTE 2: On the Logon or User Request, the SenderSubID must be set to the user id the client intends to log on.

4.1 **User Request**

The User Request message is used to log in or log out a user. A valid, logged in user is required in the SenderSubID field of all incoming business transactions.

4.2 **User Response**

The User Response message is sent as a response to a User Request. Examine the UserStatus (926) field to find out if the request was successful.

4.3 **User Notification**

The User Notification message is an unsolicited message sent when the back-end logs out a user.

4.4 **Password Management**

4.4.1 Renewal of passwords

A new password may be set by setting the NewPassword (925) field along with the current password in Password in the User Request message. The UserStatus (926) field of the User Response returned to the client can be checked to see if the new password was accepted.

4.4.2 Expired passwords

If the password has expired when a client tries to log in, the system will respond with a User Response message with UserStatus set to 101 – Password expired. To gain access, the client must issue a new User Request message with NewPassword set (along with the expired password in Password).

If the new password is not valid, the system will respond with another User Response with UserStatus set to 102 – New session password does not comply with policy. The client will be able to log in again with another new password.

4.5 **Users across multiple sessions**

The back-end does not allow multiple parallel logins for the same user. Whenever an already logged in user attempts to log in a second time, the first is logged out. This is true across protocols as well. If

a user X is logged in on an OMNet session, and the same user tries to log in over FIX, the OMNet user session will be logged out.
So care must be taken not to try to log in the same user across multiple sessions.

4.6 Message Details

4.6.1 User Request (in)

The User Request message is used to authenticate additional users on a FIX session.

Tag num	FIX Field name	Req'd	Comment
	Standard Header	Y	MsgType = BE
923	UserRequestID	Y	Unique identifier for a User Request.
924	UserRequestType	Y	Indicates the action required by a User Request Message. Valid values: 1 = Log on user 2 = Log off user 3 = Change Password For User
553	Username	Y	A valid backend username. **NOTE:** Must be in CAPTIAL LETTERS.
554	Password	Q	
925	NewPassword		New Password
	Standard Trailer	Y	

4.6.2 User Response (out)

The User Response message is a response to the User Request message.

Tag num	FIX Field name	Req'd	Comment
	Standard Header	Y	MsgType = BF
923	UserRequestID	Y	Unique identifier for a User Request.
553	Username	Y	A valid backend username.
926	UserStatus	Q	Indicates the status of a user. Valid values: 1 = Logged In 2 = Not Logged In 5 = Password Changed 6 = Other 101 = Password expired (NASDAQ OMX Extension) 102 = New Password does not comply with policy (NASDAQ OMX Extension)
927	UserStatusText		A text description associated with a user status.
	Standard Trailer	Y	

4.6.3 User Notification (out)

This message is a NASDAQ OMX Extension to FIX 4.4.
The User Notification message is an unsolicited user status message.

Tag num	FIX Field name	Req'd	Comment
	Standard Header	Y	MsgType = CB

809	NoUsernames		Q	Number of user names in this message
→	553	Username	Q	A valid backend username.
926	UserStatus		Q	Indicates the status of a user. Valid values: 7= Forced user logout by Exchange
	Standard Trailer		Y	

5 Business Level Party Identifiers

5.1 Overview

All inbound business messages are subject to marketplace authorization and must therefore specify the party being responsible for the business content of the message. Whenever applicable, the party entering the transaction (if different than business responsible) must also be entered. The SenderCompID and SenderSubID are used to identify the party entering the trade (see implicit parties section below).
The FIX Parties block is used for all other parties.

5.1.1 Parties block

This is a repeating block allowing multiple party identifiers to be set. The following fields must be set for each party:

- PartyID (448) = party identifier
- PartyIDSource (447)
  - D = Proprietary/Custom code
- PartyRole (452) = see below

5.1.1.1 Party Identifier

The PartyID field can contain different types of identifiers. When it contains a member/participant (firm) identifier, the format is as follows:
The party identifier always consists of the two-character market code followed by the up to 5 charcater firm identifier. Example: A participant on the Copenhagen exchange (market code CO) with a firm ID of XYZ, would have the party identifier of “COXYZ”.
Available market codes:

- SE = Stockholm Derivatives
- ST = Nasdaq OMX Stockholm (FI)
- CO = Nasdaq OMX Copenhagen (FI)
- HE= Nasdaq OMX Helsinki (FI)
- RI = Nasdaq OMX Riga (FI)
- VI = Nasdaq OMX Vilnius (FI)
- TA = Nasdaq OMX Tallinn (FI)
- NC = Nasdaq OMX Commodities

NOTE: This party identifier scheme is also used for SenderCompID.

5.1.2 Root Parties block

In some messages a repeating group called *Root Parties* is used instead of Parties. The contents are exactly the same as for the Parties block, but the tags have new numbers, and the names of the tags are all prefixed with “Root”. The reason for this is that in some FIX messages the Parties block is in use in a repeating group. In such cases the Root Parties block is attached to the root level of the message and used instead. Currently, the Root Parties block is in use in the following messages:

- New Order List (used for linked orders).
- Trade Capture Report

5.2 Implicit parties

All inbound business messages must contain:

- SenderCompID (49) = party identifier of the firm entering the transaction (see section 5.1.1.1).
- SenderSubID (50) = set to a valid already authenticated username (see chapter 4 for details).

These fields implicitly identify the firm and individual entering the business message. So for all non-on-behalf-of messages, the Parties block can be omitted from the message.
NOTE: For all on-behalf-of transactions, the entering party is set in the implicit parties (SenderCompID and SenderSubID) and the executing party identifiers is set in the Parties or Root Parties block.

5.3 Available Party Roles

The following roles are used:

	Business Role	PartyRole (452)	Comment
Transaction owner = party legally responsible for consequences of the message	Firm	1 = Executing Firm	Implicit for all transactions other than on-behalf-of or trade reports. Reporting party in trade reports.
	Individual user	12 = Executing Trader	Implicit for all transactions other than on-behalf-of.
Counterparty in Trade Capture Reports	Firm	17 = Contra Firm	Counterparty in Trade Capture Reports.
Clearing Firm	Firm	4 = Clearing Firm	Optional for all orders and trades. **NOTE:** Handled differently depending on if the instrument is cleared within the system or not (see section 6.3 for details).
Clearing Account	Account	83 = Clearing Account	Optional Clearing Account **NOTE:** Handled differently depending on if the instrument is cleared within the system or not (see section 6.3 for details).
Confirmed by Firm	Firm	1001 = Confirmed by Firm	*NASDAQ OMX Extension:* Contains Broker Firm ID for trades entered on the recipient's behalf. **NOTE:** Only in use for Commodities.

NOTE: individual users are not used to identify reporting party or counterparty in Trade Capture Reports.

5.4 On Behalf of Identifiers

- All inbound business messages sent on behalf of another party must include the Parties block. Two parties must be present in each on behalf of transaction: PartyRole = 1, Executing Firm set to the id of the firm the transaction is entered on behalf of (legal owner).
- PartyRole = 12, Executing Trader set to the id of the trader the transaction is entered on behalf of (legal owner).
  The party entering the transaction is set in the implicit party identifier fields (SenderCompID and SenderSubID).

6 Order Management

6.1 **Overnight orders**

Clients who wish to send overnight orders need to make sure that the ClOrdID is *unique across the entire lifetime of the order*. A simple solution is to include a date in the ClOrdID.

6.2 **Pass-thru fields**

Genium INET primarily offers three fields as pass-thru fields on incoming transactions. The values of those fields are echoed back to the client in subsequent outgoing transactions. The fields are:

Field	Tag	Mapped to	Length	Comment
Account	1	ex_client	10	This field can be used for any purpose. The value need not be an actual account. **NOTE:** If Account is to be used as an actual account, the letters should be capitals only.
AllocID	70	customer_info	15	**NOTE:** The AllocID field can be overwritten by the clearing subsystem in *post-trade* transactions such as give-ups.
OrderReference	20009	exchange_info	10	Is not available on cancel requests.

NOTE: The pass-thru fields are *not* echoed back on rejects.

6.3 **Clearing Accounts**

Clearing Account is generally split into two parts:

- Clearing Firm
- Clearing Account

The fields used for Clearing Account information is a bit different in different scenarios. Clearing Account information is carried in different fields depending on:

- The direction of the message (inbound to the exchange or outbound)
- The type of message
- If the instrument traded is cleared within the system or not

The following matrix shows which fields should be used in which scenario:

Messages	Direction	Instr. Is cleared	Clearing Account	Clearing Firm	Comment
New Order Single Cancel Replace Request Execution Report	IN/OUT	Yes	Account (1)	implied	Clearing Firm is the same as the owner of the order.
	IN/OUT	No	PartyID *(PartyRole=83)*	PartyID *(PartyRole=4)*	
Trade Reports	IN	Yes	Account (1)	implied	Clearing Firm is the same as the owner of the trade.

	IN	No	PartyID *(PartyRole=83)*	PartyID *(PartyRole=4)*	The fields are pure pass-thru
Rectify Trade	IN	Yes	AllocAccount (79)	Nested2PartyID *(Nested2PartyRole=4)*	Can only be used with instruments cleared within the system
Give-ups (Allocation Instruction, Allocation Report)	IN/OUT	Yes	AllocAccount (79)	NestedPartyID *(NestedPartyRole=14)*	Can only be used with instruments cleared within the system
Trade Confirmations	OUT	Yes/No	PartyID *(PartyRole=83)*	PartyID *(PartyRole=4)*	Same fields, but slightly different use: for non-cleared instruments the fields are pass-thru. Max lengths of the fields also differ slightly.

In some cases the Clearing Account fields can be used as pass-thru fields carry any client-specific information.

NOTE: The length limitations for the above fields differ between cleared and non-cleared instruments. See below for details.

6.3.1 Instruments cleared within the system

Within the Genium INET system, financial and commodity derivatives are cleared.
Internally, the Genium INET system represents Clearing Accounts as a concatenation of:
exchange code (2 chars), participant id (5 chars) and the actual account (10 chars). Example: *NC ABC ACC123*.

In FIX, the first two parts is concatenated to form the party indentifier (*NCABC* in the example above). See section 5.1.1.1for details. The last part, the actual account (in capital letters) is carried in a separate field.

- In order related transactions (in and out), the actual account is carried in the Account (1) field. The Clearing Firm part is implied.
- Trade reports (Trade Capture Report) sent into the system uses the same fields as order messages.
- In trade confirmations (Trade Capture Report) the account information is carried as PartyIDs with PartyRole set to Clearing Firm (4) or Clearing Account (83).
- In rectify trades (Trade Capture Report), AllocAccount (79) is used for the account, and Nested2PartyID (757) contains the Clearing Firm with Nested2PartyRole set to Clearing Firm (4).
- In Give-up transactions (Allocation Instruction, Allocation Report), AllocAccount (79) is used for the account, and NestedPartyID (524) contains the Clearing Firm with NestedPartyRole set to Giveup Clearing Firm (14).

Pre-trade, the Account field may be used as a pass-through field. **NOTE:** trade confirmations (Trade Capture Reports) will carry the actual account in the Account field.

6.3.1.1 Field length limitations for cleared instruments

PartyID	Example	Length
Party identifier	NCABC	7
actual account	ACC12345	10

6.3.2 Instruments NOT cleared within the system

For instruments not cleared within the system, such as Fixed Income instruments, the clearing account and clearing firm PartyIDs can be used as pass-thru fields. The system does not use these fields in any way.

Internally, the values will be mapped to different parts of the exchange_info_s field (Omnet).

For all in- and outbound transactions, the following two fields are used to represent Clearing Firm and Clearing Account:

- PartyID with PartyRole set to Clearing Firm (4) is used for Clearing Firm.
- PartyID with PartyRole set to Clearing Account (83) is used for the Clearing Account.

NOTE: For instruments not cleared in Genium INET, the Account (1) field can be used as a pass-through field.

6.3.2.1 Field length limitations for non-cleared instruments

The following max length limitations exist:

PartyID w. PartyRole set to	Length
4 – Clearing Firm	4
83 – Clearing Account	12

6.4 Instrument Identifiers

For any trading system, the correct identification of securities in a FIX message is of utmost importance. There are several fields within each FIX message, incoming or outgoing, that allow for identification of securities. In this implementation two alternative identifiers can be used:

- Symbol (55) which should contain the OMNet short name (ins_id_s) for the security.
- SecurityID (48) containing the Orderbook ID of the security. This is an alternative numeric identifier that can be used instead of Symbol. **NOTE:**
  - The Orderbook ID identifier is **not** provided via OMNet Reference Data.
  - The Orderbook ID *can* be different across trading days for the same security.

6.5 Multileg Orders

A multileg security is made up of multiple securities that are traded atomically. Swaps, option strategies, futures spreads, are a few examples of multileg securities. The requirement that all legs be traded in the quantities that make up the multileg security is the important distinction between a multileg order and a list order.

The trading models supported for multileg securities in this solution are:

Pre-defined Multileg Security Model
A.k.a. *Standard Combinations*. Marketplace-defined multileg securities made available for trading. In Genium INET these securities are set up and traded like any other instrument.
User-defined Multileg Security Model
Also known as *Tailor-made Combinations* (TMC). These are user-defined multileg securities made available for trading by the marketplace.

Both models results in ordinary orderbooks traded like any other instrument using ordinary Order Entry transactions such as the New Order Single. The exception is fills, where the execution reports contain a repeating group with the fill details per leg. See chapter 7, Multileg Orders for additional details.

6.6 Main Workflow

6.6.1 New Order

The order workflow starts with the user submitting a New Order Single message. In response an Execution Report is produced. The Execution Report is a reply directed to the sender of the order and will contain details of the order. If the order is rejected the Execution Report will contain relevant error messages.

6.6.2 Fills

When an order is filled the Execution Report will contain details about the fill. See section 6.11.15 for message details. In addition, a Trade Capture Report will be produced. The principal differences between the two are:

Execution Reports are messages directed to the sender of the order and are primarily intended for front-office purposes. It captures order status information as well as fills information (if applicable).

Trade Capture Reports are messages capturing the trade as such and is primarily intended for downstream processing. The Trade Capture Report is used to inform a variety of parties about a trade, e.g.: broker back office; clearing firms; clearing houses; depositories and; regulators. As such downstream processing occurs at various locations and for different purposes, the Trade Capture Report message might look slightly different depending on the receiver.
Trade Capture Report messages are also used for a large number of other purposes, including reporting of privately negotiated trades and relaying trades to parties not directly involved in the trade – but this is outside the scope of this chapter.
Trade reversals and corrections are only sent as Trade Capture Reports.

6.6.2.1 Trade Match ID

The TrdMatchID (880) contains the match id generated by the system. TrdMatchID will hold a 16 byte Base64-encoded string based on the 12 first bytes of the 16 byte binary match_id.
The encoding is performed according to RFC 2045 [2].
NOTE: TrdMatchID is also set in Trade Capture Report confirmation messages.

6.6.3 Order Modification

Order modification is accomplished through the use of the Order Cancel Replace Request message. Despite its name, it represents a modification of the existing order, not removing the old order and replacing it with a new one. However, an order modification is not a delta change to order instructions; the values set in the Cancel Replace represent the requested new order state. An Execution Report will relay the new state of the order.

- Fields not set in the Cancel Replace are *assumed to keep their previous values*.
- The required fields must be set regardless if they are changed or not.

6.6.3.1 Order Attributes allowed to change

Although FIX allows for virtually all of the Order attributes to be changed, there are limitations as to what the back-end Genium INET system allows. The following attributes are allowed to change:

- OrderQty (38)
- MaxFloor (111)
- TimeInForce (59) together with ExpireDate (432) or TradingSessionID (336)
- Account (1), pass-thru field
- AllocID (70), pass-thru field
- Price (44)
- OrderCapacity (528)
- OrderRestrictions (529)
- OrderReference (20009), pass-thru field
- PartyID (where PartyRole is *Clearing Firm* or *Clearing Account*)

NOTE: Any change to the price of an order, or increasing quantities will result in the order losing its priority in the market.
NOTE 2: Modifying an order to TimeInForce = IOC or FoK is not allowed.
NOTE 3: Modifying the price of an order to a zero is not allowed. If a zero price is desired, the order has to be deleted and a new order with price 0 entered.
NOTE 4: If MaxFloor or TimeInForce are not intended to be changed, **do not include them** in the Cancel Replace message. They may cause the order to loose priority or the Cancel Replace to be rejected.

6.6.3.2 Restatements

The Execution Report – Restatement message is used for restating the overnight orders (GTC/GTD) in the morning, In this case, the ExecRestatementReason will be set to 1 = GT renewal / restatement (no corporate action). See section 6.11.14 for message details.

6.6.3.3 Unsolicited modification of orders entered via FIX

Orders entered via FIX can be modified via other protocols like Omnet. It may also be possible for the marketplace to modify existing orders. In such an event an Execution Report – Unsolicited Order Update will be sent out over FIX. See section 6.11.13 for message details.

6.6.4 Order Cancellation

- If the user wishes to cancel a single previously sent order, the Order Cancel Request message is used.
- Execution Reports are issued relaying the status of every canceled order.
- In some cases orders may be cancelled in the system without prior request by the user. These will be sent as an Execution Report – Unsolicited Cancel to the client.
- The system will generate cancel messages (Execution Report –IOC/Fok Order Cancel) for every IOC and FoK order.
- The system will generate cancel messages (Execution Report – Market-to-Limit Order Cancel) for Market-to-Limit orders that could not be immediately matched.
- The Order Cancel Request cannot be used for partial cancels.

6.6.4.1 Cancellation of orders **not** sent in via FIX

It is possible via FIX to cancel orders originally entered via Omnet or by other means. To cancel such an order, the correct OrderID (Omnet order number), instrument identifier (Symbol or SecurityID) and Side need to be supplied. In this case the OrigClOrdID shall be set to "NONE".

6.6.4.2 Unsolicited cancellation of orders entered via FIX

Orders entered via FIX may be cancelled via other protocols like Omnet, or possibly by the marketplace. In such an event an Execution Report – Unsolicited Cancel will be sent out over FIX. See section 6.11.12 for message details.

6.6.5 Order suspension/inactivation at connection loss

The back-end can be configured to suspend outstanding orders if a FIX session is disconnected for a configurable interval. Three options are available:

- Do not suspend on disconnect
- Suspend *all* outstanding orders
- Suspend outstanding orders except for overnight orders (GTC/GTD).

Upon reconnection, Execution Reports will be sent out for all suspended orders. The Execution Reports will have OrdStatus set to 9 – Suspended. See section 6.11.16 for message details.
Suspended orders may be cancelled using ordinary Order Cancel Request messages. Suspended order cannot be activated again.
NOTE: The Execution Report –Order Suspended will not contain TargetSubID (57).
NOTE 2: Suspended orders will be cancelled at end-of-day.

6.7 Order Features

6.7.1 Order Identification

6.7.1.1 Client Order ID

Any message related to an order (entry, cancellation, modification) sent by the client, must have a unique identifier in the ClOrdID (11) field. As the standard indicates, the uniqueness of these identifiers must be maintained during the trading session. If orders with duration of more than one trading session are used, the sender needs to cater for uniqueness across those.

Once the message is accepted by the trading engine, the client receives the corresponding confirmation message with the same ClOrdID. In cases where the user immediately after sending an order wants to modify or cancel it, this can be achieved by referring to the initial order in the OrigClOrdID (41) field of the subsequent message.

Client Order IDs when the Firm uses multiple FIX sessions
Firms using multiple front-end trading applications or multiple FIX sessions should be aware of the following:

- In cases where the exchange offers drop copies of Execution Reports to FIX sessions other than the one that submitted the order, those drop copy Execution Reports will not contain a ClOrdID. The reason for excluding the ClOrdID in those cases is that various FIX sessions or the underlying trading applications might use conflicting ClOrdIDs.
- The above may also apply in cases where exchange business operations perform order management on behalf of the order owner.

6.7.1.2 Order ID

The OrderID (37) field is the order identifier assigned by the marketplace. This identifier is static and stays with the order even when it is modified.

NOTE: Genium INET OrderIDs are only unique *per orderbook and Side*. So a buy and a sell order in the same orderbook may have the same OrderID. Care must be taken to base identification of orders on OrderID, orderbook id (SecurityID/Symbol), and Side.

Users are encouraged to provide the OrderID instead of OrigClOrdID (41) on order updates and cancellations whenever possible, i.e. in all cases except for submitting order actions before the new order ack (Execution Report) is received. The OrderID is the preferred identifier for order modification and cancellation as it is the identifier used internally in the trading engine. Use of other identifiers requires a lookup which increases message latency.

Note that the OrigClOrdID field is required in standard FIX both in Cancel Replace messages and Cancels. If you wish to use the OrderID, it is recommended to set the OrigClOrdID to “NONE” (excluding the quotation marks). The system will ignore OrigClOrdID if OrderID is set in a Cancel or Cancel Replace Request.

As use of the OrderID requires the user to wait for an order acknowledgement from the trading engine, immediate actions require the use of the OrigClOrdID (41) reference field. This field could be necessary to identify the order in communications with the market by other means than FIX.

6.7.1.3 Execution ID

The ExecID (17) field is not an identifier of trades. It is an identifier assigned to each unique Execution Report message produced by the marketplace, without duplicates during the entire FIX session. The ExecID will be an integer value.

6.7.1.4 ExecType

When a fill occurs, the ExecType (150) field will be set to F = Trade.
NOTE: Post-trade corrects or reversals will not be represented on Execution Reports. Please refer to Trade Capture Reports for such functionality.

6.7.2 Order States

Order state changes are divulged in Execution Report messages. Every state change is communicated in an Execution Report.

An order can be in the following intermediate states:

- **New**. This state is applicable when an order is accepted by the trading engine and is not immediately transitioned into any other state:
  - The order is put on the book but not (partially) filled
  - The order is held outside the book waiting for activation, e.g. due to a stop condition or for a session change (as e.g. for a Trigger order).
- **Partially filled**.

The following are final states, indicating that the order is no longer in the book and no longer available for updates or status requests:

- **Rejected**. The order did not pass validation rules.
- **Canceled**. The order was removed from the system due to a cancellation request, or due to TimeInForce reasons.
- **Filled**. The order is completely filled.

- **Expired**. When a GTD order expires.
- **Suspended**. The order was suspended due to connection loss.

6.7.3 Order Types

Order type is set in the OrdType (40) field. Three order types are supported:

- Market
- Limit
- Market-to-Limit (called Market with leftover as limit in FIX).

6.7.3.1 Market Orders

Market orders are always executed at the best possible price. A market order will trade through as many price-levels as needed to be fully filled.

In continuous trading a market order cannot be stored in the book. It has to have a TimeInForce of IOC or FoK.

Market orders may be allowed to enter the book in non-matching states. Once the session changes to a matching state, the order will be executed and/or cancelled.

6.7.3.2 Market-to-Limit Orders

A Market-to-Limit order is a market order where the remaining quantity is placed in the book at the price which part of the order was executed. If there is no order on the opposing side, the Market-to-Limit order will be cancelled immediately.

In comparison to a Market order, the Market-to-Limit order only executes at the best price level and therefore does not trade through the book.

By setting TimeInForce to IOC or FoK, the Market-to-Limit order will behave like a Market order but only match at the highest price level.

NOTE: Once the order is converted to a Limit order the OrdType field of subsequent Execution Reports will be set to Limit (including the Order Ack), and the Price field set to the price of the execution.

6.7.4 Order Expiry

An order can specify various conditions for when or how it should expire or be automatically removed from the book.

The morning after a GTD order has expired, an Execution Report with OrdStatus (39) set to Expired will be sent out for that order. See section 6.11.17 for message details.

A GTC order can also expire. Example: A GTC order is suspended. If it isn't deleted or reactivated (not possible via FIX) the same day, an ER with OrdStatus set to Expired will be sent out the next day.

NOTE: Only if a GTC order expires because the instrument expires intra-day, an order expired transaction will be sent out.

Supported TimeInForce (59) values:

Value	Name	Comment
0	Day	
1	Good Till Cancel (GTC)	
3	Immediate Or Cancel (IOC)	
4	Fill or Kill (FoK)	FoK orders cannot have MatchIncrement (1089) set.
6	Good Till Date (GTD)	GTD orders must have ExpireDate (432) set.
S	Good till End of Session (GTS)	Nasdaq OMX Extension: Order expires at the end of the *first* session with the state type set in

		TradingSessionID (336). See detailed description below.

6.7.4.1 Good till End of Session (GTS)

GTS is a NASDAQ OMX Extension to FIX. A GTS order is similar to a GTD order, but instead of setting the date of expiry, the session type when the order shall expire is set. TimeInForce nust be set to S.

A GTS order is valid until the end of the *first* session of the type given in TradingSessionID (336).
The valid session types are:
1 = Auction
2 = Continuous Trading
The session type is used to group several sessions with equal order behavior into one entity to be used for session related order handling. In the cases where the same session type is run multiple times during a day, the order will expire on the earliest possible occasion.

6.7.5 Quantity Conditions

An order can specify various types of quantity conditions.
Match Increment (1089). In this solution, MatchIncrement is used to enter Block or Odd Lot orders. By setting MatchIncrement to the number of units configured as the Block Lot/Odd Lot for the orderbook of the order (Lot size), a block lot/odd lot order can be entered. The Lot sizes for a particular instrument must be looked up in Reference Data (this is **not** provided via FIX). An order with an incorrect MatchIncrement value will be rejected.
If MatchIncrement is not set, the order defaults to Round Lot.
NOTE: MatchIncrement is not allowed for FoK orders.

All-or-None (AON) is an instruction to fill an order completely or not at all; similar to a FoK but it remains in the book if not executed immediately. Tag 18, ExecInst is used for AON orders.
There are certain (logical) restrictions on how to use the AON flag in combination with other order features:

- An AON order should not have TimeInForce set to IOC. This will be interpreted as an FOK order. TimeInForce will be set to FOK, and the AON flag will not be present on the Order Ack returned.
- AON can be combined with TimeInForce=FOK, but this will be interpreted as an FOK order without the AON flag set (the definition of FOK can be said to include AON). The AON flag will not be returned in the Order Ack.
- AON orders cannot have MatchIncrement set.

Reserve Quantity (a.k.a. “Hidden” or “Iceberg”) Orders allow users to hide the full size of their order and thereby potentially limit its influence on prices.

MaxFloor (111): Used to indicate the maximum order quantity shown in the public Market Data.
NOTE: MaxFloor = 0, a completely hidden order, is **not** supported in Genium INET. Setting MaxFloor to zero will make the full order visible.
NOTE 2: MaxFloor in combination with FoK or IOC orders is not allowed.
NOTE 3: MaxFloor has been changed to behave as expected in standard FIX. The expected behavior is for MaxFloor value to be decreased when the order is partially traded. The previous implementation kept MaxFloor on the original value.

6.7.6 Triggering Instructions

The Triggering Instructions block in FIX is used to express predefined automatic order modifications. Triggers can act on different events. The TriggerType (1100) field determines what should trigger a change. The only action supported is for the triggered order to be *activated*. The trigger order remains hidden and inactive until the trigger condition is met. When the trigger hits, the order is either traded or inserted into the book as if it was a new order.
Only one triggering instruction is allowed per order. All the order attributes available for a "normal" order (e.g. Order Type, Time In Force etc) are supported for the order to be triggered.

NOTE: Trigger orders will be removed at the end of day if the triggering condition has not yet been met.

The following fields can be used:

Tag no	Name	Comment
1100	TriggerType	Determines what should trigger an order modification. Valid values: 2 = Specified Trading Session 4 = Price Movement
1101	TriggerAction	Defines the type of action to take when the trigger hits. Valid values: 1 = Activate
1102	TriggerPrice	A specified limit price to validate against price movements –the trigger hits when the price is reached.
1113	TriggerTradingSessionID	Identifier of the session type when the order is to be triggered. Valid values: 1 = Auction 2 = Continuous Trading
1103	TriggerSymbol	Symbol used for price triggers
1104	TriggerSecurityID	Identifier of the security used for price triggers.
1105	TriggerSecurityIDSource	Valid values: M = Marketplace-assigned identifier
1107	TriggerPriceType	Determines what price should be tracked for price movements. Valid values: 2 = Last Trade
1109	TriggerPriceDirection	Used to specify if the trigger should hit only on rising (Up) or falling (Down) prices. Valid values: U = Trigger if the price of the specified type goes UP to or through the specified Trigger Price D = Trigger if the price of the specified type goes DOWN to or through the specified Trigger Price.

6.7.6.1 Triggering off Session Changes

Triggering off Session Changes can be used to activate an order at a specified session. The following fields need to be set:

- TriggerType (1100) set to *Specified Trading Session*
- TriggerAction (1101) set to *Activate*
- TriggerTradingSessionID (1113) Identifier of the trading session to activate order at. Currently two triggers are supported 1 –Auction, and 2 – Continuous Trading.

6.7.6.2 Price Triggering

When an order is matched, stored, altered, expired or deleted affecting the Best Bid Offer (BBO) or the Last Match Price of the matching engine, the system checks for any “non-triggered” orders having a condition that is now met. It is possible to trigger off price movements occurring in the same orderbook or in a different orderbook.

The following fields must be set for a price trigger:

- TriggerType (1100) set to *Price Movement*
- TriggerAction (1101) set to *Activate*
- TriggerPrice (1102) set to the triggering price
- TriggerSymbol (1103) OR TriggerSecurityID+TriggerSecurityIDSource
- TriggerSecurityID (1104) set to the triggering instrument
- TriggerSecurityIDSource (1105)
- TriggerPriceType (1107) to specify the price type; *last trade* is currently the only option.
- TriggerPriceDirection (1109) to indicate price movement direction

6.7.6.3 Triggering workflow

A trigger order can go into three different states at entry:

- Not activated – the order is not immediately triggered, and is placed outside of the book waiting to be triggered.
- Immediately activated, immediately filled
- Immediately activated, placed on book – the order is immedialtely triggered but does not immediately trade.

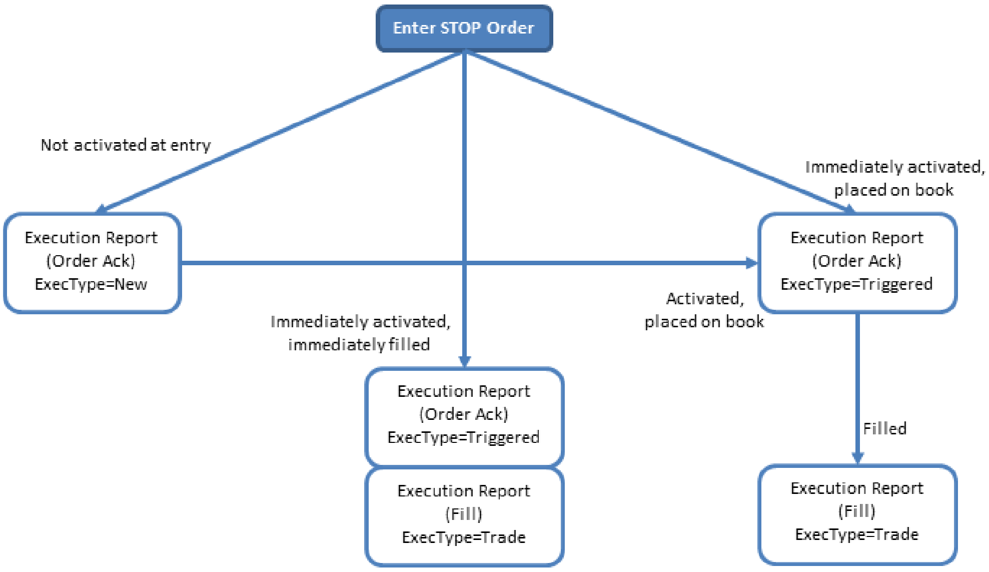


Figure 1, Trigger Order states and resulting messages

To understand the state of a trigger order, the client needs to examine the ExecType (150) field of the Execution Report messages received.

- ExecType=New (0) means the order was not activated on entry.
- ExecType=Triggered (L) means the order was activated at entry.

ExecType=Trade (F) means the order was partially or fully traded. A trigger order will always be activated before it trades.

6.7.6.4 Cancellation of Trigger Orders

An order with a trigger condition can be cancelled using the ordinary Order Cancel Request message. As a consequence of the back-end having different cancel messages for triggered and not yet triggered orders, it is unlikely but possible for a cancel of a non-triggered order to be rejected while the order is left in the book. This can only happen *once*, if the order is triggered while the cancel is sent in. *In this (unlikely) event, a second cancel must be sent for the same order.*

6.8 Missing required fields in Rejects

Due to the way the back-end works, certain fields required in standard FIX 4.4 for application-level rejects will be missing.
For Order rejects (Execution Report – reject), the following required field will not be present:
Side (54)

Also note that on Execution Report –reject messages, the Symbol field (55) will be set to "[N/A]".

6.9 Business Message Reject

The Business Message Reject is used to report rejections in situations where other reject messages are not available, e.g. when the inbound message does not reach the trading engine due to trading being closed or authorization not sufficient. See section 6.11.18 for message details.
NOTE: The user must be prepared to receive this message as an alternative response to all other business messages.

6.10 How to interpret the message details listings

6.10.1 How to interpret the Required (Req'd) column

A 'Y' marks the field as required in standard FIX (and of course also in this implementation). A 'Q' means that the field is required in this implementation although it is not required in standard FIX. No entry at all means the field is optional.

6.10.2 Repeating groups

The fields in a FIX Repeating group are marked in the message listings with an arrow. Example (Parties block):

453	NoPartyIDs				Optional repeating group only used for on behalf of transactions.
→	448	PartyID		Q	Party identifier.
→	447	PartyIDSource		Q	Valid values: D = Proprietary/Custom code
→	452	PartyRole		Q	Identifies the type of role for the PartyID specified.
→	802	NoPartySubIDs			Number of PartySubIDs present. Only used for PartyRole=Executing Firm. Will always be 1.
→	→	523	PartySubID	Q	Sub-identifier of party.
→	→	803	PartySubIDType	Q	Type of PartySubID (523) value

In the above example nested repeating groups can also be seen.
Also notice that the req'd flag on the NumInGroup field (NoPartyIDs, NoPartySubIDs). If it is present (either Y or Q), it means that the *whole repeating group will always be present*.

A Q or Y set on an individual field in a repeating group means that *it will always be present if the repeating group is present*.

6.11 **Message Details**

6.11.1 New Order Single –inbound to Marketplace (in)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = D
11	ClOrdID	Y	Unique identifier set by the client.
453	NoPartyIDs		Optional repeating group used for on behalf of transactions and/or for optional Clearing Firm and Clearing Account.
→	448 PartyID	Q	Party identifier.
→	447 PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452 PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 83 = Clearing Account
1	Account		Optional pass-thru field set by client and echoed back by marketplace.
111	MaxFloor		For hidden orders.
21	HandlInst		Instructions for order handling on Broker trading floor. Valid values: 1 = Automated execution order, private, no Broker intervention (default value)
18	ExecInst		Valid values: G = All or None (AON)
55	Instrument/Symbol		OMNet short name. Either Symbol or SecurityID+SecurityIDSource must be set.
48	Instrument/SecurityID		Orderbook ID
22	Instrument/SecurityIDSource		Valid values: M = Marketplace-assigned identifier
54	Side	Y	Valid values: 1 = Buy 2 = Sell
60	TransactTime	Y	
38	OrderQtyData/OrderQty	Y	
40	OrdType	Y	Valid values: 1 = Market 2 = Limit K = Market With Left Over as Limit (market order with unexecuted quantity becoming limit order at last price)
44	Price		Required for Limit orders
1100	TriggeringInstruction/ TriggerType		Determines what should trigger an order modification. Valid values: 2 = Specified Trading Session 4 = Price Movement
1101	TriggeringInstruction/ TriggerAction		Defines the type of action to take when the trigger hits. Valid values:

			1 = Activate
1102	TriggeringInstruction/ TriggerPrice		A specified limit price to validate against price movements –the trigger hits when the price is reached.
1113	TriggeringInstruction/ TriggerTradingSessionID		Identifier of Trading Session when the order is to be triggered. Valid values: 1 = Auction 2 = Continuous Trading
1103	TriggeringInstruction/ TriggerSymbol		Symbol used for price triggers
1104	TriggeringInstruction/ TriggerSecurityID		Identifier of the security used for price triggers.
1105	TriggeringInstruction/ TriggerSecurityIDSource		SecurityIDSource of the instrument used for price triggering. Valid values: M = Marketplace-assigned identifier
1107	TriggeringInstruction/ TriggerPriceType		Determines what price should be tracked for price movements. Valid values: 2 = Last Trade
1109	TriggeringInstruction/ TriggerPriceDirection		Used to specify if the trigger should hit only on rising (Up) or falling (Down) prices. Valid values: U = Trigger if the price of the specified type goes UP to or through the specified Trigger Price. D = Trigger if the price of the specified type goes DOWN to or through the specified Trigger Price.
59	TimeInForce	Q	Valid values: 0 = Day 1 = Good Till Cancel (GTC) 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FoK) 6 = Good Till Date (GTD) S = NASDAQ OMX Extension: Good till End of Session (GTS)
386	NoTradingSessions		Only set for GTS orders. Can only be set to 1.
→	336 TradingSessionID	Q	State type of order expiration. Conditionally required if TimeInForce = GTS. Valid values: 1 = Auction 2 = Continuous Trading Period
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD

528	OrderCapacity		Designates the capacity of the firm placing the order. Valid values:
			P = Principal
			A = Agency
			R = Riskless Principal
			NOTE: Required for Fixed Income.

529	OrderRestrictions		Restrictions associated with an order. Valid values:
			B = Issuer Holding **(requires 528=A)**
			C = Issue Price Stabilization **(requires 528=P)**
			5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**

70	AllocID		Optional pass-thru field set by client and echoed back by marketplace.
1089	MatchIncrement		NASDAQ OMX Extension: If set, must be set to correct block lot/odd lot size, otherwise it will be rejected.

20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field.
	Standard Trailer	Y	

6.11.2 Order Cancel Request (in)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = F
41	OrigClOrdID	Y	Set to "NONE" if using OrderID instead.
37	OrderID		Recommended to be used instead of OrigClOrdID.
11	ClOrdID	Y	Unique identifier set by the client.
453	NoPartyIDs		Optional repeating group only used for on behalf of transactions.
→	448 PartyID	Q	Party identifier.
→	447 PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452 PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 12 = Executing Trader
55	Instrument/Symbol		OMNet short name. Either Symbol or SecurityID+SecurityIDSource must be set.
48	Instrument/SecurityID		Orderbook ID
22	Instrument/SecurityIDSource		Valid values: M = Marketplace-assigned identifier
54	Side	Y	Valid values: 1 = Buy 2 = Sell
60	TransactTime	Y	
38	OrderQtyData/OrderQty	Y	**NOTE:** Required in FIX but ignored by the system. Partial cancels are not supported.
	Standard Trailer	Y	

6.11.3 Order Cancel Replace Request (in)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = G
37	OrderID		Recommended to be used instead of OrigClOrdID.
453	NoPartyIDs		Optional repeating group used for on behalf of transactions and/or for optional Clearing Firm and Clearing Account.
→	448 PartyID	Q	Party identifier.
→	447 PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452 PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 83 = Clearing Account
41	OrigClOrdID	Y	ClOrdID of the order to modify/cancel. Set to "NONE" if using OrderID instead.

11	ClOrdID	Y	Unique identifier set by the client.
1	Account		Optional pass-thru field set by client and echoed back by marketplace.
70	AllocID		Optional pass-thru field set by client and echoed back by marketplace.
111	MaxFloor		For hidden orders.
21	HandlInst		Instructions for order handling on Broker trading floor. Valid values: 1 = Automated execution order, private, no Broker intervention (default value)
18	ExecInst		Valid values: G = All or None (AON)
55	Instrument/Symbol		OMNet short name. Either Symbol or SecurityID+SecurityIDSource must be set.
48	Instrument/SecurityID		Orderbook ID
22	Instrument/SecurityIDSource		Valid values: M = Marketplace-assigned identifier
54	Side	Y	Required in FIX, but not allowed to change Valid values: 1 = Buy 2 = Sell
60	TransactTime	Y	
38	OrderQtyData/OrderQty	Y	
40	OrdType	Y	Required in FIX, but not allowed to change Valid values: 1 = Market 2 = Limit
44	Price		Required for Limit orders
59	TimeInForce		Valid values: 0 = Day 1 = Good Till Cancel (GTC) 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FoK) 6 = Good Till Date (GTD) S = NASDAQ OMX Extension: Good till End of Session (GTS)
386	NoTradingSessions		Only set for GTS orders. Can only be set to 1.
→	336 TradingSessionID	Q	State type of order expiration. Conditionally required if TimeInForce = GTS. Valid values: 1 = Auction 2 = Continuous Trading
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD

528	OrderCapacity		Designates the capacity of the firm placing the order.
			Valid values:
			P = Principal
			A = Agency
			R = Riskless Principal
			NOTE: Required for Fixed Income.

529	OrderRestrictions		Restrictions associated with an order. Valid values:
			B = Issuer Holding **(requires 528=A)**
			C = Issue Price Stabilization **(requires 528=P)**
			5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**

20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field.
	Standard Trailer	Y	

6.11.4 Order Cancel Reject (out)

Purpose: Reject of Order Cancel Replace Request.
Identified by: MsgType = 9 AND CxlRejResponseTo = 1

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = 9
37	OrderID	Y	From Cancel, or if CxlRejReason=1 – Unknown order, OrderID will be set to "NONE".
11	ClOrdID	Y	Unique identifier set by the client.
41	OrigClOrdID	Y	ClOrdID of the order to modify/cancel. Will be set to "NONE" for orders not originally entered via FIX, or if the order could not be found.
39	OrdStatus	Y	Valid values: 0 = New 1 = Partially filled 2 = Filled 4 = Canceled 8 = Rejected 9 = Suspended C = Expired
60	TransactTime	Q	
434	CxlRejResponseTo	Y	Valid values: 1 = Order cancel request
102	CxlRejReason		Valid values: 0 = Too late to cancel 1 = Unknown Order 2 = Broker / Exchange Option 6 = Duplicate ClOrdID (11) received
58	Text		Error description
	Standard Trailer	Y	

6.11.5 Order Cancel Reject – Cancel Replace (out)

Purpose: Reject of Order Cancel Replace Request.
Identified by: MsgType = 9 AND CxlRejResponseTo = 2

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = 9
37	OrderID	Y	From C/R, or if CxlRejReason=1 – Unknown order, OrderID will be set to "NONE".
11	ClOrdID	Y	Unique identifier set by the client.
41	OrigClOrdID	Y	ClOrdID of the order to modify/cancel. Will be set to "NONE" for orders not originally entered via FIX, or if the order could not be found.
39	OrdStatus	Y	Valid values: 0 = New 1 = Partially filled 2 = Filled 4 = Canceled 8 = Rejected 9 = Suspended C = Expired

60	TransactTime	Q	
434	CxlRejResponseTo	Y	Valid values: 2 = Order cancel/replace request
102	CxlRejReason		Valid values: 0 = Too late to cancel 1 = Unknown Order 2 = Broker / Exchange Option 6 = Duplicate ClOrdID (11) received
58	Text		Error description
	Standard Trailer	Y	

6.11.6 Execution Report – Order Ack (out)

Purpose: Order Acknowledgement.

Identified by: MsgType = 8 AND ExecType = (0 or L)

Tag	FIX tag name		Req'd	Comment
	Standard Header		Y	MsgType = 8
37	OrderID		Y	
11	ClOrdID		Q	Unique identifier set by the client.
453	NoPartyIDs			Optional repeating group used for on behalf of transactions and/or for optional Clearing Firm and Clearing Account.
→	448	PartyID	Q	Party identifier.
→	447	PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452	PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 83 = Clearing Account
17	ExecID		Y	
150	ExecType		Y	Valid values: 0 = New L = Triggered or Activated by the system
39	OrdStatus		Y	Valid values: 0 = New
1	Account			Optional pass-thru field set by client and echoed back by marketplace. From Order
55	Instrument/Symbol		Q	OMNet short name.
48	Instrument/SecurityID		Q	Orderbook ID
22	Instrument/SecurityIDSource		Q	Valid values: M = Marketplace-assigned identifier
54	Side		Y	Valid values: 1 = Buy 2 = Sell
38	OrderQtyData/OrderQty		Q	
40	OrdType		Q	Valid values: 1 = Market 2 = Limit K = Market With Left Over as Limit (market order with unexecuted quantity becoming limit order at last price)

44	Price		
1100	TriggeringInstruction/ TriggerType		Determines what should trigger an order modification. Valid values: 2 = Specified Trading Session 4 = Price Movement
1101	TriggeringInstruction/ TriggerAction		Defines the type of action to take when the trigger hits. Valid values: 1 = Activate
1102	TriggeringInstruction/ TriggerPrice		A specified limit price to validate against price movements –the trigger hits when the price is reached.
1113	TriggeringInstruction/ TriggerTradingSessionID		Identifier of Trading Session when the order is to be triggered. Valid values: 1 = Auction 2 = Continuous Trading
1103	TriggeringInstruction/ TriggerSymbol		Symbol used for price triggers
1104	TriggeringInstruction/ TriggerSecurityID		Identifier of the security used for price triggers.
1105	TriggeringInstruction/ TriggerSecurityIDSource		SecurityIDSource of the instrument used for price triggering. Valid values: M = Marketplace-assigned identifier
1107	TriggeringInstruction/ TriggerPriceType		Determines what price should be tracked for price movements. Valid values:2 = Last Trade
1109	TriggeringInstruction/ TriggerPriceDirection		Used to specify if the trigger should hit only on rising (Up) or falling (Down) prices. Valid values: U = Trigger if the price of the specified type goes UP to or through the specified Trigger Price. D = Trigger if the price of the specified type goes DOWN to or through the specified Trigger Price.
59	TimeInForce	Q	Valid values: 0 = Day 1 = Good Till Cancel (GTC) 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FoK) 6 = Good Till Date (GTD) S = NASDAQ OMX Extension: Good till End of Session (GTS)
336	TradingSessionID		State type of order expiration. Conditionally required if TimeInForce = GTS. Valid values: 1 = Auction 2 = Continuous Trading
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD
18	ExecInst		Valid values: G = All or None (AON)

528	OrderCapacity		Designates the capacity of the firm placing the order. Valid values:
			P = Principal
			A = Agency
			R = Riskless Principal
			NOTE: Required for Fixed Income.

529	OrderRestrictions		Restrictions associated with an order. Valid values:
			B = Issuer Holding **(requires 528=A)**
			C = Issue Price Stabilization **(requires 528=P)**

			5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**
151	LeavesQty	Y	Will be equal to OrderQty on Order.
14	CumQty	Y	Will be 0 on Order Ack.
6	AvgPx	Y	Note: Always set to 0.0
60	TransactTime	Q	
70	AllocID		NASDAQ OMX Extension: Optional pass-thru field set by client and echoed back by marketplace.
797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
1089	MatchIncrement		NASDAQ OMX Extension: Block Size
111	MaxFloor		For hidden orders. Contains currently shown quantity.
20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field.
	Standard Trailer	Y	

6.11.7 Execution Report – IOC/FoK Order Cancel (out)

Purpose: Cancel of IOC or FOK order. Will always be sent last in a sequence following any immediate fills.

Identified by: MsgType = 8 AND ExecType = 4 AND TimeInForce = 3 OR 4

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = 8
37	OrderID	Y	
11	ClOrdID	Q	Unique identifier set by the client.
453	NoPartyIDs		Optional repeating group used for on behalf of transactions and/or for optional Clearing Firm and Clearing Account.
→	448 PartyID	Q	Party identifier.
→	447 PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452 PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 83 = Clearing Account
17	ExecID	Y	
150	ExecType	Y	Valid values: 4 = Canceled
39	OrdStatus	Y	Valid values: 4 = Canceled
1	Account		Optional pass-thru field set by client and echoed back by marketplace. From Order
55	Instrument/Symbol	Q	OMNet short name.
48	Instrument/SecurityID	Q	Orderbook ID
22	Instrument/SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
54	Side	Y	Valid values: 1 = Buy 2 = Sell
38	OrderQtyData/OrderQty	Q	
40	OrdType	Q	Valid values:

			1 = Market 2 = Limit
44	Price		
59	TimeInForce	Q	Valid values: 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FoK)

528	OrderCapacity		Designates the capacity of the firm placing the order. Valid values:
			P = Principal
			A = Agency
			R = Riskless Principal
			NOTE: Required for Fixed Income.

529	OrderRestrictions		Restrictions associated with an order. Valid values:
			B = Issuer Holding **(requires 528=A)**
			C = Issue Price Stabilization **(requires 528=P)**
			5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**

151	LeavesQty	Y	Will be 0.
14	CumQty	Y	
6	AvgPx	Y	Note: Always set to 0.0
60	TransactTime	Q	
70	AllocID		NASDAQ OMX Extension: Optional pass-thru field set by client and echoed back by marketplace.
20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field.
797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
	Standard Trailer	Y	

6.11.8 Execution Report – Market to Limit Order Cancel (out)

Purpose: Sent if a Market to Limit order cannot be immediately executed (nothing on opposite side of the order book).

Identified by: MsgType = 8 AND ExecType = 4 AND OrdType = K

Tag	FIX tag name		Req'd	Comment
	Standard Header		Y	MsgType = 8
37	OrderID		Y	
11	ClOrdID		Q	Unique identifier set by the client.
453	NoPartyIDs			Optional repeating group used for on behalf of transactions and/or for optional Clearing Firm and Clearing Account.
→	448	PartyID	Q	Party identifier.
→	447	PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452	PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 83 = Clearing Account
17	ExecID		Y	
150	ExecType		Y	Valid values: 4 = Canceled
39	OrdStatus		Y	Valid values:

			4 = Canceled
1	Account		Optional pass-thru field set by client and echoed back by marketplace. From Order
55	Instrument/Symbol	Q	OMNet short name.
48	Instrument/SecurityID	Q	Orderbook ID
22	Instrument/SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
54	Side	Y	Valid values: 1 = Buy 2 = Sell
38	OrderQtyData/OrderQty	Q	
40	OrdType	Q	Valid values: K = Market With Left Over as Limit (market order with unexecuted quantity becoming limit order at last price)
59	TimeInForce	Q	Valid values: 0 = Day 1 = Good Till Cancel (GTC) 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FoK) 6 = Good Till Date (GTD) S = NASDAQ OMX Extension: Good till End of Session (GTS)
336	TradingSessionID		State type of order expiration. Conditionally required if TimeInForce = GTS. Valid values: 1 = Auction 2 = Continuous Trading
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD

528	OrderCapacity		Designates the capacity of the firm placing the order. Valid values:
			P = Principal
			A = Agency
			R = Riskless Principal
			NOTE: Required for Fixed Income.

529	OrderRestrictions		Restrictions associated with an order. Valid values:
			B = Issuer Holding **(requires 528=A)**
			C = Issue Price Stabilization **(requires 528=P)**
			5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**

151	LeavesQty	Y	Will be 0.
14	CumQty	Y	Will be 0 in this case.
6	AvgPx	Y	Note: Always set to 0.0
60	TransactTime	Q	
70	AllocID		NASDAQ OMX Extension: Optional pass-thru field set by client and echoed back by marketplace.
20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field.
797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
	Standard Trailer	Y	

6.11.9 Execution Report – Order Reject (out)

Purpose: Order reject.

Identified by: MsgType = 8 AND ExecType = 8

NOTE: This message lacks the required Side (54) field.
NOTE 2: The Symbol field is set to [N/A].

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = 8
37	OrderID	Y	"NONE"
11	ClOrdID	Q	Unique identifier set by the client.
17	ExecID	Y	
150	ExecType	Y	Valid values: 8 = Rejected
39	OrdStatus	Y	Valid values: 8 = Rejected
103	OrdRejReason	Q	Valid values: 0 = Broker / Exchange option
55	Instrument/Symbol	Q	Will be set to [N/A]
151	LeavesQty	Y	Will be 0 on Order Reject.
14	CumQty	Y	Will be 0 on Order Reject.
6	AvgPx	Y	Note: Always set to 0.0
60	TransactTime	Q	
58	Text		Error message
	Standard Trailer	Y	

6.11.10 Execution Report – Cancel Replace Ack (out)

Purpose: Acknowledgement of Order Cancel Replace Request.
Identified by: MsgType = 8 AND ExecType = 5

Tag	FIX tag name		Req'd	Comment
	Standard Header		Y	MsgType = 8
37	OrderID		Y	Genium INET order_number
11	ClOrdID		Q	Unique identifier set by the client.
41	OrigClOrdID			ClOrdID of the order to modify/cancel.
453	NoPartyIDs			Optional repeating group used for on behalf of transactions and/or for optional Clearing Firm and Clearing Account.
→	448	PartyID	Q	Party identifier.
→	447	PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452	PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 83 = Clearing Account
17	ExecID		Y	
150	ExecType		Y	Valid values: 5 = Replaced
39	OrdStatus		Y	Valid values: 0 = New 1 = Partially Filled 2 = Filled

			4 = Canceled
1	Account		Optional pass-thru field set by client and echoed back by marketplace.. From Order
55	Instrument/Symbol	Q	OMNet short name.
48	Instrument/SecurityID	Q	Orderbook ID
22	Instrument/SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
54	Side	Y	Valid values: 1 = Buy 2 = Sell
38	OrderQtyData/OrderQty	Q	
40	OrdType	Q	Valid values: 1 = Market 2 = Limit
44	Price		
1100	TriggeringInstruction/ TriggerType		Determines what should trigger an order modification. Valid values: 2 = Specified Trading Session 4 = Price Movement
1101	TriggeringInstruction/ TriggerAction		Defines the type of action to take when the trigger hits. Valid values: 1 = Activate
1102	TriggeringInstruction/ TriggerPrice		A specified limit price to validate against price movements –the trigger hits when the price is reached.
1113	TriggeringInstruction/ TriggerTradingSessionID		Identifier of Trading Session when the order is to be triggered. Valid values: 1 = Auction 2 = Continuous Trading
1103	TriggeringInstruction/ TriggerSymbol		Symbol used for price triggers
1104	TriggeringInstruction/ TriggerSecurityID		Identifier of the security used for price triggers.
1105	TriggeringInstruction/ TriggerSecurityIDSource		SecurityIDSource of the instrument used for price triggering. Valid values: M = Marketplace-assigned identifier
1107	TriggeringInstruction/ TriggerPriceType		Determines what price should be tracked for price movements. Valid values:2 = Last Trade
1109	TriggeringInstruction/ TriggerPriceDirection		Used to specify if the trigger should hit only on rising (Up) or falling (Down) prices. Valid values: U = Trigger if the price of the specified type goes UP to or through the specified Trigger Price. D = Trigger if the price of the specified type goes DOWN to or through the specified Trigger Price.
59	TimeInForce	Q	Valid values: 0 = Day 1 = Good Till Cancel (GTC) 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FoK) 6 = Good Till Date (GTD) S = NASDAQ OMX Extension: Good till End of Session (GTS)
336	TradingSessionID		State type of order expiration. Conditionally required if TimeInForce = GTS. Valid values: 1 = Auction

			2 = Continuous Trading
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD
18	ExecInst		Valid values: G = All or None (AON)

528	OrderCapacity		Designates the capacity of the firm placing the order. Valid values:
			P = Principal
			A = Agency
			R = Riskless Principal
			NOTE: Required for Fixed Income.

529	OrderRestrictions		Restrictions associated with an order. Valid values:
			B = Issuer Holding **(requires 528=A)**
			C = Issue Price Stabilization **(requires 528=P)**
			5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**

151	LeavesQty	Y	
14	CumQty	Y	
6	AvgPx	Y	Note: Always set to 0.0
60	TransactTime	Q	
70	AllocID		NASDAQ OMX Extension: Optional pass-thru field set by client and echoed back by marketplace.
797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
1089	MatchIncrement		NASDAQ OMX Extension: Block Size
111	MaxFloor		For hidden orders. Contains currently shown quantity.
20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field.
	Standard Trailer	Y	

6.11.11 Execution Report – Cancel Ack (out)

Purpose: Acknowledgement of Order Cancel Request.
Identified by: MsgType = 8 AND ExecType = 4

Tag	FIX tag name		Req'd	Comment
	Standard Header		Y	MsgType = 8
37	OrderID		Y	Genium INET order_number
11	ClOrdID		Q	Unique identifier set by the client.
41	OrigClOrdID			ClOrdID of the order to modify/cancel. Will not be set for orders not entered via FIX.
453	NoPartyIDs			Optional repeating group only used for on behalf of transactions.
→	448	PartyID	Q	Party identifier.
→	447	PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452	PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 12 = Executing Trader
17	ExecID		Y	
150	ExecType		Y	Valid values: 4 = Canceled

39	OrdStatus	Y	Valid values: 4 = Canceled
1	Account		Optional pass-thru field set by client and echoed back by marketplace. From Order
55	Instrument/Symbol	Q	OMNet short name.
48	Instrument/SecurityID	Q	Orderbook ID
22	Instrument/SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
54	Side	Y	Valid values: 1 = Buy 2 = Sell
38	OrderQtyData/OrderQty	Q	

528	OrderCapacity		Designates the capacity of the firm placing the order. Valid values:
			P = Principal
			A = Agency
			R = Riskless Principal
			NOTE: Required for Fixed Income.

529	OrderRestrictions		Restrictions associated with an order. Valid values:
			B = Issuer Holding **(requires 528=A)**
			C = Issue Price Stabilization **(requires 528=P)**
			5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**

151	LeavesQty	Y	Will be 0 on Cancel Ack.
14	CumQty	Y	
6	AvgPx	Y	Always set to 0.0
60	TransactTime	Q	
70	AllocID		NASDAQ OMX Extension: Optional pass-thru field set by client and echoed back by marketplace.
20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field. **NOTE:** Only available for fixed income.
797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
	Standard Trailer	Y	

6.11.12 Execution Report – Unsolicited Cancel (out)

Purpose: Order was cancelled outside of FIX (via other protocol or by the marketplace).
Identified by: MsgType = 8 AND ExecType = 4 AND ExecRestatementReason = 8

Tag	FIX tag name		Req'd	Comment
	Standard Header		Y	MsgType = 8
37	OrderID		Y	Genium INET order_number
11	ClOrdID		Q	Unique identifier set by the client.
453	NoPartyIDs			Optional repeating group used for on behalf of transactions.
→	448	PartyID	Q	Party identifier.
→	447	PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452	PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 12 = Executing Trader
17	ExecID		Y	

150	ExecType	Y	Valid values: 4 = Canceled
39	OrdStatus	Y	Valid values: 4 = Canceled
1	Account		Optional pass-thru field set by client and echoed back by marketplace. From Order
55	Instrument/Symbol	Q	OMNet short name.
48	Instrument/SecurityID	Q	Orderbook ID
22	Instrument/SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
54	Side	Y	Valid values: 1 = Buy 2 = Sell
38	OrderQtyData/OrderQty	Q	

528	OrderCapacity		Designates the capacity of the firm placing the order. Valid values:
			P = Principal
			A = Agency
			R = Riskless Principal
			NOTE: Required for Fixed Income.

529	OrderRestrictions		Restrictions associated with an order. Valid values:
			B = Issuer Holding **(requires 528=A)**
			C = Issue Price Stabilization **(requires 528=P)**
			5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**

151	LeavesQty	Y	Will be 0 for a canceled order
14	CumQty	Y	
6	AvgPx	Y	Always set to 0.0
60	TransactTime	Q	
378	ExecRestatementReason	Q	Valid values: 8 = Market (Exchange) option
70	AllocID		NASDAQ OMX Extension: Optional pass-thru field set by client and echoed back by marketplace.
20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field. **NOTE:** Only available for fixed income.
797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
58	Text		Information on why the order was canceled
	Standard Trailer	Y	

6.11.13 Execution Report – Unsolicited Order Update (out)

Purpose: Order was updated outside of FIX (via other protocol or by the marketplace).
Identified by: MsgType = 8 AND ExecType = D AND ExecRestatementReason = 8

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = 8
37	OrderID	Y	
11	ClOrdID	Q	Unique identifier set by the client.
453	NoPartyIDs		Optional repeating group used for on behalf of transactions and/or for optional Clearing Firm and Clearing Account.
→	448 PartyID	Q	Party identifier.
→	447 PartyIDSource	Q	Valid values: D = Proprietary/Custom code

→	452	PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 83 = Clearing Account
17	ExecID		Y	
150	ExecType		Y	Valid values: D = Restated
39	OrdStatus		Y	Valid values: 0 = New 1 = Partially Filled
1	Account			Optional pass-thru field set by client and echoed back by marketplace. From Order
55	Instrument/Symbol		Q	OMNet short name.
48	Instrument/SecurityID		Q	Orderbook ID
22	Instrument/SecurityIDSource		Q	Valid values: M = Marketplace-assigned identifier
54	Side		Y	Valid values: 1 = Buy 2 = Sell
38	OrderQtyData/OrderQty		Q	Order quantity
40	OrdType		Q	Valid values: 1 = Market 2 = Limit
44	Price			Order price
1100	TriggeringInstruction/ TriggerType			Determines what should trigger an order modification. Valid values: 2 = Specified Trading Session 4 = Price Movement
1101	TriggeringInstruction/ TriggerAction			Defines the type of action to take when the trigger hits. Valid values: 1 = Activate
1102	TriggeringInstruction/ TriggerPrice			A specified limit price to validate against price movements –the trigger hits when the price is reached.
1113	TriggeringInstruction/ TriggerTradingSessionID			Identifier of Trading Session when the order is to be triggered. Valid values: 1 = Auction 2 = Continuous Trading
1103	TriggeringInstruction/ TriggerSymbol			Symbol used for price triggers
1104	TriggeringInstruction/ TriggerSecurityID			Identifier of the security used for price triggers.
1105	TriggeringInstruction/ TriggerSecurityIDSource			SecurityIDSource of the instrument used for price triggering. Valid values: M = Marketplace-assigned identifier
1107	TriggeringInstruction/ TriggerPriceType			Determines what price should be tracked for price movements. Valid values:2 = Last Trade
1109	TriggeringInstruction/ TriggerPriceDirection			Used to specify if the trigger should hit only on rising (Up) or falling (Down) prices. Valid values: U = Trigger if the price of the specified type goes UP to or through the specified Trigger Price. D = Trigger if the price of the specified type goes

			DOWN to or through the specified Trigger Price.
59	TimeInForce	Q	Valid values: 0 = Day 1 = Good Till Cancel (GTC) 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FoK) 6 = Good Till Date (GTD) S = NASDAQ OMX Extension: Good till End of Session (GTS)
336	TradingSessionID		State type of order expiration. Conditionally required if TimeInForce = GTS. Valid values: 1 = Auction 2 = Continuous Trading
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD
18	ExecInst		Valid values: G = All or None (AON)

528	OrderCapacity		Designates the capacity of the firm placing the order. Valid values:
			P = Principal
			A = Agency
			R = Riskless Principal
			NOTE: Required for Fixed Income.

529	OrderRestrictions		Restrictions associated with an order. Valid values:
			B = Issuer Holding **(requires 528=A)**
			C = Issue Price Stabilization **(requires 528=P)**
			5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**

151	LeavesQty	Y	
14	CumQty	Y	
6	AvgPx	Y	Note: Always set to 0.0
60	TransactTime	Q	
378	ExecRestatementReason	Q	Valid values: 8 = Market (Exchange) option
70	AllocID		NASDAQ OMX Extension: Optional pass-thru field set by client and echoed back by marketplace.
797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
1089	MatchIncrement		NASDAQ OMX Extension: Block Size
111	MaxFloor		For hidden orders. Contains currently shown quantity.
20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field. **NOTE:** Only available for fixed income.
	Standard Trailer	Y	

6.11.14 Execution Report – Restatement (out)

Purpose: Restatement of overnight (GTC/GTD) orders in the morning.
Identified by: MsgType = 8 AND ExecType = D AND ExecRestatementReason = 1

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = 8
37	OrderID	Y	Genium INET order_number

11	ClOrdID		Q	Unique identifier set by the client.
453	NoPartyIDs			Optional repeating group used for on behalf of transactions and/or for optional Clearing Firm and Clearing Account.
→	448	PartyID	Q	Party identifier.
→	447	PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452	PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 83 = Clearing Account
17	ExecID		Y	
150	ExecType		Y	Valid values: D = Restated
39	OrdStatus		Y	Valid values: 0 = New 1 = Partially Filled
1	Account			Optional pass-thru field set by client and echoed back by marketplace. From Order
55	Instrument/Symbol		Q	OMNet short name.
48	Instrument/SecurityID		Q	Orderbook ID
22	Instrument/SecurityIDSource		Q	Valid values: M = Marketplace-assigned identifier
54	Side		Y	Valid values: 1 = Buy 2 = Sell
38	OrderQtyData/OrderQty		Q	
40	OrdType		Q	Valid values: 2 = Limit
44	Price			
59	TimeInForce		Q	Valid values: 1 = Good Till Cancel (GTC) 6 = Good Till Date (GTD)
432	ExpireDate			Date of order expiration. Conditionally required if TimeInForce = GTD
18	ExecInst			Valid values: G = All or None (AON)

528	OrderCapacity			Designates the capacity of the firm placing the order. Valid values:
				P = Principal
				A = Agency
				R = Riskless Principal
				NOTE: Required for Fixed Income.

529	OrderRestrictions			Restrictions associated with an order. Valid values:
				B = Issuer Holding **(requires 528=A)**
				C = Issue Price Stabilization **(requires 528=P)**
				5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**

151	LeavesQty		Y	
14	CumQty		Y	
6	AvgPx		Y	Always set to 0.0

60	TransactTime	Q	
378	ExecRestatementReason	Q	Valid values: 1 = GT renewal / restatement (no corporate action)
70	AllocID		NASDAQ OMX Extension: Optional pass-thru field set by client and echoed back by marketplace.
111	MaxFloor		For hidden orders. Contains currently shown quantity.
1089	MatchIncrement		NASDAQ OMX Extension: Block Size
20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field. **NOTE:** Only available for fixed income.
797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
	Standard Trailer	Y	

6.11.15 Execution Report – Fill (out)

Purpose: Order Fill.
NOTE: For Multileg (Combination) order fills, see section 7.5.5.
Identified by: MsgType = 8 AND ExecType = F

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = 8
37	OrderID	Y	
11	ClOrdID	Q	Unique identifier set by the client.
453	NoPartyIDs		Optional repeating group used for on behalf of transactions and/or for optional Clearing Firm and Clearing Account.
→	448 PartyID	Q	Party identifier.
→	447 PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452 PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 83 = Clearing Account
880	TrdMatchID	Q	Match ID assigned by the matching engine.
17	ExecID	Y	Unique identifier of execution message
150	ExecType	Y	Valid values: F = Trade
39	OrdStatus	Y	Valid values: 1 = Partially Filled 2 = Filled
1	Account		Optional pass-thru field set by client and echoed back by marketplace.
55	Instrument/Symbol	Q	OMNet short name.
48	Instrument/SecurityID	Q	Orderbook ID
22	Instrument/SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
54	Side	Y	Valid values: 1 = Buy 2 = Sell
38	OrderQtyData/OrderQty	Q	

40	OrdType	Q	Valid values: 1 = Market 2 = Limit K = Market With Left Over as Limit (market order with unexecuted quantity becoming limit order at last price)
44	Price		
59	TimeInForce	Q	Valid values: 0 = Day 1 = Good Till Cancel (GTC) 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FoK) 6 = Good Till Date (GTD) S = NASDAQ OMX Extension: Good till End of Session (GTS)
336	TradingSessionID		State type of order expiration. Conditionally required if TimeInForce = GTS. Valid values: 1 = Auction 2 = Continuous Trading
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD
18	ExecInst		Valid values: G = All or None (AON)

528	OrderCapacity		Designates the capacity of the firm placing the order. Valid values:
			P = Principal
			A = Agency
			R = Riskless Principal
			NOTE: Required for Fixed Income.

529	OrderRestrictions		Restrictions associated with an order. Valid values:
			B = Issuer Holding **(requires 528=A)**
			C = Issue Price Stabilization **(requires 528=P)**
			5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**

32	LastQty	Q	Quantity (e.g. shares) bought/sold on this (last) fill.
31	LastPx	Q	Price of this (last) fill.
151	LeavesQty	Y	Quantity open for further execution.
14	CumQty	Y	Currently executed quantity for chain of orders. **NOTE:** Will be 0 for fills on quotes.
6	AvgPx	Y	Note: Always set to 0.0
60	TransactTime	Q	
70	AllocID		NASDAQ OMX Extension: Optional pass-thru field set by client and echoed back by marketplace.
797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
111	MaxFloor		For hidden orders. Contains currently shown quantity.
1089	MatchIncrement		NASDAQ OMX Extension: Block Size
20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field. **NOTE:** Only available for fixed income.
	Standard Trailer	Y	

6.11.16 Execution Report – Order Suspended (out)

Purpose: Order Suspended (likely caused by temporary loss of connectivity).
Identified by: MsgType = 8 AND ExecType = 9

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = 8
37	OrderID	Y	
11	ClOrdID	Q	Unique identifier set by the client.
453	NoPartyIDs		Optional repeating group used for on behalf of transactions and/or for optional Clearing Firm and Clearing Account.
→	448 PartyID	Q	Party identifier.
→	447 PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452 PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 83 = Clearing Account
17	ExecID	Y	
150	ExecType	Y	Valid values: 9 = Suspended
39	OrdStatus	Y	Valid values: 9 = Suspended
1	Account		Account or client information passed on to downstream clearing system. From Order
55	Instrument/Symbol	Q	Short name of security
48	Instrument/SecurityID	Q	Orderbook ID
22	Instrument/SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
54	Side	Y	Valid values: 1 = Buy 2 = Sell
38	OrderQtyData/OrderQty	Q	
40	OrdType	Q	Valid values: 1 = Market 2 = Limit
44	Price		
1100	TriggeringInstruction/ TriggerType		Determines what should trigger an order modification. Valid values: 2 = Specified Trading Session 4 = Price Movement
1101	TriggeringInstruction/ TriggerAction		Defines the type of action to take when the trigger hits. Valid values: 1 = Activate
1102	TriggeringInstruction/ TriggerPrice		A specified limit price to validate against price movements –the trigger hits when the price is reached.
1113	TriggeringInstruction/ TriggerTradingSessionID		Identifier of Trading Session when the order is to be triggered. Valid values: 1 = Auction 2 = Continuous Trading
1103	TriggeringInstruction/ TriggerSymbol		Symbol used for price triggers
1104	TriggeringInstruction/ TriggerSecurityID		Identifier of the security used for price triggers.
1105	TriggeringInstruction/		SecurityIDSource of the instrument used for price

	TriggerSecurityIDSource		triggering. Valid values: M = Marketplace-assigned identifier
1107	TriggeringInstruction/ TriggerPriceType		Determines what price should be tracked for price movements. Valid values:2 = Last Trade
1109	TriggeringInstruction/ TriggerPriceDirection		Used to specify if the trigger should hit only on rising (Up) or falling (Down) prices. Valid values: U = Trigger if the price of the specified type goes UP to or through the specified Trigger Price. D = Trigger if the price of the specified type goes DOWN to or through the specified Trigger Price.
59	TimeInForce	Q	Valid values: 0 = Day 1 = Good Till Cancel (GTC) 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FoK) 6 = Good Till Date (GTD) S = NASDAQ OMX Extension: Good till End of Session (GTS)
336	TradingSessionID		State type of order expiration. Conditionally required if TimeInForce = GTS. Valid values: 1 = Auction 2 = Continuous Trading
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD
18	ExecInst		Valid values: G = All or None (AON)

528	OrderCapacity		Designates the capacity of the firm placing the order. Valid values:
			P = Principal
			A = Agency
			R = Riskless Principal
			NOTE: Required for Fixed Income.

529	OrderRestrictions		Restrictions associated with an order. Valid values:
			B = Issuer Holding **(requires 528=A)**
			C = Issue Price Stabilization **(requires 528=P)**
			5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**

151	LeavesQty	Y	Quantity open for further execution.
14	CumQty	Y	Currently executed quantity for chain of orders.
6	AvgPx	Y	Note: Always set to 0.0
60	TransactTime	Q	
70	AllocID		NASDAQ OMX Extension: Optional pass-thru field set by client and echoed back by marketplace.
797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
111	MaxFloor		For hidden orders. Contains currently shown quantity.
1089	MatchIncrement		NASDAQ OMX Extension: Block Size
20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field. **NOTE:** Only available for fixed income.
	Standard Trailer	Y	

6.11.17 Execution Report – Expired (out)

Purpose: GTD, GTS **or GTC** Order Expired. For GTD orders the Expired transaction will be sent the day after the order expired. For GTS orders the Expired message is sent at order expiry. Order expiry can occur for GTC orders under certain conditions. See Order Expiry section for details.

Identified by: MsgType = 8 AND ExecType = C

Tag	FIX tag name		Req'd	Comment
	Standard Header		Y	MsgType = 8
37	OrderID		Y	Genium INET order_number
11	ClOrdID		Q	Unique identifier set by the client.
453	NoPartyIDs			Optional repeating group used for on behalf of transactions and/or for optional Clearing Firm and Clearing Account.
→	448	PartyID	Q	Party identifier.
→	447	PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452	PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 83 = Clearing Account
17	ExecID		Y	Identifier for this execution report. Integer value.
150	ExecType		Y	Valid values: C = Expired
39	OrdStatus		Y	Valid values: C = Expired
1	Account			Account or client information passed on to downstream clearing system. From Order
55	Instrument/Symbol		Q	Short name of security
48	Instrument/SecurityID		Q	Orderbook ID
22	Instrument/SecurityIDSource		Q	Valid values: M = Marketplace-assigned identifier
54	Side		Y	Valid values: 1 = Buy 2 = Sell
38	OrderQtyData/OrderQty			Order quantity
40	OrdType		Q	Valid values: 2 = Limit
44	Price			Order price
59	TimeInForce		Q	Valid values: 1 = Good Till Cancel (GTC) 6 = Good Till Date (GTD) S = NASDAQ OMX Extension: Good till End of Session (GTS)
336	TradingSessionID			State type of order expiration. Conditionally required if TimeInForce = GTS. Valid values: 1 = Auction 2 = Continuous Trading
18	ExecInst			Valid values: G = All or None (AON)
528	OrderCapacity			Designates the capacity of the firm placing the order. Valid values: P = Principal

			A = Agency
			R = Riskless Principal
			NOTE: Required for Fixed Income.

529	OrderRestrictions		Restrictions associated with an order. Valid values:
			B = Issuer Holding **(requires 528=A)**
			C = Issue Price Stabilization **(requires 528=P)**
			5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**

151	LeavesQty	Y	Will be 0 on expired orders.
14	CumQty	Y	
6	AvgPx	Y	Always set to 0.0
60	TransactTime	Q	
70	AllocID		NASDAQ OMX Extension: Optional pass-thru field set by client and echoed back by marketplace.
797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
111	MaxFloor		For hidden orders. Contains currently shown quantity.
1089	MatchIncrement		NASDAQ OMX Extension: Block Size
20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field. **NOTE:** Only available for fixed income.
	Standard Trailer	Y	

6.11.18 Business Message Reject (out)

Purpose: Business message reject.
Identified by: MsgType = j

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = j
45	RefSeqNum		MsgSeqNum of rejected message
372	RefMsgType	Y	The MsgType of the FIX message being referenced.
380	BusinessRejectReason	Y	Valid values: 0 = Other 1 = Unknown ID 2 = Unknown Security 3 = Unsupported Message Type 4 = Application not available 5 = Conditionally required field missing
58	Text		Free format text describing the error
	Standard Trailer	Y	

7 Multileg Orders

7.1 Overview

A multileg security is made up of multiple securities that are traded atomically. Swaps, option strategies, futures spreads, are a few examples of multileg securities. The requirement that all legs be traded in the quantities that make up the multileg security is the important distinction between a multileg order and a list order.

Two generalized approaches to trading multileg securities are supported by FIX. The first approach involves a market maintaining multileg securities as separate products for which markets can be created. This “product approach” is often used in electronic trading systems. The second approach is to trade the multileg security as a group of separate securities.

The multileg order can be traded using one of the following FIX trading models. The first two models are variations on the multileg security as a separate tradable product. The last models permits trading of multileg securities in environments where the multileg securities are not productized.

Pre-defined Multileg Security Model
A.k.a. *Standard Combinations*. Marketplace-defined multileg securities made available for trading. In Genium INET, Standard Combination orders are treated exactly as single orders. To mimic this behavior, the FIX representation of entering a Standard Combination order is a normal New Order Single.

User-defined Multileg Security Model
A.k.a. *Tailor-Made Combinations (TMC)*. User-defined multileg securities made available for trading.

Strategy orders
A.k.a. *Non-Standard Combinations*. Multileg orders for combinations of security where a product is not defined or made available for others to trade.
NOTE: *Strategy Orders are not supported in this solution.*

7.2 Multileg Order Features

Multileg orders are traded just like ordinary single orders, i.e. they;

- Have the same types of trading instructions, although the set of possibilities is limited.
- Use the same response messages, e.g. Execution Reports
- Are canceled using the Order Cancel Request or message
- Share the same type of workflows as New Order Single and Order Cancel Replace Request

Please see chapter 6, Order Management for information on aspects shared with single order messages.

7.2.1 Creating a Tailor-Made Combination Instrument

When trading a TMC the properties of each the legs are important. Each leg has the following properties:

- The instrument of the leg. This is represented by the LegSymbol (600) or LegSecurityID (602) fields.
- The Ratio Quantity of the leg. The relative number of contracts between the TMC legs. The FIX field to be used is LegRatioQuantity (623).
- The Side of each leg. The Side for each leg is relative to the TMC itself. The

The Security Definition Request is used to request creation of a TMC.

7.2.2 Multileg Order Limitations

Multileg orders have some limitations compared to regular orders. Most regular order features are available for multileg orders as well. The exceptions are:

- Overnight orders (TimeInForce= GTC or GTD) are not allowed.
- Reserve (Hidden) orders are not allowed.
- Triggers are not allowed.

7.3 Main Workflow

7.3.1 Submitting a Tailor-Made Combination Instrument Definition

A TMC is created by submitting a Security Definition Request to the marketplace. The system will respond with a Security Definition – TMC registration response (see section 7.5.2 for message details).

7.3.1.1 TMC Registration Response

The response to a submitted TMC registration request is a Security Definition message. This message will only contain the omnet series struct in integer format. *The actual instrument definition is only provided via reference data sessions, not via the FIX session where the registration was made.* The values in the FIX response can be used to identify the instrument definition in reference data. Using the instrument data received over omnet, the user can extract the omnet short name used as Symbol (55) in FIX to trade the instrument.

The SecurityResponseType (tag 323) will indicate whether the request was successful or not.

7.3.2 New Order

The multileg order workflow starts with user submitting an order.
In this solution, multileg orders are sent as ordinary New Order Single messages both for Standard Combination and Tailor-Made Combination Orders.
In response one Execution Report is produced for the multileg itself. The response will contain the OrderID that will be present in all later Execution Reports.

7.3.3 Order Modification

Order modification is accomplished using the Order Cancel Replace message. The message is used to modify an existing order and does not support delta updates (all relevant fields must be supplied). In response one Execution Report is produced for the multileg itself.

7.3.4 Multileg Status Reporting

Entering, cancelling or modifying an existing multileg order works exactly like any other instrument. Acknowledgements and rejects (Execution Report or Order Cancel Rejects) also look exactly like those for “ordinary” orders. See chapter 6 for details.
The only difference is with fills, which are sent per leg (see section 7.3.5).

NOTE:
A multileg order has a single OrderID (37) and ClOrdID (11), just like other orders. The legs are not considered to be orders in their own right.

7.3.5 Fills

When multileg orders are filled, Execution Reports are issued. The Execution Report – Combination Order Fill is used for multileg fills. See section 7.5.5 for message details.

Different models can be used in FIX to represent a fill. The model used in this solution is:

- **Multi-Leg only.** In this model a single Execution Report –Combination Order Fill is sent for the combination as a whole. The repeating group starting with the NoLegs (555) field (InstrmntLegExecGrp) will contain one entry per match that occurred in each leg. Each entry contains price and quantity.
  NOTE: It is entirely possible to receive more entries than the number of legs. There may have been more than one trade in each leg in a single matching round.

7.4 Workflows

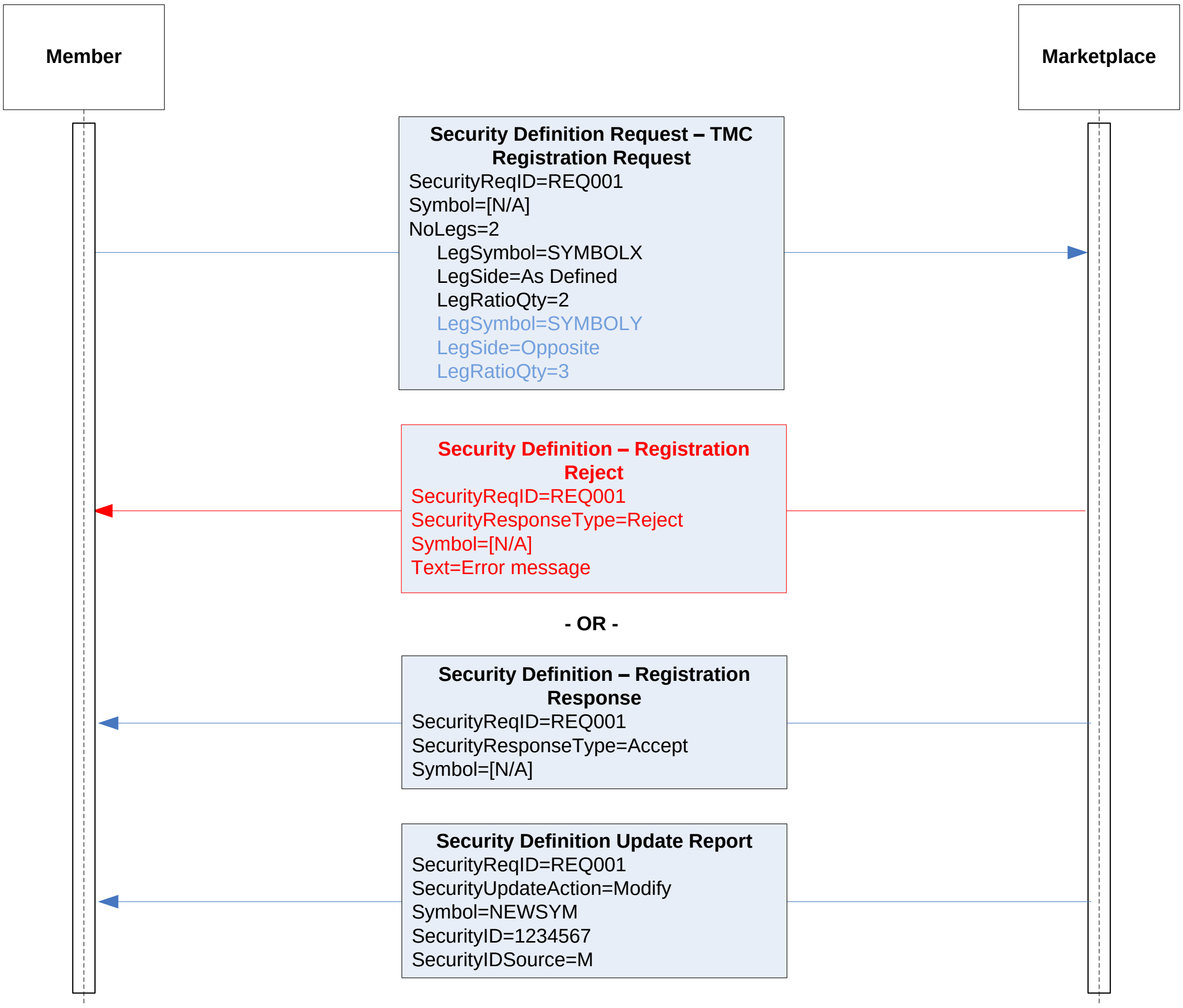
7.4.1 Registering a new TMC instrument

This example shows how to create a new Tailor-Made Combination with two legs. The two legs of requested TMC have the following properties:
Leg A (SYMBOLX):

- When a quantity of one (1) of the TMC is bought, a quantity of *2* (LegRatioQty=2) is *bought* (LegSide=As Defined).

Leg B (SYMBOLY):

- When a quantity of one (1) of the TMC is bought, a quantity of *3* (LegRatioQty=3) is *sold* (LegSide=Opposite).



Security Definition Request – TMC Registration Request
SecurityReqID=REQ001
Symbol=[N/A]
NoLegs=2
LegSymbol=SYMBOLX
LegSide=As Defined
LegRatioQty=2
LegSymbol=SYMBOLY
LegSide=Opposite
LegRatioQty=3

Security Definition – Registration Reject
SecurityReqID=REQ001
SecurityResponseType=Reject
Symbol=[N/A]
Text=Error message

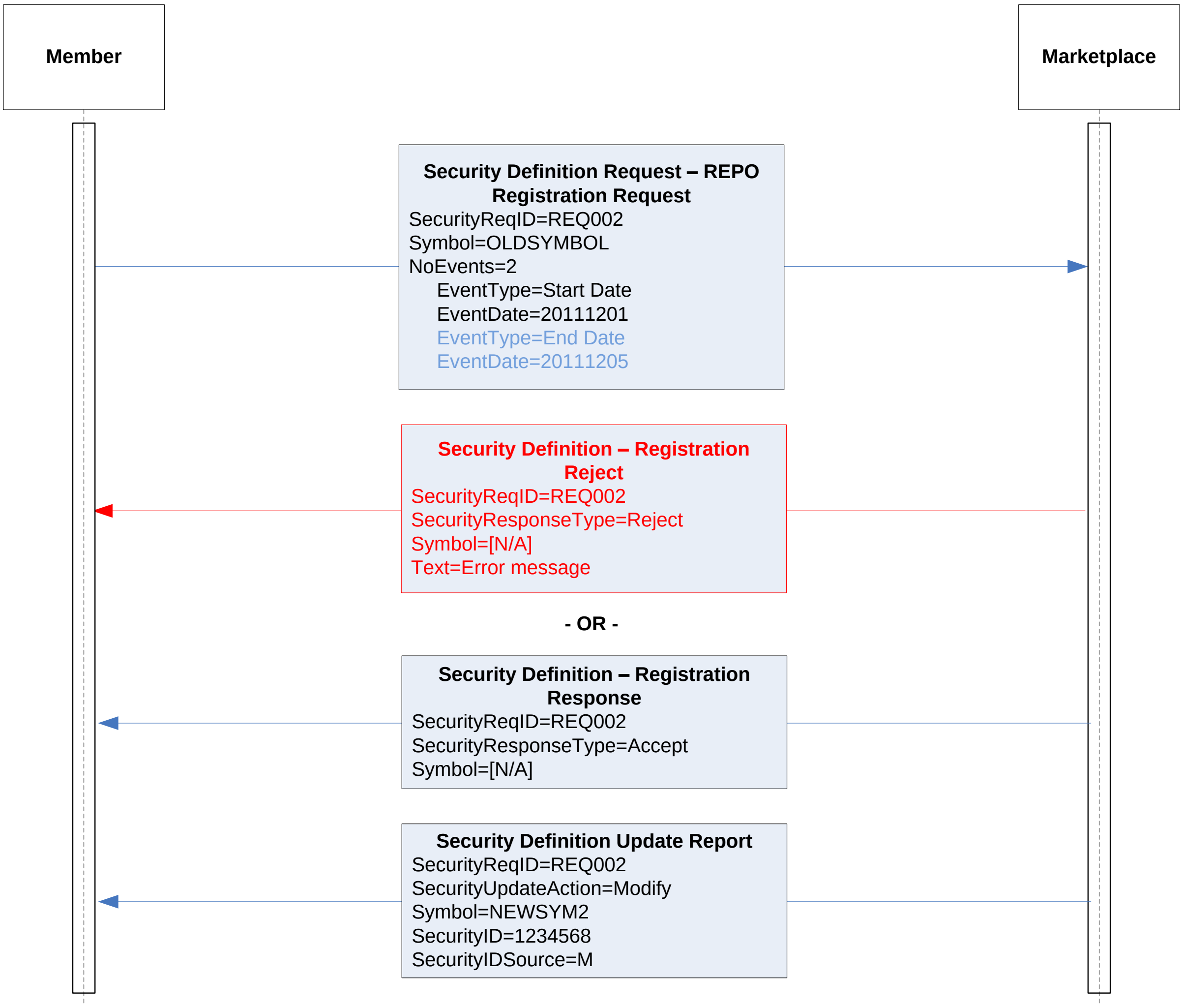
- OR -

Security Definition – Registration Response
SecurityReqID=REQ001
SecurityResponseType=Accept
Symbol=[N/A]

Security Definition Update Report
SecurityReqID=REQ001
SecurityUpdateAction=Modify
Symbol=NEWSYM
SecurityID=1234567
SecurityIDSource=M

7.4.2 Registering a new REPO instrument

This example shows how to register a new REPO instrument. The requested instrument has the same properties as OLDSYMBOL referred to in the request, but with new start and end dates.



Security Definition Request – REPO Registration Request
SecurityReqID=REQ002
Symbol=OLDSYMBOL
NoEvents=2
EventType=Start Date
EventDate=20111201
EventType=End Date
EventDate=20111205

Security Definition – Registration Reject
SecurityReqID=REQ002
SecurityResponseType=Reject
Symbol=[N/A]
Text=Error message

Security Definition – Registration Response
SecurityReqID=REQ002
SecurityResponseType=Accept
Symbol=[N/A]

Security Definition Update Report
SecurityReqID=REQ002
SecurityUpdateAction=Modify
Symbol=NEWSYM2
SecurityID=1234568
SecurityIDSource=M

7.5 Message Details

7.5.1 Security Definition Request –TMC Registration Request (in)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = c
320	SecurityReqID	Y	Client-generated identifier.
321	SecurityRequestType	Y	Type of Security Definition request. Valid values: 1 = Request Security identity for the specifications provided (name of the security is not supplied)
55	Instrument/Symbol	Y	Should be set to [N/A]
555	NoLegs	Q	Number of legs
→	600 LegSymbol	Q	OMNet short name for this leg. **NOTE:** if LegSecurityID+LegSecurityIDSource are used instead of LegSymbol, LegSymbol must be set to [N/A].

→	602	LegSecurityID		Orderbook ID for this leg.
→	603	LegSecurityIDSource		Valid values: M = Marketplace-assigned identifier

→	624	LegSide	Q	The side of this individual leg (multileg security).
				Valid values:
				B = As Defined
				C = Opposite

→	623	LegRatioQty	Q	The ratio of quantity for this individual leg relative to the entire multileg security.
	Standard Trailer		Y	

7.5.2 Security Definition Request –Repo Registration Request (in)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = c
320	SecurityReqID	Y	Client-generated identifier.
321	SecurityRequestType	Y	Type of Security Definition request. Valid values: 1 = Request Security identity for the specifications provided (name of the security is not supplied)
55	Symbol		Symbol or SecurityID+SecurityIDSource must be set to an existing repo instrument.
48	SecurityID		
22	SecurityIDSource		
864	NoEvents	Q	
→	865 EventType	Q	Valid values: 101 = Start Date (NASDAQ OMX Extension) 102 = End Date (NASDAQ OMX Extension)
→	866 EventDate	Q	Date of Event
	Standard Trailer	Y	

7.5.3 Security Definition – Registration Response (out)

Purpose: Accept of a TMC or REPO registration request.
Identified by: MsgType = d AND SecurityResponseType = 1

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = d
320	SecurityReqID	Y	Client-generated identifier.
323	SecurityResponseType	Q	Type of Security Definition message response. Valid values: 1 = Accept security proposal as-is
55	Instrument/Symbol	Q	Symbol not generated when this message is generated. Will be set to [N/A].
48	Instrument/SecurityID	Q	Contains the Omnet series struct in integer form with the fields separated by colons “:”: country_c : market_c : instrument_group_c :

			modifier_c : commodity_n : expiration_date_n : strike_price_i
22	Instrument/SecurityIDSource	Q	101 = Genium INET series definition (NASDAQ OMX Extension)
	Standard Trailer	Y	

7.5.4 Security Definition –Registration Reject (out)

Purpose: Reject of a TMC or REPO registration request.
Identified by: MsgType = d AND SecurityResponseType = 5

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = d
320	SecurityReqID	Y	Client-generated identifier.
323	SecurityResponseType	Q	Type of Security Definition message response. Valid values: 5 = Reject security proposal
58	Text	Q	Error message
	Standard Trailer	Y	

7.5.5 Security Definition Update Report (out)

Purpose: Return instrument identifiers usable for FIX.
Identified by: MsgType = BP

Tag	FIX tag name		Req'd	Comment
	Standard Header		Y	MsgType = BP
320	SecurityReqID		Y	Client-generated identifier.
980	SecurityUpdateAction		Q	Valid values: M = Modify
55	Symbol		Q	Symbol of created instrument.
48	SecurityID		Q	Order book ID of created instrument.
22	SecurityIDSource		Q	Valid values: M = Marketplace-assigned identifier
454	NoSecurityAltID		Q	Number of alternate SecurityIDs. Will always be 1.
→	455	SecurityAltID	Q	Contains the Omnet series struct in integer form with the fields separated by colons ":": country_c : market_c : instrument_group_c : modifier_c : commodity_n : expiration_date_n : strike_price_i
→	456	SecurityAltIDSource	Q	101 = Genium INET series definition (NASDAQ OMX Extension)
555	NoLegs			Number of legs (for strategy/combination) instruments.
→	600	LegSymbol		Short name of leg instrument.
→	602	LegSecurityID		Order book ID of leg instrument.
→	603	LegSecurityIDSource		Valid values: M = Marketplace-assigned identifier
→	623	LegRatioQty		The ratio of quantity for this individual leg relative to the entire multileg security.
→	624	LegSide		The side of this individual leg (multileg security). Valid values:

				B = As Defined
				C = Opposite

	Standard Trailer		Y	

7.5.6 Execution Report – Combination Order Fill (out)

Purpose: Combination Order Fill.
Identified by: MsgType = 8 AND ExecType = F AND MultiLegReportingType = 3

Tag	FIX tag name		Req'd	Comment
	Standard Header		Y	MsgType = 8
37	OrderID		Y	
11	ClOrdID		Q	
453	NoPartyIDs			Optional repeating group used for on behalf of transactions and/or for optional Clearing Firm and Clearing Account.
→	448	PartyID	Q	Party identifier.
→	447	PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452	PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 83 = Clearing Account
880	TrdMatchID		Q	Match ID assigned by the matching engine.
17	ExecID		Y	
150	ExecType		Y	Valid values: F = Trade
39	OrdStatus		Y	Valid values: 1 = Partially filled 2 = Filled
1	Account			Optional pass-thru field set by client and echoed back by marketplace.
55	Instrument/Symbol		Q	Combination orderbook OMNet short name.
48	Instrument/SecurityID		Q	Combination Orderbook ID
22	Instrument/SecurityIDSource		Q	Valid values: M = Marketplace-assigned identifier
54	Side		Y	Valid values: 1 = Buy 2 = Sell
40	OrdType		Q	Valid values: 1 = Market 2 = Limit
44	Price		Q	Net price of the combination as entered in the order.
59	TimeInForce		Q	Valid values: 0 = Day 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FoK) S = NASDAQ OMX Extension: Good till End of Session (GTS)
336	TradingSessionID			State type of order expiration. Conditionally

				required if TimeInForce = GTS. Valid values: 1 = Auction 2 = Continuous Trading
32	LastQty			Quantity (e.g. shares) bought/sold on this (last) fill.
31	LastPx			Net price of this (last) multileg fill.
151	LeavesQty		Y	
14	CumQty		Y	
6	AvgPx		Y	Note: Always set to 0.0
60	TransactTime		Y	
442	MultiLegReportingType		Q	Valid values: 3 = Multi-leg security
555	NoLegs		Q	Number of legs involved in execution
→	600	LegSymbol	Q	Omnet short name of leg security
→	602	LegSecurityID	Q	Orderbook ID of leg security
→	603	LegSecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
→	637	LegLastPx	Q	Trade price for this leg
→	1418	LegLastQty	Q	NASDAQ OMX Extension: Quantity traded in this leg
70	AllocID			NASDAQ OMX Extension: Optional pass-thru field set by client and echoed back by marketplace.
797	CopyMsgIndicator			Set to ‘Y’ on Drop Copy messages
	Standard Trailer			

8 Contingent (Linked) Orders

Contingent orders (a.k.a. Linked Orders or Alternative Orders) are orders that have a dependency on other orders. The orders of a contingency are entered together in a single message. A Contingent Order can be regarded as a multileg order where a fill in one leg affects the other legs. It can also be described as a multileg order with an OR condition between the legs, instead of an AND condition. In the Contingent Order case, the multileg itself is generally not a product but individual securities. As the legs of a Contingent order is in fact separate orders, they should also be treated as separate orders from a messaging flow (Execution Report, etc) point of view.

NOTE: The List Order messages of the FIX Standard are also used for the trading of baskets, programs and similar – that functionality is currently not supported!

The Contingent Order (or rather the individual orders of it) is allowed to sit on the book; it is made public by displaying each individual order as a separate order over market data. There will be no resulting trade for the Contingent Order as such; all trades are for the individual security.

There are various kinds of contingent orders, but this solution only supports the **One Updates the Other model (OUO)**.

8.1 One Updates the Other (OUO)

An OUO order is an order whose execution results in the immediate reduction of quantity in another order linked to it. The quantity reduction happens on a best effort basis. In an OUO order, the linked orders are live in the marketplace at the same time. The execution of either order triggers an attempt to reduce the remaining quantity of the other order(s), partial executions included. The other orders are reduced in proportion to the filled quantity.

Example: Order A is for 100; Order B is for 50; Order C is for 80.

- When order B is partially filled for 25 (50 %), order A is restated to a leaves quantity of 50 (50 %) and order C is restated to a leaves quantity of 40 (50%).

8.2 Main Workflow

A set of contingent orders are entered using the New Order List message. As the contingency is accepted or rejected, a List Status message is returned including the reason for a reject if applicable. The orders making up the contingency are validated together. If one leg is invalid, the whole New Order List will be rejected.
State changes for the individual contingent orders are relayed using the Execution Report message. All other actions follow the ordinary order messaging (see chapter 5), but note that updating the individual contingent orders is subject to restrictions not applicable for non-contingent orders.
NOTE: Contingent orders may be subject to limitations regarding what order conditions apply. A discussion of these rules is outside the scope of this specification.

8.2.1 Cancel a List

If the user wishes to cancel the entire contingency, a List Cancel Request specifying the relevant ListID must be sent. The client will receive a List Status message as an acknowledgement/reject. If the cancel was accepted, the client will also receive individual cancel messages (Execution Report – Unsolicited Order Cancel) per order in the contingency.

8.2.1.1 Cancel an order within a list

A specific order belonging to the list can be canceled using a regular Order Cancel Request message. Note that if one order (leg) is cancelled all other legs will also be cancelled (Execution Report – Unsolicited Cancel).

8.2.2 Order Updates

When a fill occurs to one of the orders in the contingency, the linked orders will also be affected. Following an Execution Report – Fill, one Execution Report – Unsolicited Order Update will be sent for each linked order, reducing the quantity (OrderQty) proportionally to the fill.

8.3 Order Identifiers

Individual Orders of the contingency are identified using ordinary ClOrdID (11) and OrderID (37) fields.
The contingent order itself has a ListID (66) to identify it. This ListID is present on all Execution Reports for the orders within the contingency.

8.4 Common Properties

The following fields are set per leg, but the values of each are required to be the same across all legs:

- TimeInForce (59)
- OrderCapacity (528)
- OrderRestrictions (529)
- OrderReference (20009)

8.5 Restrictions

There is no update message for a list of Contingent Orders, if the user wishes to update the entire list he must cancel the whole Contingency and submit a new one.

An order that is part of a Contingent Order cannot be removed from the contingency. If an Order Cancel Request is sent against an individual order in the Contingent Order list, all of the orders are cancelled. To remove a single order from the contingency, the entire contingent order must be cancelled and reentered without the order that should be removed.

An order cannot be added to the contingency – there is no ListID (66) in the New Order Single message. To add a new order to the contingency, the original contingency order must be cancelled and a new contingent order with the additional order must be submitted to the marketplace.

Contingent orders are implicitly good for continuous trading sessions only. When the orderbook of one of the legs shift away from continuous matching, that leg is cancelled.

The number of orders allowed for each type of contingency is bilaterally agreed.

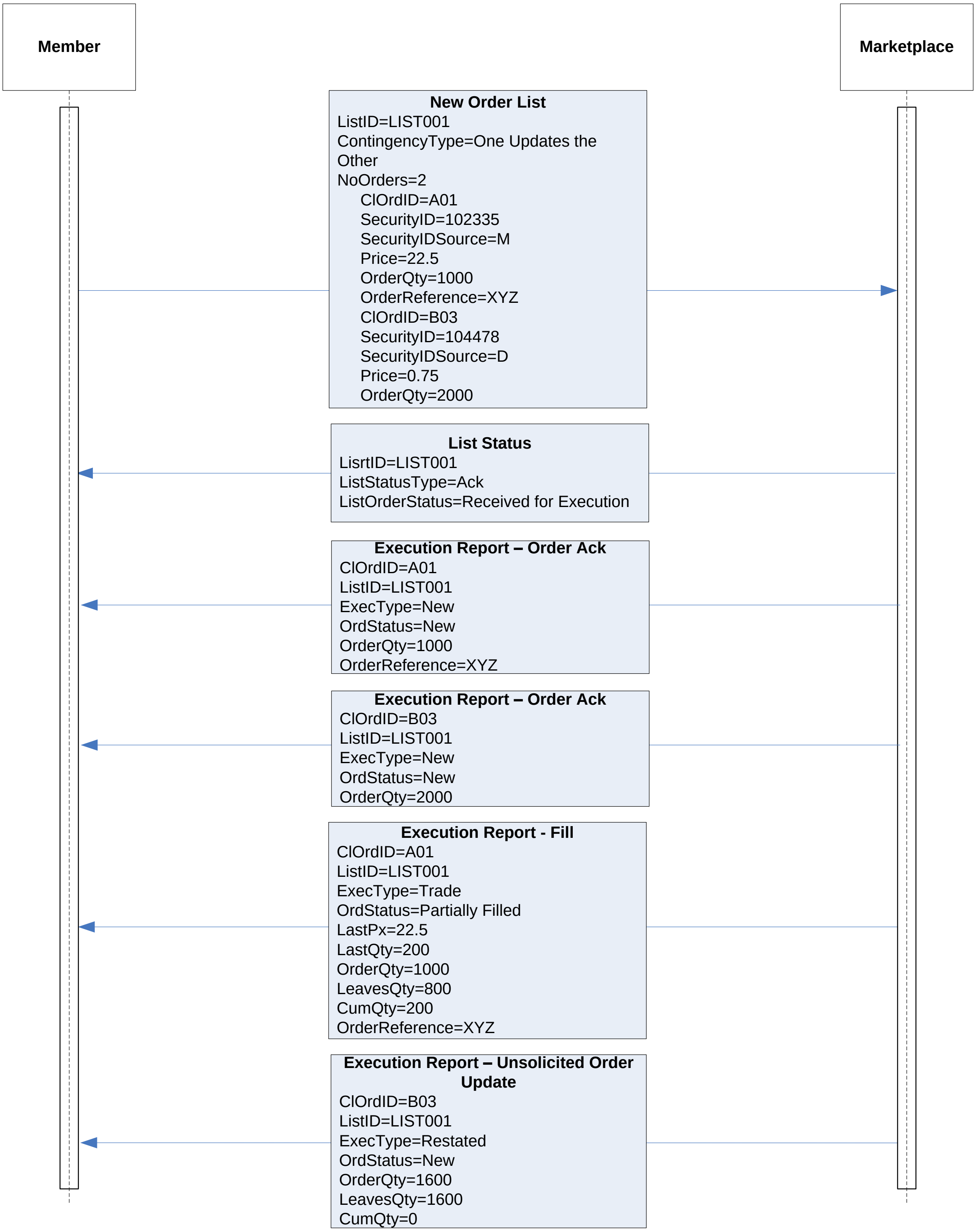
A Reserve size (hidden order) is not allowed.

The number of lot sizes, i.e. Leg qty / Leg Lot Size, must be the same for all legs. If not, the linked order as a whole is rejected.

8.6 **Workflows**

8.6.1 Entering a Linked Order, followed by a partial fill in one of the orders

A New Order List containing two orders is entered. After reception of List Ack and individual order acks, one of the orders is partially filled. The other linked order is reduced in quantity in proportion to the fill in the other order.



New Order List
ListID=LIST001
ContingencyType=One Updates the Other
NoOrders=2
ClOrdID=A01
SecurityID=102335
SecurityIDSource=M
Price=22.5
OrderQty=1000
OrderReference=XYZ
ClOrdID=B03
SecurityID=104478
SecurityIDSource=D
Price=0.75
OrderQty=2000

List Status
LisrtID=LIST001
ListStatusType=Ack
ListOrderStatus=Received for Execution

Execution Report – Order Ack
ClOrdID=A01
ListID=LIST001
ExecType=New
OrdStatus=New
OrderQty=1000
OrderReference=XYZ

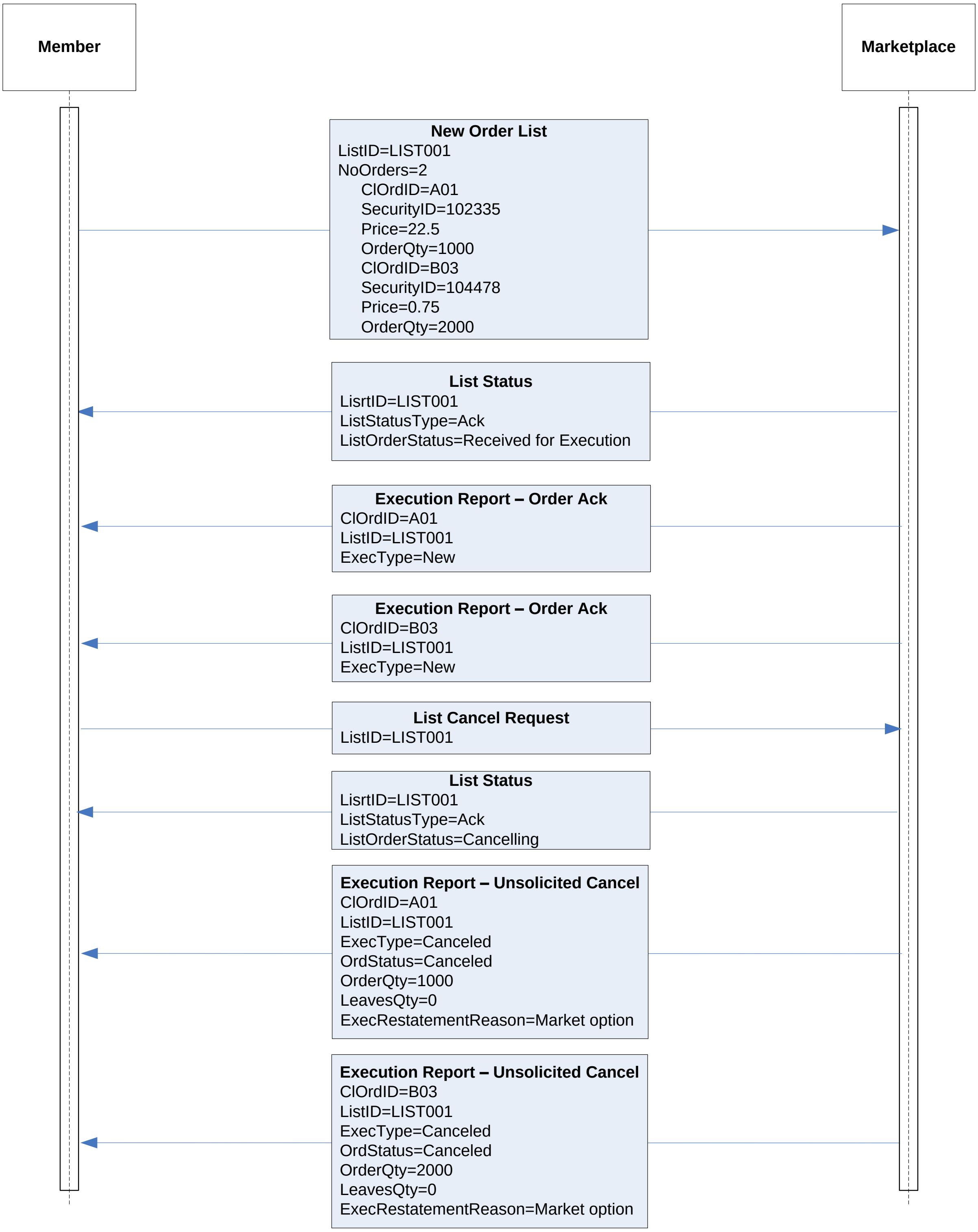
Execution Report – Order Ack
ClOrdID=B03
ListID=LIST001
ExecType=New
OrdStatus=New
OrderQty=2000

Execution Report - Fill
ClOrdID=A01
ListID=LIST001
ExecType=Trade
OrdStatus=Partially Filled
LastPx=22.5
LastQty=200
OrderQty=1000
LeavesQty=800
CumQty=200
OrderReference=XYZ

Execution Report – Unsolicited Order Update
ClOrdID=B03
ListID=LIST001
ExecType=Restated
OrdStatus=New
OrderQty=1600
LeavesQty=1600
CumQty=0

8.6.2 New Order List followed by List Cancel Request

In this example a New Order List containing two linked orders is sent in. After accept, the list is cancelled using the List Cancel Request.



New Order List
ListID=LIST001
NoOrders=2
    ClOrdID=A01
    SecurityID=102335
    Price=22.5
    OrderQty=1000
    ClOrdID=B03
    SecurityID=104478
    Price=0.75
    OrderQty=2000

List Status
LisrtID=LIST001
ListStatusType=Ack
ListOrderStatus=Received for Execution

Execution Report – Order Ack
ClOrdID=A01
ListID=LIST001
ExecType=New

Execution Report – Order Ack
ClOrdID=B03
ListID=LIST001
ExecType=New

List Cancel Request
ListID=LIST001

List Status
LisrtID=LIST001
ListStatusType=Ack
ListOrderStatus=Cancelling

Execution Report – Unsolicited Cancel
ClOrdID=A01
ListID=LIST001
ExecType=Canceled
OrdStatus=Canceled
OrderQty=1000
LeavesQty=0
ExecRestatementReason=Market option

Execution Report – Unsolicited Cancel
ClOrdID=B03
ListID=LIST001
ExecType=Canceled
OrdStatus=Canceled
OrderQty=2000
LeavesQty=0
ExecRestatementReason=Market option

8.7 **Message Details**

8.7.1 New Order List (in)

Tag	FIX tag name		Req'd	Comment
	Standard Header		Y	MsgType = E
66	ListID		Y	Unique identifier for list as assigned by sender
394	BidType		Y	Code to identify the type of Bid Request. Valid values: 3 = No bidding process
1385	ContingencyType		Q	NASDAQ OMX Extension: Defines the type of contingency. Valid values: 4 = One Updates the Other (OUO) – proportional Quantity Reduction
68	TotNoOrders		Y	FIX required field used to support fragmentation, which is not supported in this solution. Value is ignored.
1116	NoRootPartyIDs			Optional repeating group used for on behalf of transactions and/or for optional Clearing Firm and Clearing Account.
→	1117	RootPartyID	Q	Party identifier.
→	1118	RootPartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	1119	RootPartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 83 = Clearing Account
73	NoOrders		Y	Number of orders in this message.
→	11	ClOrdID	Y	Client Order ID
→	67	ListSeqNo	Y	Required in FIX, but ignored
→	1089	MatchIncrement		NASDAQ OMX Extension
→	55	Symbol		OMNet short name. Symbol or SecurityID+SecurityIDSource must be set.
→	48	SecurityID		Orderbook ID
→	22	SecurityIDSource		Valid values: M = Marketplace-assigned identifier
→	54	Side	Y	Valid values: 1 = Buy 2 = Sell
→	38	OrderQty	Y	List order quantity
→	40	OrdType	Q	Valid values: 1 = Market 2 = Limit
→	44	Price		List order price
→	59	TimeInForce	Q	**NOTE:** Must be the same for all legs. Valid values: 0 = Day 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FoK) S = NASDAQ OMX Extension: Good till End of Session (GTS)
→	386	NoTradingSessions		Only set for GTS orders. Can only be set to 1.

→	→	336	TradingSessionID		State type of order expiration. Conditionally required if TimeInForce = GTS. Valid values: 1 = Auction 2 = Continuous Trading
→	528	OrderCapacity			Designates the capacity of the firm placing the order. Valid values: P = Principal A = Agency R = Riskless Principal **NOTE:** Required for Fixed Income.
→	529	OrderRestrictions			Restrictions associated with an order. Valid values: B = Issuer Holding **(requires 528=A)** C = Issue Price Stabilization **(requires 528=P)** 5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**
→	20009	OrderReference			*NASDAQ OMX Extension:* Order Reference pass-thru field. **NOTE:** Only available for fixed income.
	Standard Trailer			Y	

8.7.2 List Status – List Ack/Reject (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = N
66	ListID	Y	Unique identifier for list as assigned by sender
429	ListStatusType	Y	Code to represent the status type. Valid values: 1 = Ack
82	NoRpts	Y	Total number of messages required to status complete list. Will always be 1.
431	ListOrderStatus	Y	Code to represent the status of a list order. Valid values: 2 = Received for execution (ack) 7 = Reject
1385	ContingencyType	Q	NASDAQ OMX Extension: Defines the type of contingency. Valid values: 4 = One Updates the Other (OUO) – proportional Quantity Reduction
1386	ListRejectReason		Identifies the reason for rejection of a New Order List message. Valid values: 4 = Too late to enter 5 = Unknown order 6 = Duplicate order (ClOrdID or ListID) 11 = Unsupported order characteristic 99 = Other
83	RptSeq	Y	FIX required field, value is ignored.
444	ListStatusText		Error message on rejects
60	TransactTime		
68	TotNoOrders	Y	FIX required field used to support fragmentation, which is not supported in this solution. Set to 0.
	Standard Trailer	Y	

8.7.3 List Cancel Request (in)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = K

1116	NoRootPartyIDs			Optional repeating group used for on behalf of transactions.
→	1117	RootPartyID	Q	Party identifier.
→	1118	RootPartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	1119	RootPartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 12 = Executing Trader
66	ListID		Y	Unique identifier for list as assigned by sender
60	TransactTime		Y	
	Standard Trailer		Y	

8.7.4 List Status – List Cancel Ack/Reject (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = N
66	ListID	Y	Unique identifier for list as assigned by sender
429	ListStatusType	Y	Code to represent the status type. Valid values: 2 = Response
82	NoRpts	Y	Total number of messages required to status complete list. Will always be 1.
431	ListOrderStatus	Y	Code to represent the status of a list order. Valid values: 2 = Cancelling (ack) 7 = Reject
1385	ContingencyType	Q	NASDAQ OMX Extension: Defines the type of contingency. Valid values: 4 = One Updates the Other (OUO) – proportional Quantity Reduction
1386	ListRejectReason		Identifies the reason for rejection of a New Order List message. Valid values: 4 = Too late to enter 5 = Unknown order 99 = Other
83	RptSeq	Y	FIX required field, value is ignored.
444	ListStatusText		Error message on rejects
60	TransactTime		
68	TotNoOrders	Y	FIX required field used to support fragmentation, which is not supported in this solution. Set to 0.
	Standard Trailer	Y	

9 General Quote Handling

9.1 Introduction

The Mass Quote message is used by market makers and other actors with similar responsibilities to send quotes into a market. The quote messages, as described in this section, are typically used to send continuous unsolicited quotes in markets with tradable quoting. Such quotes are sent by quote issuers (market makers, specialists, liquidity providers or similar), i.e. actors that have an obligation to provide continuous liquidity in the market.
A quote is two-sided, i.e. normally contains both bid and offer price and size. Many marketplaces limit market makers to one (two-sided) quote in each security.
The Mass Quote message allows the user to submit multiple quotes in a single message.
Responses (acks / rejects) for Mass Quote messages are subject to bilateral agreement between parties and/or as specified in the QuoteResponseLevel (301) field of the request message.

Mass Quotes are also used to enter indicative (non-tradable) quotes. See below for details.

NOTE: The Time-in-Force for continuous quotes is considered to be *Day* in this solution.

9.2 Solution restrictions

Quoting in FIX for Genium INET has the following restrictions:

- Only one two-sided quote per actor per instrument is allowed. This simplifies quote cancellation and generation of quote identifiers, see below.
- The response to a Mass Quote is restricted to negative acks (QuoteResponseLevel = 1). Indicative quotes do allow individual quote acknowledgements.
- All quotes are assumed to be valid until end of day (or until canceled).
- Replacing a quote is as simple as sending a new Mass Quote for the same instrument(s).
- Cancel of a mass quote is achieved by sending a new mass quote with all prices and quantities to 0 (see section Quote Cancellation).
- The pass-thru fields (Account and AllocID) supported in order entry and trade reporting, are **not** supported in quoting transactions.

9.3 Quote Modification

Quote modification is accomplished through the use of the same messages as when adding a quote, i.e. through the Mass Quote message. Replacing a quote in a single quote market is straightforward as every update replaces the old one based on the quote issuer, security (series) and side.

9.4 Quote Cancellation

A quote can be canceled (or withdrawn) by sending a Mass Quote message with bid and offer prices and sizes all set to zero:

- BidPx (132) = 0
- OfferPx (133) = 0
- BidSize (134) = 0
- OfferSize (135) = 0

9.5 Indicative Quotes

An Indicative quote is a special type of quote, where the issuer is able to show his intentions to buy or sell an instrument. An Indicative Quote is *not* tradable.
The indicative quote is sent in using the FIX Mass Quote transaction with QuoteType (537) set to 0 – Indicative.

9.5.1 Undisclosed price and quantity

The indicative quote allows the quote issuer to avoid disclosing the price and/or quantity of a quote. The following four fields can be undisclosed by not being present in the message:

- BidPx (132)
- OfferPx (133)
- BidSize (134)
- OfferSize (135)

9.5.2 Zero price

Indicative quotes may have a zero price. It is possible for both yields and combinations.

9.5.3 Cancelling an Indicative Quote

The Indicative quote is cancelled just like regular quotes; by sending in a new quote with quote price and size set to zero. In fact, since price is allowed to be zero, it is sufficient to set size to zero. A quote with an undisclosed size (the size was not present in the quote message), is cancelled by sending in a new quote with the size present and set to zero.

9.5.4 Maximum number of indicative quotes in one transaction

The system limits the number of indicative quotes that can be entered in a single Mass Quote to **30**.

9.5.5 Request acknowledgement of successful Mass Quotes

For indicative quotes it is possible to request that also successful Mass Quotes get an acknowledgement. This is in contrast to regular quotes where nothing is returned if all quotes were successfully entered.
If QuoteResponseLevel (301) is set to 2 – Acknowledge each quote message, a Mass Quote Ack message will always be returned, even if all quotes were accepted (see chapter 9.11.5 for message details).

9.5.6 Entering Indicative Quotes On-behalf-of another participant

Just as for regular orders, the Parties block of the Mass Quote message for indicative quotes can be used to enter OBO data. See chapter 5.4 for details.

9.6 Main Workflow

9.6.1 Mass Quotes

The Mass Quote message can contain quotes for multiple securities to support applications that allow for the mass quoting of e.g. an option series. Two levels of repeating groups have been provided to minimize the amount of data required to submit a set of quotes for a class of options (e.g. all option series for IBM).
A QuoteSet specifies the first level of repeating fields for the Mass Quote message. It represents a group of related quotes and can, for example, represent an option class.

Each QuoteSet contains a repeating group of QuoteEntries where each entry represents an individual two-sided quote.
NOTE: This flexible construct is not fully supported in this implementation. We limit each Mass Quote to contain a single Quote Set. The number of quote entries supported is limited by the back-end. See note below.

It is possible that the number of Quote Entries for a Quote Set could exceed one's physical or practical message size. It may be necessary to fragment a message across multiple quote messages.

NOTE: The maximum number of quotes in a Mass Quote of *tradable quotes* is configured in the back-end system. See relevant documentation. See section 9.5.4 for details on the limit for indicative quotes.

The grouping of quotes is as follows:

- NoQuoteSets – specifies the number of sets of quotes contained in the message. Will always be one in this solution.
    - QuoteSetID – Is a unique ID given to the quote set within the message. Required in FIX. Will be ignored by the back-end.
    - TotQuoteEntries – defines the number of quotes for the quote set across all messages
    - NoQuoteEntries – defines the number of quotes contained within this message for this quote set
        - QuoteEntryID – Is a unique ID given to a specific quote entry. Can be set to 1, since only one quote per instrument is allowed.
        - Information regarding the security/book to which the quote belong
        - Information regarding the specific quote (bid/ask size and price).

NOTE: It is strongly recommended to set the QuoteEntryIDs as an increasing number starting from 1 on the first entry in each Mass Quote message. This enables the quote issuer to easily identify what quotes have been rejected in case that happens.

9.6.1.1 Limitations

The Mass Quote message can be populated with quotes for different securities as long as they belong to the same partition in Genium INET. Please see relevant Genium INET documentation for information on how to tell which partition a security belongs.

9.6.2 Mass Quote Acknowledgement

Mass Quote Acknowledgement is used as the application level response to a Mass Quote message. The Mass Quote Acknowledgement contains a field for reporting the reason in the event that the entire quote is rejected (QuoteRejectReason [300]). The Mass Quote Acknowledgement also contains a field for each quote that is used in the event that the quote entry is rejected (QuoteEntryRejectReason [368]). The ability to reject an individual quote entry is important so that the majority of quotes can be successfully applied to the market instead of having to reject the entire Mass Quote for a minority of rejected quotes. The Mass Quote Ack has three uses:

- Acknowledge all quotes. See 9.11.5 for message details. **NOTE:** This is only available for *indicative quotes* and only when QuoteResponseLevel is set to 2 on the inbound Mass Quote.
- Some quotes rejected. See section 9.11.2 for message details.
- All quotes rejected. See 9.11.3 for message details.

9.6.3 Quote Rejects

The Mass Quote Acknowledgement message – is primarily used to reject Mass Quotes.

9.7 Quote Identifiers

9.7.1 Quote Message identifier

Every inbound quote message must be associated with a unique message identifier per FIX session. The message identifier can be used to keep an audit trail of quote updates and is used to link a request message to responses. The message identifier is echoed back on response, fill and other messages that are sent out based on a quote.

- **Quote ID**
  The QuoteID (117) is the message identifier used in Mass Quote messages.

The message identifier is relayed back in the following messages:

Request Message	Response Message	Message Identifier Mapping
Mass Quote	Mass Quote Acknowledgement	MQ.QuoteID → MQA.QuoteID
N/A	Execution Report, Trade Capture Report	MQ.QuoteID → ClOrdID

When alternative fields (“or”) are shown in the table, the field to use depends on what message was last used to update the quote.

NOTE: It is **strongly** recommended that the QuoteIDs are taken from the same numbering series as the ClOrdID in cases where Orders and Quotes are submitted through the same FIX session. Quote issuers using multiple sessions or even trading applications should ensure QuoteID uniqueness.

9.7.2 Quote Entity Identifier

Every quote must be associated with a unique entity identifier. The identifier is used to identify an individual quote when updating quotes. The identifier can be compared to the OrderID (37) of orders, but is normally static over time as the same quote is continuously updated. Another difference from the OrderID is that the quote issuer includes the identifier in the Quote messages, it is not returned by the marketplace in responses to quotes.

- Quote Entry ID
  The QuoteEntryID (299) is the entity identifier used in Mass Quote messages. Since only a single quote is allowed per orderbook and side per issuer, there is no strict FIX requirement to set this to a unique value. However, when the back-end rejects a particular quote in a Mass Quote, it only returns the number of the quote entry counting from the first entry in the Mass Quote. So it is strongly recommended to adopt the same numbering scheme for QuoteEntryIDs; *Number the first entry in the Mass Quote 1, the following 2 etc. This way it will be easy to identify rejected entries*.

It should be noted that a quote issuer is never allowed to have more than one two-sided quote in a single book – irrespective of what identifiers are used.

The Quote entity ID (QuoteEntryID) is echoed back in the following messages:

Request Message	Response Message	Quote Entity Identifier Mapping
Mass Quote	Mass Quote Acknowledgement	MQ.QuoteEntryID → MQA.QuoteEntryID

N/A	Execution Report, Trade Capture Report	MQ.QuoteID → ClOrdID

When alternative fields ("or") are shown in the table, the field to use depends on what message was last used to update the quote.

9.8 Quote Response Level

Derivative markets are characterized by high bandwidth consumption – due to a change in an underlying security price causing multiple (often in the hundreds) of quotes to be recalculated and retransmitted to the market. For that reason the ability for market participants (and the market) to be able to set the level of response requested for a Mass Quote message is specified using the QuoteResponseLevel (301) field.

For *regular Mass Quotes* the only supported value is:

- 1 = Requests acknowledgement of invalid or erroneous quote messages only (negative)

For *Indicative quotes* the following additional value is supported:

- 2 = Acknowledge each quote message

9.9 Quote State Changes

A quote is, in principle, regarded as a permanent representation of interest from the relevant market maker, even though it may not always be externally visible (or implemented in the trading engine). This means the quote always has a state: it can only be "not found" when the market maker does not have any side quoted for a security. Empty (or "zero") quote sides are represented using BidPx, OfferPx, BidSize and OfferSize
= 0. However, zero quotes can also be implemented as non-existing quotes.

Quote state changes are divulged by:

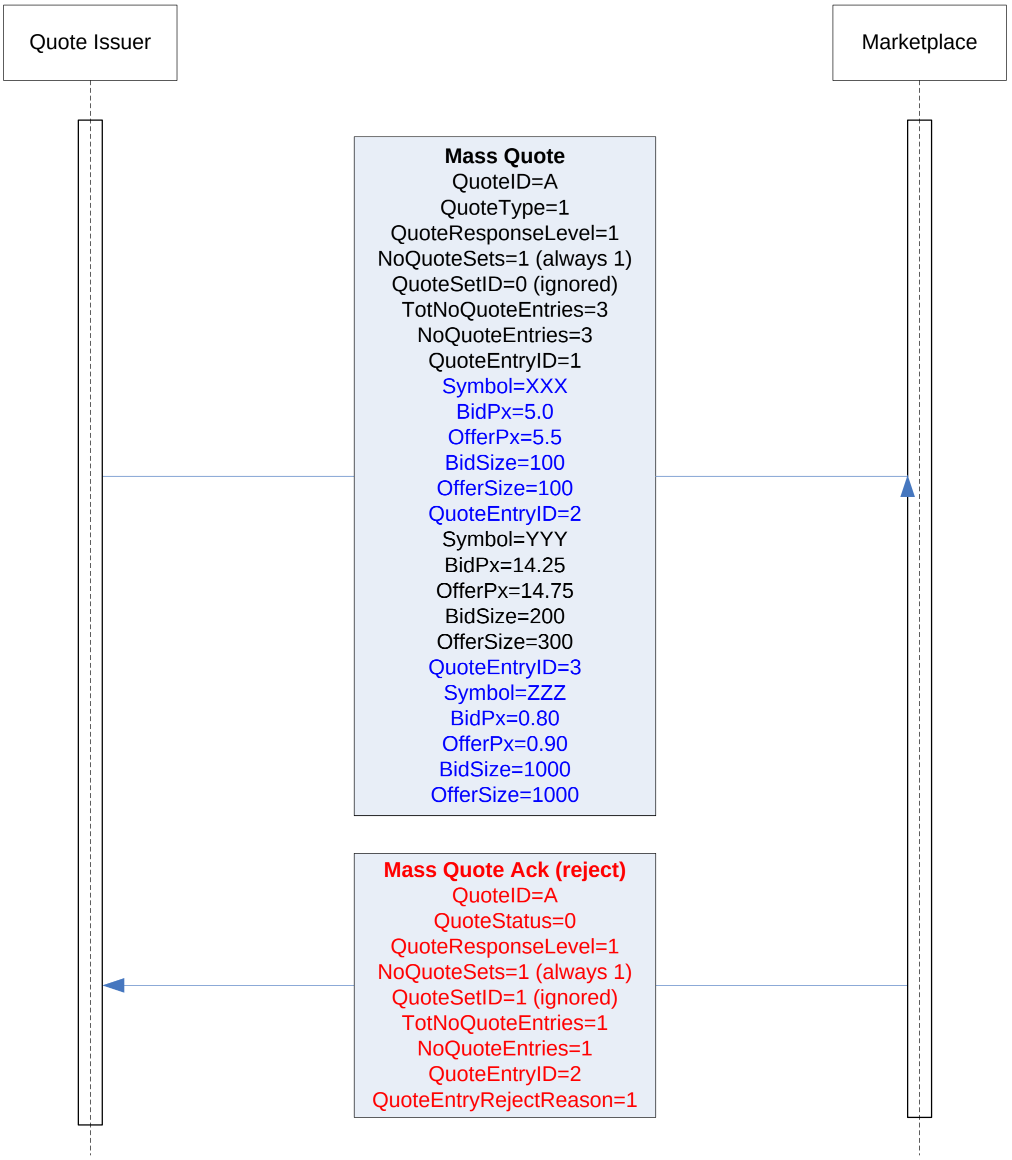
- The Mass Quote Acknowledgement message after a quote update (excluding fills) and subject to the specified or bilaterally agreed QuoteReponseLevel
- Execution Reports after fills

A quote can move from any state to any other state.

9.10 Workflows

9.10.1 Mass Quote with individual reject

In this scenario a Mass Quote with three entries is sent in. The second entry in the Mass Quote is rejected. Notice how the second entry is numbered 2 (QuoteEntryID=2) in the reject. This will be true regardless how QuoteEntryIDs are set in the inbound Mass Quote. So it is strongly advised to set the QuoteEntryIDs similarlyon the inbound Mass Quotes.



Mass Quote
QuoteID=A
QuoteType=1
QuoteResponseLevel=1
NoQuoteSets=1 (always 1)
QuoteSetID=0 (ignored)
TotNoQuoteEntries=3
NoQuoteEntries=3
QuoteEntryID=1
Symbol=XXX
BidPx=5.0
OfferPx=5.5
BidSize=100
OfferSize=100
QuoteEntryID=2
Symbol=YYY
BidPx=14.25
OfferPx=14.75
BidSize=200
OfferSize=300
QuoteEntryID=3
Symbol=ZZZ
BidPx=0.80
OfferPx=0.90
BidSize=1000
OfferSize=1000

Mass Quote Ack (reject)
QuoteID=A
QuoteStatus=0
QuoteResponseLevel=1
NoQuoteSets=1 (always 1)
QuoteSetID=1 (ignored)
TotNoQuoteEntries=1
NoQuoteEntries=1
QuoteEntryID=2
QuoteEntryRejectReason=1

9.11 **Message Details**

9.11.1 Mass Quote (in)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = i
117	QuoteID	Y	Quote issuer assigned message identifier
537	QuoteType	Q	Identifies the type of quote. Valid values: 1 = Tradeable
301	QuoteResponseLevel	Q	Level of Response requested from receiver of quote messages. Valid values: 1 = Acknowledge only negative or erroneous quotes

293	DefBidSize				Default Bid Size
294	DefOfferSize				Default Offer Size
296	NoQuoteSets			Y	Only one Quote set allowed in this solution.
→	302	QuoteSetID		Y	Required in FIX. Will be ignored by the back-end.
→	304	TotNoQuoteEntries		Y	Total number of quotes for all quote sets (will be equal to NoQuoteEntries in this solution).
→	295	NoQuoteEntries		Y	Number of double-sided quotes in Quote Set.
→	→	299	QuoteEntryID	Y	Recommended to be set to an increasing number, starting with 1 in each Mass Quote.
→	→	55	Instrument/Symbol		OMNet short name. Symbol or SecurityID+SecurityIDSource must be set.
→	→	48	Instrument/SecurityID		Orderbook ID
→	→	22	Instrument/SecurityIDSource		Valid values: M = Marketplace-assigned identifier
→	→	132	BidPx	Q	
→	→	133	OfferPx	Q	
→	→	134	BidSize	Q	
→	→	135	OfferSize	Q	
	Standard Trailer			Y	

9.11.2 Mass Quote Acknowledgement – some quotes rejected (out)

Tag	FIX tag name			Req'd	Comment
	Standard Header			Y	MsgType = b
117	QuoteID				
297	QuoteStatus			Y	Identifies the status of the mass quote acknowledgement. Valid values: 0 = Accept
301	QuoteResponseLevel			Q	Level of Response requested from receiver of quote messages. Valid values: 1 = Acknowledge only negative or erroneous quotes
537	QuoteType			Q	Identifies the type of quote. Valid values: 0 = Indicative 1 = Tradeable
453	NoPartyIDs				Optional repeating group used for on behalf of transactions.
→	448	PartyID		Q	Party identifier.
→	447	PartyIDSource		Q	Valid values: D = Proprietary/Custom code
→	452	PartyRole		Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 12 = Executing Trader
296	QuoteSetAckGrp/NoQuoteSets			Y	Multiple quote sets not supported. Will always be 1.
→	302	QuoteSetID		Y	Required in FIX. Will be set to 1.
→	295	NoQuoteEntries		Y	Number of double-sided quotes in Quote Set.
→	→	299	QuoteEntryID	Y	Will be set to the number in the

					order the entries appeared in the incoming Mass Quote (regardless of the QuoteEntryIDs actually set in the Mass Quote). Example: Will be set to 2 if the second entry in the Mass Quote was rejected.
→	→	368	QuoteEntryRejectReason		Reject reason for this individual quote.
58	Text				Will contain the error message(s) from the back-end. **NOTE:** If more than one quote is rejected, the error messages are separated by a “#”.
	Standard Trailer			Y	

9.11.3 Mass Quote Acknowledgement – All Quotes Rejected (out)

Tag	FIX tag name		Req’d	Comment
	Standard Header		Y	MsgType = b
117	QuoteID		Q	From Mass Quote
297	QuoteStatus		Y	Identifies the status of the mass quote acknowledgement. Valid values: 5 = Reject
300	QuoteRejectReason			Reason Quote was rejected. Valid values: 6 = Duplicate Quote IDs 99 = Other
301	QuoteResponseLevel		Q	Level of Response requested from receiver of quote messages. Valid values: 1 = Acknowledge only negative or erroneous quotes
537	QuoteType		Q	Identifies the type of quote. Valid values: 0 = Indicative 1 = Tradeable
453	NoPartyIDs			Optional repeating group used for on behalf of transactions.
→	448	PartyID	Q	Party identifier.
→	447	PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452	PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 12 = Executing Trader
58	Text			
	Standard Trailer		Y	

9.11.4 Mass Quote – indicative quotes (in)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = i

117	QuoteID			Y	Quote issuer assigned message identifier
537	QuoteType			Q	Identifies the type of quote. Valid values: 0 = Indicative
301	QuoteResponseLevel			Q	Level of Response requested from receiver of quote messages. Valid values: 1 = Acknowledge only negative or erroneous quotes 2 = Acknowledge each quote message
453	NoPartyIDs				Optional repeating group used for on behalf of transactions and/or for optional Clearing Firm and Clearing Account.
→	448	PartyID		Q	Party identifier.
→	447	PartyIDSource		Q	Valid values: D = Proprietary/Custom code
→	452	PartyRole		Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 12 = Executing Trader
293	DefBidSize				Default Bid Size
294	DefOfferSize				Default Offer Size
296	NoQuoteSets			Y	Only one Quote set allowed in this solution.
→	302	QuoteSetID		Y	Required in FIX. Will be ignored by the back-end.
→	304	TotNoQuoteEntries		Y	Total number of quotes for all quote sets (will be equal to NoQuoteEntries in this solution).
→	295	NoQuoteEntries		Y	Number of double-sided quotes in Quote Set.
→	→	299	QuoteEntryID	Y	Recommended to be set to an increasing number, starting with 1 in each Mass Quote.
→	→	55	Symbol		OMNet short name. Symbol or SecurityID+SecurityIDSource must be set.
→	→	48	SecurityID		Orderbook ID
→	→	22	SecurityIDSource		Valid values: M = Marketplace-assigned identifier
→	→	132	BidPx		Bid Price. A zero price is allowed. **NOTE:** An undisclosed Bid price is signaled by not setting this field at all.
→	→	133	OfferPx		Offer Price. A zero price is allowed. **NOTE:** An undisclosed Offer price is signaled by not setting this field at all.
→	→	134	BidSize		Bid Quantity. A quantity of 0 means the quote is deleted. **NOTE:** An undisclosed Bid size is signaled by not setting this field at all.
→	→	135	OfferSize		Offer Quantity. A quantity of 0 means the quote is deleted. **NOTE:** An undisclosed Offer size is signaled by not setting this field at all.
	Standard Trailer			Y	

9.11.5 Mass Quote Ack – All quotes accepted (out)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = b

117	QuoteID		Q	From Mass Quote
537	QuoteType		Q	Identifies the type of quote. Valid values: 0 = Indicative
453	NoPartyIDs			Optional repeating group used for on behalf of transactions.
→	448	PartyID	Q	Party identifier.
→	447	PartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	452	PartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 12 = Executing Trader
297	QuoteStatus		Y	Identifies the status of the mass quote acknowledgement. Valid values: 0 = Accepted
301	QuoteResponseLevel		Q	Level of Response requested from receiver of quote messages. Valid values: 2 = Acknowledge each quote message
37	OrderID		Q	Omnet order number.
	Standard Trailer		Y	

10 One-Sided Auctions

A one-sided auction is a call auction where one actor is alone on one side of the book. One-sided auctions are typically used to issue new fixed income instruments, but can also be used to issue other security. The functionality is also used to buy-back previously issued securities.
The auctions are manually initiated by the issuer, and the initiator controls when the auction starts and ends. The time of the uncross is specified when starting the auction.

10.1 **Main Workflow**

10.1.1 Initiating the auction

The auction is started by sending a One-Sided Auction Request message defining the terms of the auction. The marketplace validates the request and responds with a One-Sided Auction Request Ack. When the auction request is approved, an announcement is made to other actors.
NOTE: This announcement is not sent via FIX.

10.1.2 Bidding

Participants may enter bids (in an issuing auction) or offers (in a buy-back auction) for a defined period of time, possibly extending to more than one trading day. Regular FIX order or quoting transactions can be used to enter bids or offers.
Subject to marketplace rules and auction parameters, market data is distributed during the bidding period.
NOTE: Market data is not distributed via FIX.
Preliminary auction results will be sent to the auction initiator.
NOTE: This information is not sent via FIX.

10.1.3 Execution

At a certain time the auction is closed and the auction enters a state called “Issuer Position Modification” where the auction initiator is the only actor allowed to interact with the book. The auction initiator may now enter or modify his bid (or offer) and thereby change the outcome of the auction. He may also, subject to marketplace rules, be allowed to remove orders on the opposite side of the book. Private order updates will be sent to all actors using regular FIX messages (Execution Reports).
During this period, MBO market data will be published to the actors only if the auction is open. For hidden auctions no market data will be produced to the actors.
NOTE: The market data is not published via FIX.
Preliminary auction results will be sent to the auction initiator.
NOTE: This information is not sent via FIX.

10.1.4 Uncross

Finally, the auction is executed and the result published to the actors. Private order updates, executions and trades are sent to the actors. Public market data such as trade tickers and trade statistics is published.
NOTE: The auction result, trade tickers and trade statistics are not provided via FIX.

10.1.5 Cancelling an Auction

It is possible to cancel an ongoing auction by submitting a One-Sided Cancel Auction Request. The response will be sent as a One-Sided Auction Requerst Ack.

10.2 Message Details

10.2.1 One-Sided Auction Request (in)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = UB
20001	AuctionRequestID	Q	Client-generated unique Auction Request identifier
55	Instrument/Symbol		OMNet short name. Symbol or SecurityID+SecurityIDSource must be set.
48	Instrument/SecurityID		Orderbook ID
22	Instrument/SecurityIDSource		Valid values: M = Marketplace-assigned identifier
20002	AuctionType	Q	Type of auction. Valid values: 1 = Issuing auction 2 = Buy-back auction
20010	QuantityLimit		Maximum order quantity allowed for a single order within the auction.
20011	ReferencePrice		Used when price limit checks are enabled.
20003	BookTransparency	Q	Specifies if the auction is open or hidden. Valid values: 1 = Open 2 = Hidden
64	SettlDate		Settlement Date. Only needed for non-standard settlement dates.
730	SettlPrice		Net price for settlement. The net price used when calculating settlement price in an one-sided auction.
20004	AuctionUncrossTime	Q	When the auction uncross will be performed.
	Standard Trailer	Y	

10.2.2 One-Sided Cancel Auction Request (in)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = UC
20001	AuctionRequestID	Q	Client-generated unique Auction Request identifier
20012	OrigAuctionRequestID	Q	ID of the request to cancel.
55	Instrument/Symbol		OMNet short name. Symbol or SecurityID+SecurityIDSource must be set.
48	Instrument/SecurityID		Orderbook ID
22	Instrument/SecurityIDSource		Valid values: M = Marketplace-assigned identifier
	Standard Trailer	Y	

10.2.3 One-Sided Auction Request Ack (out)

Purpose: Accept or Reject of Auction Request or Cancel Auction Request
Identified by: MsgType = UD

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = UD
20001	AuctionRequestID	Q	Unique Auction Request identifier

20012	OrigAuctionRequestID		ID of the request to cancel. Only set on acks for canel requests.
20005	AuctionRequestResult		Shows if the auction request was approved or not. Valid values: 1 = Accepted 2 = Rejected
58	Text		Free text describing used if the auction request was rejected.
	Standard Trailer	Y	

11 Reporting of privately negotiated Trades

11.1 **Introduction**

Trades may, subject to regulations or bilateral agreement, be reported to the marketplace in the following cases:

- Trades negotiated between market participants without using execution mechanisms provided by the Marketplace
- Trades formed at other execution venues but reported to the marketplace for regulatory or publication reasons. Such execution venues may include (systematic) internalizers, ECN's, ATS's, MTF's and others regulated markets. *(Not supported in this solution)*

The marketplace can allow trades to be reported using a set of different mechanisms, the mechanisms currently supported over FIX are:

One-Party Report for Matching
Used when both parties report their trade half. The marketplace matches the reports on security, price, quantity and possibly other conditions.

Two-Party Reports
Used when one of the parties report both sides of a trade by agreement between the parties. Generally allowed only when the marketplace can verify that such an agreement exists between the parties.

11.2 **Identifiers**

11.2.1 Trade Report ID

The TradeReportID (571) is similar to the ClOrdID used for orders and executions. A unique Trade Report ID must be set on all reported trades (TCR) inbound to the marketplace. If a client wants to cancel a previous Trade Report, he can use the TradeReportRefID to refer to the original TraderReportID. There is one important exception to the analogy of ClOrdIDs. The marketplace sets its own TradeReportIDs on outbound TCRs (like confirmed trades).

11.2.2 Trade Report Reference ID

The TradeReportRefID (572) is used to refer to a previous TCR. A submitter of a reported trade can use TradeReportRefID in subsequent cancellations to the reported trade. The marketplace, which sets its own TradeReportIDs on outbound trade confirmations, uses the TradeReportRefID to reference *the submitters TradeReportID* from the original trade report, for example on confirmations to reported trades.

11.2.3 Secondary Trade Report ID

This ID (818) is set by the marketplace on Trade Capture Report Ack messages. It is an interim identifier assigned to the trade that is valid until the trade is confirmed. The Secondary Trade Report ID carries the Genium INET order_number. Analogous to the OrderID on Orders, this is the preferred identifier to use when canceling a previous Trade Capture Report since it requires no lookup in the gateway. To use it in a Trade Cancel, set SecondaryTradeReportRefID (881) to the value received in 818 in the previous TCR Ack message.

11.2.4 Secondary Trade Report Reference ID

The SecondaryTradeReportRefID (881) is the preferred ID to use when canceling a previously reported trade that has not yet been confirmed by the marketplace.

11.2.5 Clearing Accounts

See section 6.3 for details on how to use/set clearing account.

11.2.6 Timestamps

11.2.6.1 Settlement Date

SettlDate (64) contains the Settlement date.

11.2.6.2 Time of Agreement

Time of agreement is shown by the TransBkdTime (483) field.

11.2.6.3 Deferred Publication

Set TradePublishIndicator (1390) to 2 – Defered publication to ask for deferred publication.

11.3 Main Workflow

11.3.1 Trade Capture Report

The Trade Capture Report message is used for the following purposes:

- To submit a new Trade Report (one-party or two-party)
- To update a Trade Report (not supported in this solution)
- To cancel a Trade Report
- For the marketplace to publish trade confirmations (see chapter 13)
- For the marketplace to publish updates to previous trade confirmations (see chapter 13)
- To cancel a confirmed trade (see chapter 13)
- For the marketplace to notify the contra party when a one-party report has been sent in.

11.3.1.1 Submitting a new Trade Report

The TCR message is used to submit off-exchange negotiated trades to the marketplace. Trade Reporting is limited to two models:

- The *one-party report for matching* model, where both parties report the trade to the marketplace. The marketplace always responds with a Trade Capture Report Ack accepting or rejecting the trade report. When both parties have submitted their side of the trade it is matched by the marketplace and a confirmed trade (also using TCR) is issued.
- The *two-party report* model, where one party reports for both sides. An agreement must be in place between the parties. The marketplace always responds with a Trade Capture Report Ack accepting or rejecting the trade report. If the report is accepted, a trade confirmation is sent to both parties.

11.3.1.2 Update a Reported Trade

Request to modify a reported trade is not supported by the system. If the trade has not yet been confirmed, it can be canceled and a new trade reported.
NOTE: To Cancel a Confirmed Trade, see section 13.1.2.

11.3.1.3 Trade Types

The TrdType tag (828) is used to specify the type of trade being reported to the marketplace. A complete list of available trade types can be found in Appendix C, Trade types.
Note that there may be limitations on which trade types are allowed for a certain instrument and/or participant. It is out of scope of this document to fully specify all such rules. Please refer to the member trading rules for further information.

11.3.1.4 Canceling a reported trade

As long as the reported trade has not been confirmed by the marketplace it is possible to cancel it using a TCR. If a Trade Capture Report Ack has been received for the original trade report, users are encouraged to use the SecondaryTradeReportRefID (881) to refer to the original trade report. If a TCR Ack has not been received, the client has to use TradeReportRefID (572) to reference the original trade report. This will require an additional lookup in the gateway and is marginally slower.

11.3.1.5 Marketplace notification to counterparty

When a one-party report for matching is first entered, the marketplace will send a TCR as a notification to the counterparty. See section 11.5.5 for message details.
If the counterparty cancels the trade report causing the notification, a Delete Notification to counterparty will be sent out. See section 11.5.6 for message details.
When the trade report causing the notification gets matched, a Delete Notification to counterparty will be sent out. See workflow example 13.3.2 for details.
NOTE: Notification to counterparty messages will not contain the TargetSubID (57). See section 13.1.3 for details.

11.3.1.6 Marketplace publication of Confirmed Trades

The marketplace uses the TCR to publish confirmed trades, whether auto-matched or reported by clients. See chapter 13 for details.

11.3.2 Trade Capture Report Acknowledgement

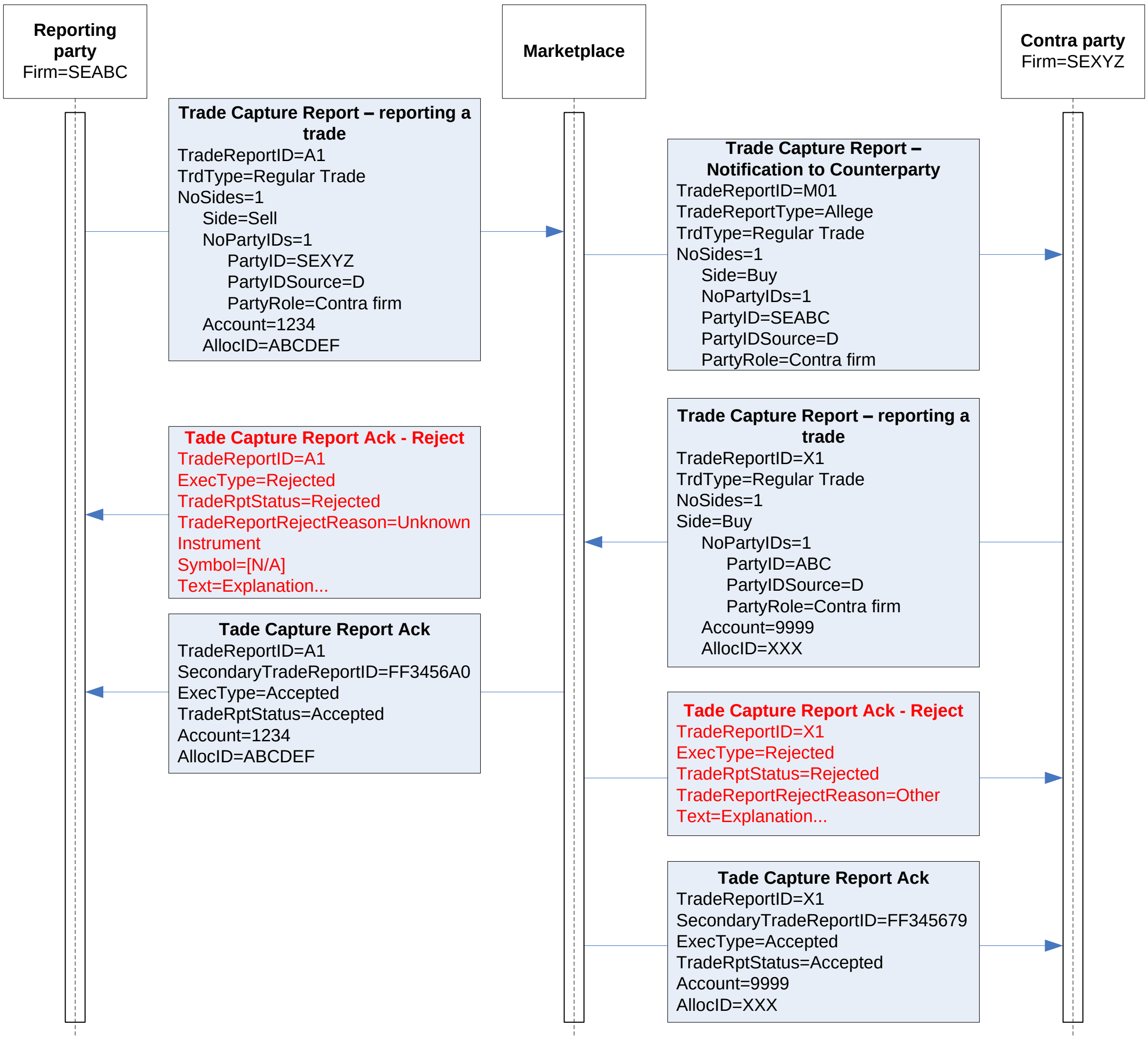
The TCR Ack is used to acknowledge or reject a Trade Capture Report submitted to the exchange. You will always receive a Trace Capture Report Ack when reporting a trade.

11.4 Workflows

The workflows presented here are meant to clarify the use of the most important fields in the Trade Capture Report and Trade Capture Report Ack messages. The workflows are based on the tables in FIX Protocol Specification 5.0 SP1 [2], Volume 5, Appendix B. They have been modified for this solution.

11.4.1 One-party Report for matching

In this example Each Side of a trade reports its own side. When the first party has reported his side, the counterparty receives a notification (see section 11.5.5 for message details).
When both sides have reported the trade it is matched. The resulting confirmation sent out to both reporting parties is described in detail in chapter 13.



Trade Capture Report – reporting a trade
TradeReportID=A1
TrdType=Regular Trade
NoSides=1
Side=Sell
NoPartyIDs=1
PartyID=SEXYZ
PartyIDSource=D
PartyRole=Contra firm
Account=1234
AllocID=ABCDEF

Trade Capture Report – Notification to Counterparty
TradeReportID=M01
TradeReportType=Allege
TrdType=Regular Trade
NoSides=1
Side=Buy
NoPartyIDs=1
PartyID=SEABC
PartyIDSource=D
PartyRole=Contra firm

Trade Capture Report – reporting a trade
TradeReportID=X1
TrdType=Regular Trade
NoSides=1
Side=Buy
NoPartyIDs=1
PartyID=ABC
PartyIDSource=D
PartyRole=Contra firm
Account=9999
AllocID=XXX

Tade Capture Report Ack - Reject
TradeReportID=A1
ExecType=Rejected
TradeRptStatus=Rejected
TradeReportRejectReason=Unknown Instrument
Symbol=[N/A]
Text=Explanation...

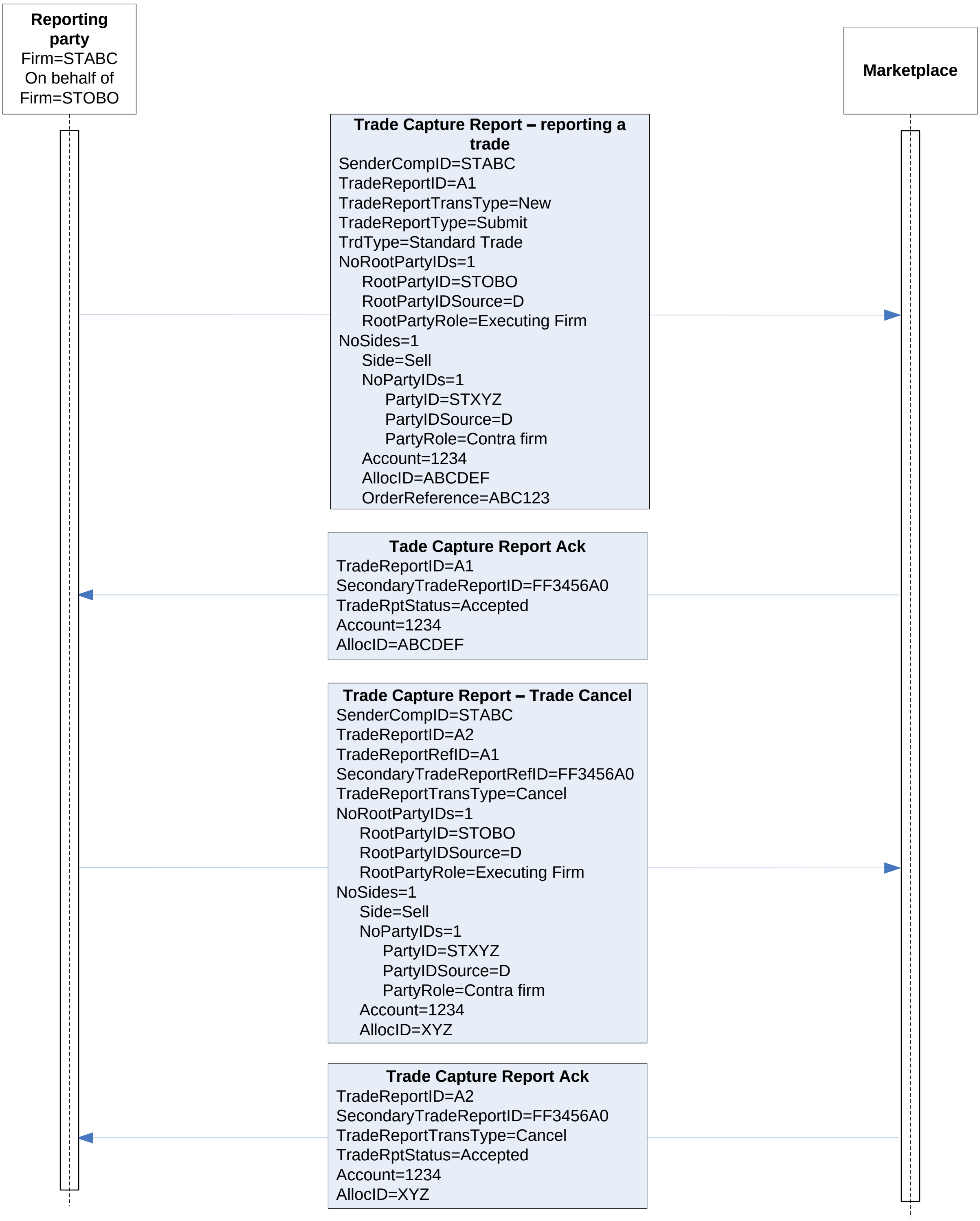
Tade Capture Report Ack
TradeReportID=A1
SecondaryTradeReportID=FF3456A0
ExecType=Accepted
TradeRptStatus=Accepted
Account=1234
AllocID=ABCDEF

Tade Capture Report Ack - Reject
TradeReportID=X1
ExecType=Rejected
TradeRptStatus=Rejected
TradeReportRejectReason=Other
Text=Explanation...

Tade Capture Report Ack
TradeReportID=X1
SecondaryTradeReportID=FF345679
ExecType=Accepted
TradeRptStatus=Accepted
Account=9999
AllocID=XXX

11.4.2 Cancel a Reported Trade that has not yet been matched

Firm STABC enters a one-party trade report on behalf of firm STOBO. STABC then cancels the trade report before it has been matched.
NOTE: A cancel generates a delete notification to the counterparty.
NOTE 2: A cancel will only be accepted *before* the report has been matched and confirmed.



Trade Capture Report – reporting a trade

SenderCompID=STABC
TradeReportID=A1
TradeReportTransType=New
TradeReportType=Submit
TrdType=Standard Trade
NoRootPartyIDs=1
    RootPartyID=STOBO
    RootPartyIDSource=D
    RootPartyRole=Executing Firm
NoSides=1
    Side=Sell
    NoPartyIDs=1
        PartyID=STXYZ
        PartyIDSource=D
        PartyRole=Contra firm
    Account=1234
    AllocID=ABCDEF
    OrderReference=ABC123

Tade Capture Report Ack

TradeReportID=A1
SecondaryTradeReportID=FF3456A0
TradeRptStatus=Accepted
Account=1234
AllocID=ABCDEF

Trade Capture Report – Trade Cancel

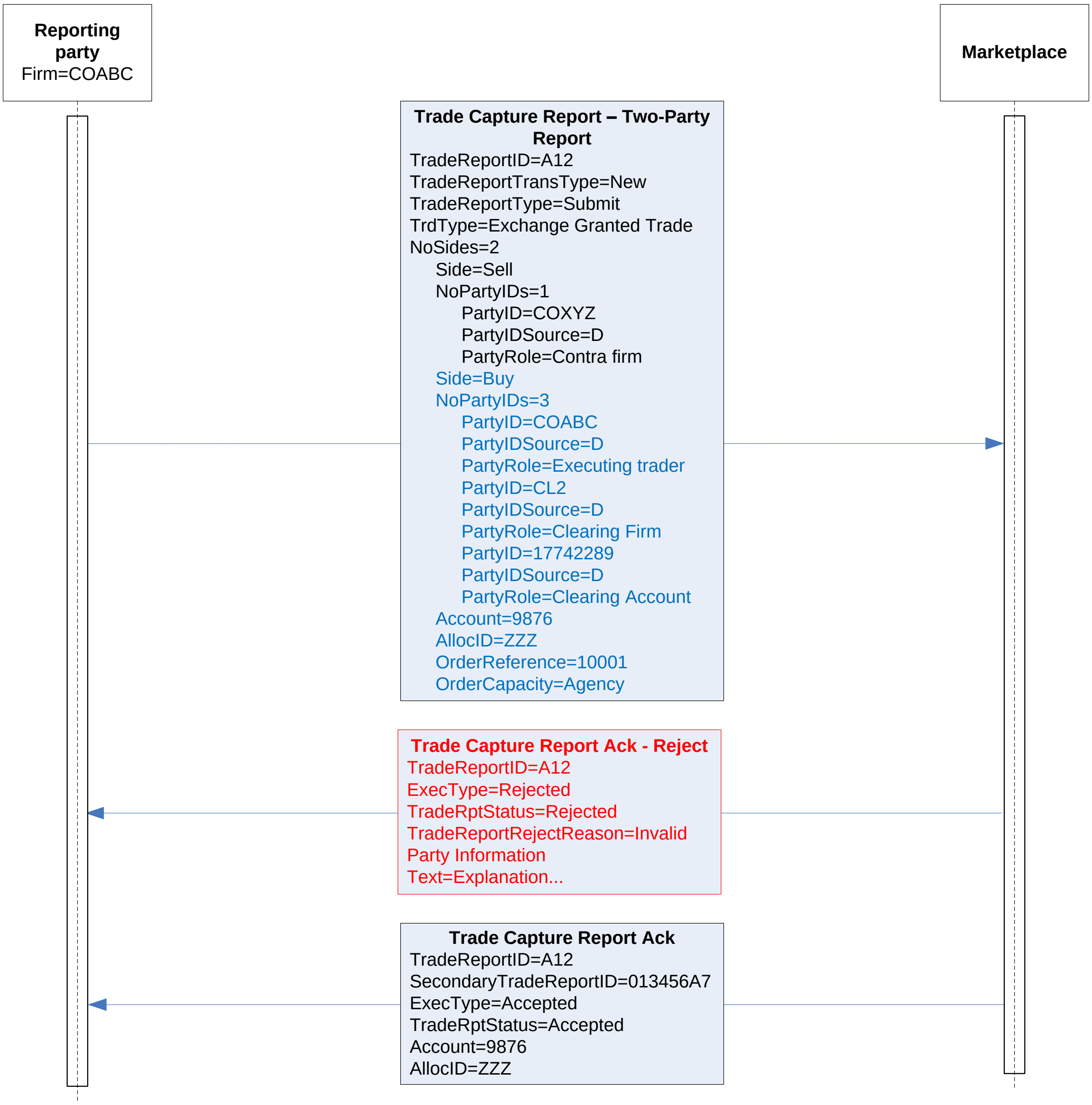
SenderCompID=STABC
TradeReportID=A2
TradeReportRefID=A1
SecondaryTradeReportRefID=FF3456A0
TradeReportTransType=Cancel
NoRootPartyIDs=1
    RootPartyID=STOBO
    RootPartyIDSource=D
    RootPartyRole=Executing Firm
NoSides=1
    Side=Sell
    NoPartyIDs=1
        PartyID=STXYZ
        PartyIDSource=D
        PartyRole=Contra firm
    Account=1234
    AllocID=XYZ

Trade Capture Report Ack

TradeReportID=A2
SecondaryTradeReportID=FF3456A0
TradeReportTransType=Cancel
TradeRptStatus=Accepted
Account=1234
AllocID=XYZ

11.4.3 Two-party Report

The reporting party reports for both sides. When the Two-Party report is accepted a confirmation will be sent out to both parties. See chapter 13 for details.



Trade Capture Report – Two-Party Report
TradeReportID=A12
TradeReportTransType=New
TradeReportType=Submit
TrdType=Exchange Granted Trade
NoSides=2
    Side=Sell
    NoPartyIDs=1
        PartyID=COXYZ
        PartyIDSource=D
        PartyRole=Contra firm
    Side=Buy
    NoPartyIDs=3
        PartyID=COABC
        PartyIDSource=D
        PartyRole=Executing trader
        PartyID=CL2
        PartyIDSource=D
        PartyRole=Clearing Firm
        PartyID=17742289
        PartyIDSource=D
        PartyRole=Clearing Account
    Account=9876
    AllocID=ZZZ
    OrderReference=10001
    OrderCapacity=Agency

Trade Capture Report Ack - Reject
TradeReportID=A12
ExecType=Rejected
TradeRptStatus=Rejected
TradeReportRejectReason=Invalid Party Information
Text=Explanation...

Trade Capture Report Ack
TradeReportID=A12
SecondaryTradeReportID=013456A7
ExecType=Accepted
TradeRptStatus=Accepted
Account=9876
AllocID=ZZZ

11.5 Message Details

11.5.1 Trade Capture Report – One-Party Report for Matching (in)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = AE
571	TradeReportID	Y	Client-generated identifier
487	TradeReportTransType	Q	Valid values: 0 = New
856	TradeReportType	Q	Valid values: 0 = Submit
828	TrdType	Q	For valid values, please see Appendix C, Trade types.
570	PreviouslyReported	Y	Indicates if the trade capture report was previously reported to the counterparty Valid values: N = No
1116	NoRootPartyIDs		Number of party id entries (used for on-behalf-of transactions and/or Clearing Firm and Clearing Account)
→	1117 RootPartyID	Q	Party identifier.
→	1118 RootPartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	1119 RootPartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 83 = Clearing Account
55	Instrument/Symbol		OMNet short name. Either Symbol or SecurityID+SecurityIDSource must be set.
48	Instrument/SecurityID		Orderbook ID
22	Instrument/SecurityIDSource		Valid values: M = Marketplace-assigned identifier
32	LastQty	Y	Traded quantity
31	LastPx	Y	Trade Price
75	TradeDate	Y	Trade Date. Must be set to a valid date. Required in FIX but ignored.

60	TransactTime			Y	Time of execution/order creation
64	SettlDate				Settlement Date
552	NoSides			Y	Set to 1, only counterparty given
→	54	Side		Y	Valid values: 1 = Buy 2 = Sell
→	37	OrderID		Y	Required in FIX, but ignored
→	453	NoPartyIDs		Q	Normally set to 1 (counterparty). Can be set to 2 if trade is to be given up on entry.
→	→	448	PartyID	Q	Counterparty ID/Take-up Firm
→	→	447	PartyIDSource	Q	Valid values : D = Propr. Code
→	→	452	PartyRole	Q	Valid values: 14 = Giveup Clearing Firm 17 = Contra Firm
→	1	Account			Optional pass-thru field set by client and echoed back by marketplace.
→	70	AllocID			Optional pass-thru field set by client and echoed back by marketplace.
→	528	OrderCapacity			*NASDAQ OMX Extension:* Designates the capacity of the firm placing the order. Valid values: P = Principal A = Agency R = Riskless Principal **NOTE:** Required for Fixed Income.
→	529	OrderRestrictions			*NASDAQ OMX Extension:* Restrictions associated with an order. Valid values: B = Issuer Holding **(requires 528=A)** C = Issue Price Stabilization **(requires 528=P)** 5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**
→	483	TransBkdTime			*NASDAQ OMX Extension:* Time of agreement.
→	20006	CleanPrice			*NASDAQ OMX Extension:* The price of a Bond excluding accrued interest. Only used when reporting REPO trades.
→	20009	OrderReference			*NASDAQ OMX Extension:* Order Reference pass-thru field.
1390	TradePublishIndicator				*NASDAQ OMX Extension:* Indicates if a trade should be reported via a market reporting service. Valid values: 2 = Deferred publication
	Standard Trailer			Y	

11.5.2 Trade Capture Report – Two-Party Report (in)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = AE
571	TradeReportID	Y	Client-generated identifier
487	TradeReportTransType	Q	Valid values: 0 = New
856	TradeReportType	Q	Valid values: 0 = Submit

828	TrdType			Q	For valid values, please see Appendix C, Trade types.
570	PreviouslyReported			Y	Indicates if the trade capture report was previously reported to the counterparty Valid values: N = No
55	Instrument/Symbol				OMNet short name. Either Symbol or SecurityID+SecurityIDSource must be set.
48	Instrument/SecurityID				Orderbook ID
22	Instrument/SecurityIDSource				Valid values: M = Marketplace-assigned identifier
32	LastQty			Y	Traded quantity
31	LastPx			Y	Trade Price
75	TradeDate			Y	Trade Date. Must be set to a valid date. Required in FIX but ignored.
60	TransactTime			Y	Time of execution/order creation
64	SettlDate				Settlement Date
552	NoSides			Y	Set to 2 for two-party reports
→	54	Side		Y	Valid values: 1 = Buy 2 = Sell
→	37	OrderID		Y	Required in FIX, but ignored
→	453	NoPartyIDs		Q	
→	→	448	PartyID	Q	Reporting party/Counterparty ID
→	→	447	PartyIDSource	Q	Valid values : D = Propr. Code
→	→	452	PartyRole	Q	Valid values: 1 = Executing Firm 4 = Clearing Firm 14 = Giveup Clearing Firm 17 = Contra Firm 83 = Clearing Account
→	528	OrderCapacity			*NASDAQ OMX Extension:* Designates the capacity of the party. Valid values: P = Principal A = Agency R = Riskless Principal **NOTE:** Required for Fixed Income.
→	529	OrderRestrictions			*NASDAQ OMX Extension:* Restrictions associated with an order. Valid values: B = Issuer Holding **(requires 528=A)** C = Issue Price Stabilization **(requires 528=P)** 5 = Acting as Market Maker or Specialist in the security **(requires 528=P)**
→	483	TransBkdTime			*NASDAQ OMX Extension:* Time of agreement. **NOTE:** Can only set on the reporting party side.
→	20009	OrderReference			*NASDAQ OMX Extension:* Order Reference pass-thru field.
→	1	Account			Optional pass-thru field set by client and echoed back by marketplace. Only valid for PartyRole = Executing Firm or Contra Firm)

→	70	AllocID		Optional pass-thru field set by client and echoed back by marketplace. Only valid for PartyRole = Executing Firm or Contra Firm)
1390	TradePublishIndicator			*NASDAQ OMX Extension:* Indicates if a trade should be reported via a market reporting service. Valid values: 2 = Deferred publication
	Standard Trailer		Y	

11.5.3 Trade Capture Report Ack (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = AR
571	TradeReportID	Y	The client-generated identifier
818	SecondaryTradeReportID		Genium INET order number.
150	ExecType	Y	Valid values: 0 = Accepted
939	TradeRptStatus	Q	Valid values: 0 = Accepted
55	Instrument/Symbol	Q	OMNet short name
48	Instrument/SecurityID	Q	Orderbook ID
22	Instrument/SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
70	AllocID		Optional pass-thru field set by client and echoed back by marketplace.
1	Account		Optional pass-thru field set by client and echoed back by marketplace.
797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
	Standard Trailer	Y	

11.5.4 Trade Capture Report Ack – Reject (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = AR
571	TradeReportID	Y	The client-generated identifier
150	ExecType	Y	Valid values: 8 = Rejected
939	TradeRptStatus	Q	Valid values: 1 = Rejected
751	TradeReportRejectReason	Q	Valid values: 1 = Invalid Party Information 2 = Unknown Instrument 3 = Unauthorized To Report Trades 4 = Invalid Trade Type 99 = Other
55	Instrument/Symbol	Y	**NOTE:** Set to [N/A]
58	Text		Can contain error message
	Standard Trailer	Y	

11.5.5 Trade Capture Report – Notification to Counterparty (out)

Tag	FIX tag name	Req'd	Comment

	Standard Header	Y	MsgType = AE
571	TradeReportID	Y	Identifier assigned by marketplace
487	TradeReportTransType	Q	Valid values: 0 = New
856	TradeReportType	Q	Valid values: 1 = Alleged
828	TrdType	Q	For valid values, please see Appendix C, Trade types.
573	MatchStatus	Q	Valid values: 1= Uncompared, unmatched or unaffirmed
570	PreviouslyReported	Y	Indicates if the trade capture report was previously reported to the counterparty Valid values: N = No
55	Instrument/Symbol	Q	OMNet short name
48	Instrument/SecurityID	Q	Orderbook ID
22	Instrument/SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
32	LastQty	Y	Traded quantity
31	LastPx	Y	Trade Price
75	TradeDate	Y	Always set to date of trade.
60	TransactTime	Y	Time of execution/order creation
64	SettlDate		Settlement Date
552	NoSides	Y	Always 1 Side
→	54 Side	Y	Valid values: 1 = Buy 2 = Sell
→	37 OrderID	Y	OrderID is required in FIX, but set to “NONE”
→	453 NoPartyIDs	Q	Always set to 1
→	→ 448 PartyID	Q	Counterparty ID (reporting party)
→	→ 447 PartyIDSource	Q	Valid values : D = Propr. Code
→	→ 452 PartyRole	Q	Valid values: 17 = Contra Firm
→	483 TransBkdTime		*NASDAQ OMX Extension:* Time of agreement.
→	20006 CleanPrice		*NASDAQ OMX Extension:* The price of a Bond excluding accrued interest. Only used when reporting REPO trades.
1390	TradePublishIndicator		NASDAQ OMX Extension: Indicates if a trade should be reported via a market reporting service. Valid values: 2 = Deferred publication
797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
	Standard Trailer	Y	

11.5.6 Trade Capture Report – Delete Notification to Counterparty (out)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = AE
571	TradeReportID	Y	Identifier assigned by marketplace
572	TradeReportRefID		TradeReportID of previous notification to be

					modified or cancelled.
487	TradeReportTransType			Q	Valid values: 1 = Cancel
856	TradeReportType			Q	Valid values: 1 = Alleged
828	TrdType			Q	For valid values, please see Appendix C, Trade types.
573	MatchStatus			Q	Valid values: 1= Uncompared, unmatched or unaffirmed
570	PreviouslyReported			Y	Indicates if the trade capture report was previously reported to the counterparty Valid values: N = No
55	Instrument/Symbol			Q	OMNet short name
48	Instrument/SecurityID			Q	Orderbook ID
22	Instrument/SecurityIDSource			Q	Valid values: M = Marketplace-assigned identifier
32	LastQty			Y	Traded quantity
31	LastPx			Y	Trade Price
75	TradeDate			Y	Always set to date of trade.
60	TransactTime			Y	Time of execution/order creation
64	SettlDate				Settlement Date
552	NoSides			Y	Always 1 Side
→	54	Side		Y	Valid values: 1 = Buy 2 = Sell
→	37	OrderID		Y	OrderID is required in FIX, but set to “NONE”
→	453	NoPartyIDs		Q	Always set to 1
→	→	448	PartyID	Q	Counterparty ID (reporting party)
→	→	447	PartyIDSource	Q	Valid values : D = Propr. Code
→	→	452	PartyRole	Q	Valid values: 17 = Contra Firm
→	483	TransBkdTime			*NASDAQ OMX Extension:* Time of agreement.
→	20006	CleanPrice			*NASDAQ OMX Extension:* The price of a Bond excluding accrued interest. Only used when reporting REPO trades.
1390	TradePublishIndicator				NASDAQ OMX Extension: Indicates if a trade should be reported via a market reporting service. Valid values: 2 = Deferred publication
797	CopyMsgIndicator				Set to ‘Y’ on Drop Copy messages
	Standard Trailer			Y	

11.5.7 Trade Capture Report – Trade cancel (in)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = AE
571	TradeReportID	Y	Client-generated identifier
572	TradeReportRefID		TradeReportID of report to cancel. Can be used

					instead of SecondaryTradeReportRefID.
881	SecondaryTradeReportRefID				Can be set to SecondaryTradeReportID (Genium INET order number) of the previously received Trade Capture Report Ack, This is the preferred identifier since it requires no lookups.
487	TradeReportTransType			Q	Valid values: 1 = Cancel
856	TradeReportType			Q	Valid values: 0 = Submit
570	PreviouslyReported			Y	Indicates if the trade capture report was previously reported to the counterparty Valid values: N = No
1116	NoRootPartyIDs				Number of party id entries (used for on-behalf-of transactions)
→	1117	RootPartyID		Q	Party identifier.
→	1118	RootPartyIDSource		Q	Valid values: D = Proprietary/Custom code
→	1119	RootPartyRole		Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 12 = Executing Trader
55	Instrument/Symbol				OMNet short name. Either Symbol or SecurityID+SecurityIDSource needs to be set.
48	Instrument/SecurityID				Orderbook ID
22	Instrument/SecurityIDSource				Valid values: M = Marketplace-assigned identifier
32	LastQty			Y	Not validated
31	LastPx			Y	Not validated
75	TradeDate			Y	Trade Date. Must be set to a valid date. Required in FIX but ignored
60	TransactTime			Y	Time of execution/order creation
552	NoSides			Y	Set to 1, only counterparty given
→	54	Side		Y	Counterparty Side. Valid values: 1 = Buy 2 = Sell
→	37	OrderID		Y	Required in FIX, but ignored
→	453	NoPartyIDs		Q	Always set to 1 (counterparty)
→	→	448	PartyID	Q	Counterparty ID
→	→	447	PartyIDSource	Q	Valid values : D = Propr. Code
→	→	452	PartyRole	Q	Valid values: 17 = Contra Firm
→	1	Account			Optional pass-thru field set by client and echoed back by marketplace. Only valid on the reporting Side (where PartyRole=Executing Firm)
→	70	AllocID			Optional pass-thru field set by client and echoed back by marketplace. Only valid on the reporting

				Side (where PartyRole=Executing Firm)
	Standard Trailer		Y	

11.5.8 Trade Capture Report Ack – Trade Cancel (out)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = AR
571	TradeReportID	Y	The client-generated identifier
487	TradeReportTransType	Q	Valid values: 1 = Cancel
818	SecondaryTradeReportID	Q	Genium INET order number.
150	ExecType	Y	Type of Execution being reported. Valid values: 4 = Canceled
939	TradeRptStatus	Q	Valid values: 0 = Accepted
55	Instrument/Symbol	Q	Short name of security
48	Instrument/SecurityID	Q	
22	Instrument/SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
70	AllocID		Optional pass-thru field set by client and echoed back by marketplace.
1	Account		Optional pass-thru field set by client and echoed back by marketplace.
797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
	Standard Trailer	Y	

11.5.9 Trade Capture Report Ack – Trade Cancel Reject (out)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = AR
571	TradeReportID	Y	The client-generated identifier
487	TradeReportTransType	Q	Valid values: 1 = Cancel
150	ExecType	Y	Type of Execution being reported. Valid values: 8 = Rejected
939	TradeRptStatus	Q	Valid values: 1 = Rejected
751	TradeReportRejectReason	Q	Valid values: 99 = Other
55	Instrument/Symbol	Y	**NOTE:** Set to [N/A]
58	Text		Free text description of reject
	Standard Trailer	Y	

12 OTC Trade Reporting

12.1 **Introduction**

NASDAQ OMX offers clearing of certain OTC traded products. The OTC trading does however follow different conventions and habits compared to what is used in the exchange traded area. This chapter primarily describes the differences between OTC Trade Reporting and regular reporting of exchange traded instruments described in the previous chapter.

12.2 **Identifiers**

12.2.1 Trade Report ID

The TradeReportID (571) is similar to the ClOrdID used for orders and executions. A unique Trade Report ID must be set on all reported trades (TCR) inbound to the marketplace. There is one important exception to the analogy of ClOrdIDs. The marketplace sets its own TradeReportIDs on outbound TCRs (like confirmed trades).

12.2.2 Trade Report Reference ID

The TradeReportRefID (572) is used to refer to a previous TCR. The marketplace, which sets its own TradeReportIDs on outbound trade confirmations, uses the TradeReportRefID to reference *the submitters TradeReportID* from the original trade report, for example on updates to reported OTC Trades.

12.2.3 Trade ID

The TradeID (1003) field is formatted a bit differently than for regular trades. TradeID will contain the system *trade report number* formatted as a single hex-encoded string. *Trade report number* is a long (64-bit integer). For OTC Trade Reports, TradeID is unique across all instruments over time.

12.3 **OTC Pre-novation trade management**

The OTC trade report workflow typically consists of the following steps:

1. An OTC trade is reported using the transactions described in the previous chapter.
2. Initial validation of the trade
3. Matching of trade reports. If the incoming transaction consists of an unmatched trade report, then the trade is put in the queue waiting to be matched. Matching occurs when two trade reports have identical values in all the fields required for match, have each other as counterparties and opposite buy/sell. For most standard instruments, the following fields are used for matching:
    - Instrument identity
    - Quantity
    - Price
    - Counterparty (equals the participant in the opposite trade report)
    - Buy/sell (each side should have opposite values)

4. Matched trades may be subject to **affirmation**, this means that a representative of the party responsible for clearing of the trade agrees to the correctness of trade reports and to take on the responsibility for the reported trade. Affirmation may be explicit or implicit/automatic, depending on rules connected to the instrument, source of the trade and result of limit checks. Examples of affirmations are:
    - Broker reports and the parties of the trade affirm.
    - A NCM reports a trade and the GCM affirms
5. Matched trades are confirmed for clearing by the clearing house. This means that the clearing house accepts to novate the trade (i.e. agrees to be the seller to the buyer and buyer to the seller of the reported deal) and "**novation**" occurs. This procedure can either be automatic or manual, depending on rules connected to the instrument, source of the trade, parties involved in the deal and result of limits checks.
6. Each party involved in the deal gets a **trade confirmation** as soon as the trade is confirmed by the clearing house.

12.3.1 OTC Trade Report States

The trade report manager is built on the concept that a deal consist of two matched trades (or trade reports). All of the communication in the workflow is however based on the individual trade reports. In order to communicate the total picture regarding where a specific trade is in the workflow there are four main attributes connected to each trade that show the full information where it is in the processing:
The attributes are:

- State, This attribute shows the main state of the "deal". E.g. Un-matched, Matched, Novated, Cancelled)
- Sub state, This attribute shows the main state of the single trade report. E.g. Pending for affirmation, Pending for clearing house confirmation, Rejected by clearing house, etc.
- Reason, This attribute shows the reason for being in a certain sub state.
- Affirmed (if the trade report has been affirmed or not)

12.4 **Message Details**

12.4.1 Trade Capture Report – One-Party Report for Matching (in)

Please see chapter 11.5.1 for details.

12.4.2 Trade Capture Report – Two-Party Report (in)

Please see chapter 11.5.2 for details.

12.4.3 Trade Capture Report Ack (out)

Please see chapter 11.5.3 for details.

12.4.4 Trade Capture Report Ack – Reject (out)

Please see chapter **Error! Reference source not found.** for details.

12.4.5 Trade Capture Report – Notification to Counterparty (out)

Please see chapter 11.5.5 for details.

12.4.6 Trade Capture Report – Delete Notification to Counterparty (out)

Please see chapter 11.5.6 for details.

12.4.7 Trade Capture Report - Clearing Member Accept/Reject OTC Trade (in)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = AE
571	TradeReportID	Y	The client-generated identifier
487	TradeReportTransType	Q	Valid values: 2 = Replace
856	TradeReportType	Q	Valid values: 2 = Accept 3 = Decline
570	PreviouslyReported	Y	Indicates if the trade capture report was previously reported to the counterparty Valid values: N = No
55	Symbol		OMNet short name. Either Symbol or SecurityID+SecurityIDSource must be set.
48	SecurityID		Orderbook ID
22	SecurityIDSource		Valid values: M = Marketplace-assigned identifier
32	LastQty	Y	Required in FIX. Value ignored.
31	LastPx	Y	Required in FIX. Value ignored.
75	TradeDate	Y	Trade Date. Must be set to a valid date. Required in FIX but ignored.
60	TransactTime	Y	Time of execution/order creation. Required in FIX. Value ignored.
1003	TradeID	Q	Must be taken from the outbound OTC Trade Report. **NOTE:** For OTC trades, this is formatted as a single hex-encoded 64-bit value.
	Standard Trailer	Y	

12.4.8 Trade Capture Report – OTC Trade Report Accepted/Rejected (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = AR
571	TradeReportID	Y	From request
150	ExecType	Y	Valid values: 0 = Accepted 8 = Rejected
487	TradeReportTransType	Q	Valid values: 2 = Replace
856	TradeReportStatus	Q	Valid values: 0 = Accepted 8 = Rejected
55	Symbol	Q	OMNet short name. **NOTE:** Set to [N/A] on rejects
48	SecurityID		Orderbook ID

22	SecurityIDSource		Valid values: M = Marketplace-assigned identifier
1003	TradeID	Q	
58	Text		Reject reason
	Standard Trailer	Y	

12.4.9 Trade Capture Report – OTC Trade Report (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = AE
571	TradeReportID	Y	Identifier assigned by marketplace
1003	TradeID	Q	NASDAQ OMX Extension: Unique identifier for trade. **NOTE:** For OTC trades, this is formatted as a single hex-encoded 64-bit value.
1040	SecondaryTradeID		NASDAQ OMX Extension: Trade id assigned by external system.
1126	OrigTradeID		NASDAQ OMX Extension: Used to refer to original trade in case of modifications
1127	OrigSecondaryTradeID		NASDAQ OMX Extension: Original trade id assigned by external system.
572	TradeReportRefID	Q	From inbound TCR
818	SecondaryTradeReportID	Q	Genium INET order_number. Also present in previous TCR Ack message.
487	TradeReportTransType	Q	Valid values: 2 = Replace 4 = Reverse
856	TradeReportType	Q	Valid values: 0 = Submit
828	TrdType		For valid values, please see Appendix C, Trade types.
829	TrdSubType	Q	Further qualification of the trade type (omnet trade_type_c). NASDAQ OMX Extension values: 1001 = Standard. The trade is a normally registered trade. 1002 = Transitory. The trade is placed on a transitory account. 1003 = Overtaking. The trade is a result of a rectify operation. 1004 = Reversing. The trade is a result of a rectify operation. 1005 = Transfer. The trade is a result of a transfer from a daily account. 1008 = Closing. The trade is a result of a closing series operation. 1009 = Issue 1010 = New contract. The trade is a result where delivery is new contract. 1011 = Delivery

					1012 = Dummy trade 1013 = Alias 1014 = Offsetting 1015 = Superseeding 1016 = State change 1017 = Giveup 1018 = Takeup
573	MatchStatus			Q	Valid values: 0 = Compared, matched or affirmed 1 = Uncompared, unmatched, or unaffirmed
880	TrdMatchID			Q	Match ID assigned by the matching engine.
570	PreviouslyReported			Y	Indicates if the trade capture report was previously reported to the counterparty Valid values: N = No
55	Instrument/Symbol			Q	OMNet short name
48	Instrument/SecurityID			Q	Orderbook ID
22	Instrument/SecurityIDSource			Q	Valid values: M = Marketplace-assigned identifier
32	LastQty			Y	Traded quantity
31	LastPx			Y	Trade Price
75	TradeDate			Y	Always set to date of trade.
60	TransactTime			Y	**NOTE:** Contains Time of Trade Execution
64	SettlDate				Settlement date
552	NoSides			Y	Always 2 Sides
→	54	Side		Y	Side. Valid values: 1 = Buy 2 = Sell
→	37	OrderID		Y	Required in FIX. Set to “NONE”.
→	453	NoPartyIDs		Q	Number of party id entries
→	→	448	PartyID	Q	party identifier
→	→	447	PartyIDSource	Q	Valid values : D = Propr. Code
→	→	452	PartyRole	Q	Valid values: 1 = Executing Firm 7 = Entering Firm 12 = Executing Trader 14 = Giveup Clearing Firm (Takeup Firm) 17 = Contra Firm 36 = Entering Trader 38 = Position Account 83 = Clearing Account 1001 = Confirmed by Firm (NASDAQ OMX Extension) 1002 = Confirmed by User (NASDAQ OMX Extension) 1003 Reported by Firm (NASDAQ OMX Extension) 1004 Reported by User (NASDAQ OMX Extension)

				1005 = Affirmed by Firm (NASDAQ OMX Extension) 1006 = Affirmed by User (NASDAQ OMX Extension) 1007 = Give-up Account (NASDAQ OMX Extension)
→	483	TransBkdTime		*NASDAQ OMX Extension:* Time of agreement.
→	20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field. **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)

→	77	PositionEffect		Defines the *requested* position update for the account. Valid values:
				C = Close
				O = Open
				D = Default
				M = Mandatory Close (NASDAQ OMX Extension)

→	70	AllocID		Optional pass-thru field set by client and echoed back by marketplace. **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
715		ClearingBusinessDate		
855		SecondaryTrdType		Contains Genium INET deal_source value.
793		SecondaryAllocID		*NASDAQ OMX Extension:* Contains Genium INET Give_up_number.
21000		DealID		*NASDAQ OMX Extension:* Contains the numeric Genium INET deal_number.
21013		TradeReportState	Q	*NASDAQ OMX Extension:* Current state of the OTC Trade Report. Valid values: 0 = None 1 = Unmatched 2 = Paired 3 = Matched 4 = Cancelled 5 = Rejected 6 = Novated 7 = Terminated 8 = Deleted
21014		TradeReportSubState	Q	*NASDAQ OMX Extension:* Current sub state of the OTC Trade Report. Valid values: 0 = None 1 = Pending Cancel 2 = Pending Fixing 3 = Pending Termination 4 = Netted to zero 13 = Pending Clearing Member Acceptance

			14 = Rejected by Clearing Member
21015	TradeReportInstrType		*NASDAQ OMX Extension:* Type of trade report. Valid values: 0 = None 1 = Standard 2 = Tailormade 3 = Fixed Income 4 = Discount Security 5 = FRA 6 = IR Swap 7 = FX 8 = Cash 9 = Repo 10 = Agreement 11 = SSI 12 = Equity 13 = XCUR Swap
21016	TraderReportReason		*NASDAQ OMX Extension:*The reason a trade report is in a certain state or the action to a trade report. Valid values: 0 = None 1 = Counterparty has cancelled 2 = pending counterparty cancel 3 = counterparty has terminated 4 = pending counterparty termination 5 = Party Clearing Member 6 = Counterparty Clearing Member 7 = Party lacks collateral 8 = counterparty lacks collateral 9 = old account lacks collateral 10 = New account lacks collateral 11 = Both accounts are lacking collateral 12 = manually confirmed by clearing house 13 = manually rejected by clearing house 14 = Automatic end of day cleanup 15 = rejected by counterparty 16 = exposure exceeded 17 = exposure exceeded and lacking collateral 18 = record update 19 = Confirmation Due on Termination Date 20 = Configuration Error 21 = Party Exposure Limit Exceeded 22 = Counterparty Exceeded Exposure Limit 23 = Member defined exposure limit exceeded
21017	AuthorizationState		*NASDAQ OMX Extension:*Valid values: 0 = None 1 = Autorized 2 = Needed 3 = Not needed
21018	AffimationState		*NASDAQ OMX Extension:* Valid values: 0 = Not required 1 = Holding

			2 = Affirmed by Party 3 = Automatically Affirmed 4 = Rejected 5 = Auto Limit Exceeded
21019	DeliveryUnit		*NASDAQ OMX Extension:*
21020	OrigClearingBusinessDate		*NASDAQ OMX Extension:*
21021	StrategyMarker		*NASDAQ OMX Extension:*
797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
	Standard Trailer	Y	

13Trade Confirmation and Management

A confirmed trade occurs when orders or quotes are executed and when privately negotiated trades are approved. The marketplace publishes confirmed trades to counterparties and possibly to other actors involved in the downstream processing of trades. Such actors can include:

- Broker back-office
- Broker clearing firms
- Clearing houses, Central Counter Parties (CCP)
- Central Securities Depositories (CSD)

Subject to marketplace rules, users are also allowed to request amendments and cancelation of previsouly confirmed trades.

NOTE: The Execution Report message is also used to report fills, but this message is primarily intended as responses to orders and quotes, i.e. for front-office use. The Trade Capture Report message as described in this chapter is primarily intended for actors that process trades in the downstream part of the transaction chain – and thereby designed to contain complete trade information. Market Data messages are used to publish public trade information for so called trade tickers.

NOTE 2: Due to the nature of Confirmed Trades, they are not normally sent on an order entry and trade reporting FIX session. But it is possible to enable the publication of Confirmed Trades on any FIX session.

13.1 **Trade Confirmation features**

13.1.1 Marketplace unsolicited modification of a confirmed trade

The marketplace may need to modify a trade after it has been confirmed. In this scenario, a client will receive two Trade Capture Report-confirmation messages. The first Trade Capture Report received will be a *reversal* of the original confirmation. It will have TradeReportTransType set to *Reverse*. The Side field will be the opposite of the original transaction (the logic is that the reversal should net out the original trade report).
The second Trade Capture Report *replaces* the original. It will have TradeReportTransType set to *Replace*.
The format of these two transactions follow the format of the trade it modifies (either auto-matched trade or confirmation). See sections 13.4.1 and 13.4.2 respectively for message details.

NOTE: All types of confirmed trades (including auto-matched trades) can be modified as described above.

13.1.2 Cancelling a Confirmed Trade

It is possible to cancel a confirmed trade. The following conditions must be met:

- Marketplace rules must allow Confirmed Trade Cancellations for the given instrument (see separate documentation)
- Both parties involved in the trade must send in Trade Cancel messages within a given time limit (see separate documentation).

The Confirmed Trade Cancel message must contain the TradeID of the confirmed trade. See 13.4.4 for message details.

13.1.3 Amending a Confirmed Trade

The marketplace may allow confirmed trades to be amended (also called rectified). The exact rules of how and when a confirmed trade is allowed to be changed are out of scope for this specification. The following parameters are allowed to change:

- Account
- PositionEffect (Open/Close position)
- AllocID/CustomerInfo (pass-though information)

In addition, the trade can be split into multiple accounts using the rectify trade transaction.

The Trade Capture Report – Rectify Confirmed Trade message is used to rectify a trade (see section 13.4.7 for message details). The fields within the NoAllocs repeating group is used to change the trade parameters listed above. To split a trade into multiple accounts, use multiple instances of this repating group.

13.1.3.1 Opening or Closing the position

There are two flavours of rectify trades:

- If you *only* want to request closing of the position, then tag 77 can be set. In this case, no other change (Account, AllocID or trade split) is allowed in the transaction. This request can be processed without extensive validation from the clearing system.
  NOTE: ***This option is currently unavailable. Setting tag 77 to C will cause a reject. The same result can be achieved by using the AllocPositionEffect (tag 1047) within the NoAllocs repeating group.***
- All other changes are validated by the clearing system.

NOTE: In FIX, setting tag 77 to C – Close indicates that the first flavor is used. Then no other changes are allowed in the rectify trade transaction.

13.1.4 Give-Ups

It is possible to give up a confirmed trade to another member. Use the Allocation Instruction message to request give-up. The Take-Up gets notified via a Allocation Report message. The Take-Up firm uses the Allocation Instruction message to accept or reject the give-up.

13.1.5 Automatic Give-Ups

A trade can be automatically given up to another account. An automatic give up is executed if enabled by the CCP for the product and the take up account. Automatic give up can be initated from the Trade Capture Report message:

In Trade Capture Report there are a number of tags used to specify the take up account. For One-Party Report for Matching (in) the following tags are used:

- Tag 452 is set to 14
- Tag 448 state the take up member
- Tag 1119 is set to 83
- Tag 1117 state the take up clearing account ID
- Tag 20009 can be used for give up free text

For Two-Party Report (in):

- Tag 452 is set to 14
- Tag 448 state the take up member
- Tag 452 is set to 83
- Tag 448 state the take up clearing account ID
- Tag 20009 can be used for give up free text

The automatic take up is notified to the take up party via an Allocation Report message. This message should be interpreted as informational message based on the text supplied in tag 58 in Allocation Instruction – Give Up Request (out).

13.1.6 Missing TargetSubIDs on some outbound Trade Capture Reports

Some outbound Trade Capture Report messages will not contain a TargetSubID commonly used to identify the trader that originally entered the transaction. The reason is that in some cases there has been no original transaction prior to receiving a TCR from the marketplace. The two situations are:

- When receiving a two-party confirmation to counterparty (in this case the counterparty reported the trade). See section 13.4.3 for message details.
- When receiving a notification to counterparty (in this case the counterparty has issued a one-party report and the counterparty receives a notification. See section 11.5.5 for message details.
- When receiving a Delete Notification to counterparty. See section 11.5.6 for message details.

13.1.7 Timestamps

13.1.7.1 Settlement Date

SettlDate (64) contains the Settlement date.

13.1.7.2 Time of Agreement

Time of agreement is shown by the TransBkdTime (483) field.

13.1.7.3 Time of Execution

Time of Execution is shown by the TransactTime (60) field.

13.1.7.4 Deferred Publication Time

The DeferredPublicationTime (20013) field contains the *number of minutes* the publication of this trade will be delayed. The time is relative to time of agreement (TransBkdTime, tag 483).
NOTE: A value of -1 means until end of day.

13.1.8 Aggressor Indicator

The AggressorIndicator (1057) field is set on auto-matched trades to show which side is the aggressive side. It is found in the TrdCapRptSideGrp on the “own” side.

13.2 Identifiers

13.2.1 Trade Report ID

The TradeReportID (571) is similar to the ClOrdID used for orders and executions. A unique Trade Report ID must be set on all reported trades (TCR) inbound to the marketplace.
The marketplace sets its own TradeReportIDs on outbound TCRs (like confirmed trades).

13.2.2 Trade Report Reference ID

The TradeReportRefID (572) is used to refer to a previous TCR. The marketplace, which sets its own TradeReportIDs on outbound trade confirmations, uses the TradeReportRefID to reference *the submitters TradeReportID* from the original trade report, for example on confirmations to reported trades.

13.2.3 Secondary Trade Report ID

This ID (818) is set by the marketplace on Trade Capture Report Ack messages. It is an interim identifier assigned to the trade that is valid until the trade is confirmed. The Secondary Trade Report ID carries the Genium INET order_number. Analogues to the OrderID on Orders, this is the preferred identifier to use when canceling a previous Trade Capture Report since it requires no lookup in the gateway.
Secondary Trade Report ID is also set in confirmations.

13.2.4 Trade Match ID

The TrdMatchID (880) contains the match id generated by the system. TrdMatchID will hold a 16 byte Base64-encoded string based on the 12 first bytes of the 16 byte binary match_id.
The encoding is performed according to RFC 2045 [4].

13.2.5 Trade ID

TradeID is an identifier unique per day and orderbook, assigned by the marketplace on confirmed trades. TradeID is formatted as a string containing two hex-encoded integers separated by a colon ":". The format is:

instrument_type:trade_number (where instrument_type and trade_number are omnet field names).

Trade ID is unique per day (regardless of order book).
Trade ID is also used to cancel and rectify confirmed trades.
NOTE: In certain cituations, such as when rectifying a trade, the value zero can be used as instrument_type part (i.e. TradeID = 0:x, where x is the trade number).

13.2.6 Original Trade ID

The OrigTradeID (1126) is a field that is used when the marketplace publishes updates to confirmed trades. As the name suggests, it is used to refer to the Trade ID of the original trade. It has the same format as TradeID.
Whenever the marketplace modifies a confirmed trade this sequence of messages is followed:

1. A Trade Capture Report (TCR) reversing the previous trade is issued.
2. A TCR replacing the original is sent out.

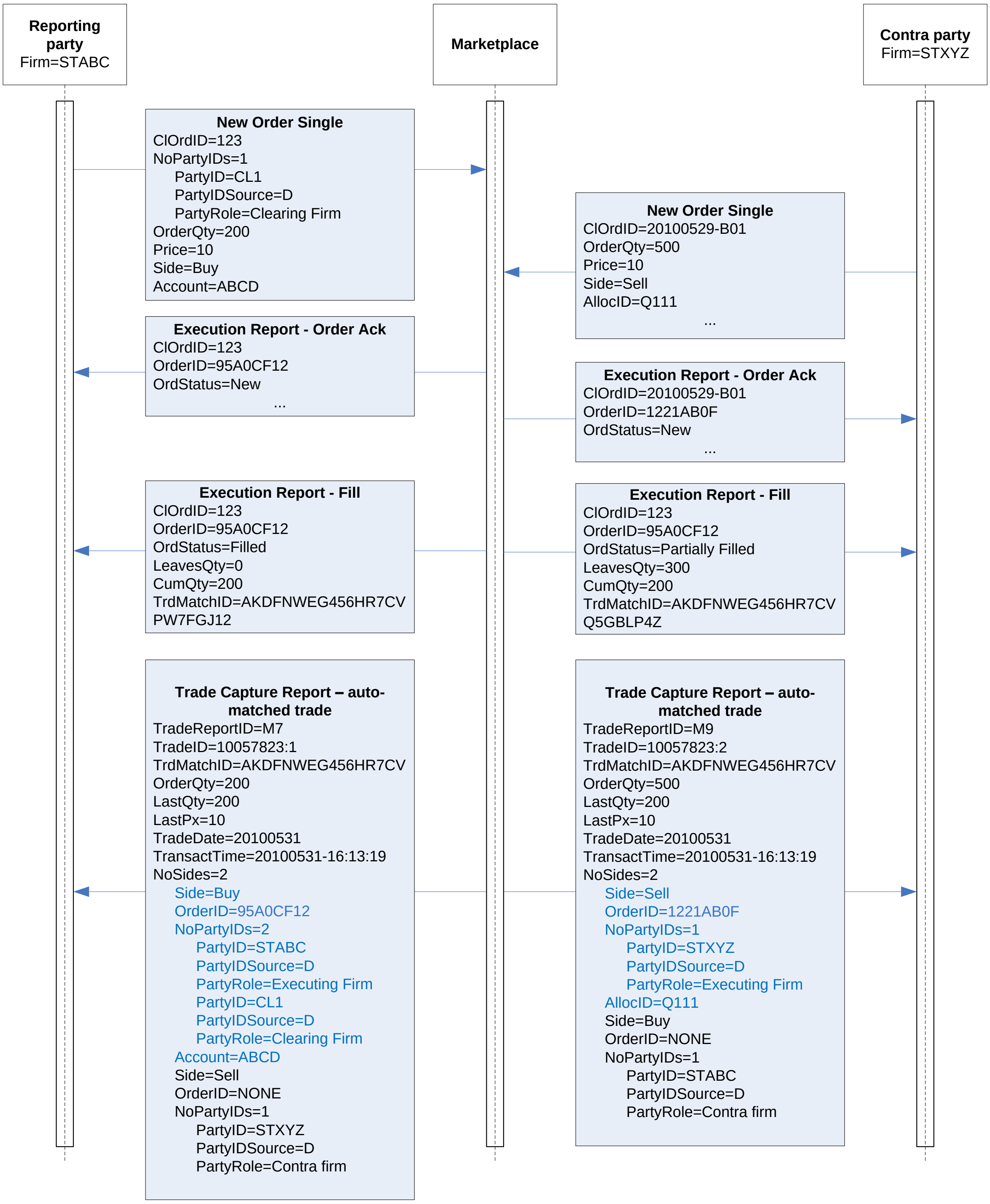
13.2.7 Deal ID

DealID (21000) contains the Genium INET deal_number as a decimal number (FIX datatype: int).

13.3 Workflows

13.3.1 Trade Confirmation for an order that was matched

A regular order is placed in the book. When it is matched the client receives an Execution Report – Fill. In addition, at a later point a Trade Capture Report – auto-matched trade is received.
NOTE: in a typical setup, the confirmations are sent on a separate back-office FIX session to the client.
NOTE 2: For derivatives the Contra Side of the auto-matched trade will not be shown.



New Order Single
ClOrdID=123
NoPartyIDs=1
PartyID=CL1
PartyIDSource=D
PartyRole=Clearing Firm
OrderQty=200
Price=10
Side=Buy
Account=ABCD

New Order Single
ClOrdID=20100529-B01
OrderQty=500
Price=10
Side=Sell
AllocID=Q111
...

Execution Report - Order Ack
ClOrdID=123
OrderID=95A0CF12
OrdStatus=New
...

Execution Report - Order Ack
ClOrdID=20100529-B01
OrderID=1221AB0F
OrdStatus=New
...

Execution Report - Fill
ClOrdID=123
OrderID=95A0CF12
OrdStatus=Filled
LeavesQty=0
CumQty=200
TrdMatchID=AKDFNWEG456HR7CV
PW7FGJ12

Execution Report - Fill
ClOrdID=123
OrderID=95A0CF12
OrdStatus=Partially Filled
LeavesQty=300
CumQty=200
TrdMatchID=AKDFNWEG456HR7CV
Q5GBLP4Z

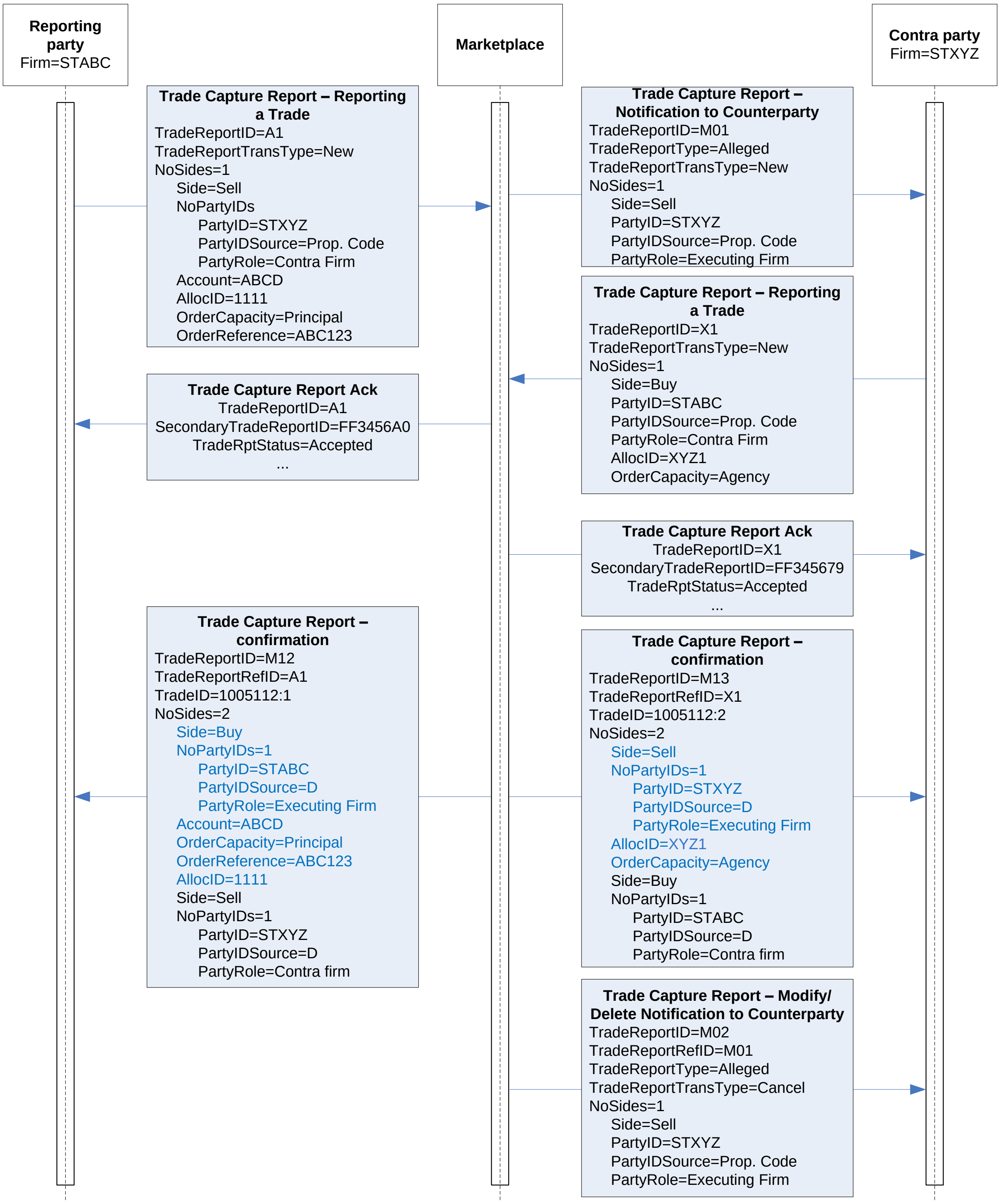
Trade Capture Report – auto-matched trade
TradeReportID=M7
TradeID=10057823:1
TrdMatchID=AKDFNWEG456HR7CV
OrderQty=200
LastQty=200
LastPx=10
TradeDate=20100531
TransactTime=20100531-16:13:19
NoSides=2
Side=Buy
OrderID=95A0CF12
NoPartyIDs=2
PartyID=STABC
PartyIDSource=D
PartyRole=Executing Firm
PartyID=CL1
PartyIDSource=D
PartyRole=Clearing Firm
Account=ABCD
Side=Sell
OrderID=NONE
NoPartyIDs=1
PartyID=STXYZ
PartyIDSource=D
PartyRole=Contra firm

Trade Capture Report – auto-matched trade
TradeReportID=M9
TradeID=10057823:2
TrdMatchID=AKDFNWEG456HR7CV
OrderQty=500
LastQty=200
LastPx=10
TradeDate=20100531
TransactTime=20100531-16:13:19
NoSides=2
Side=Sell
OrderID=1221AB0F
NoPartyIDs=1
PartyID=STXYZ
PartyIDSource=D
PartyRole=Executing Firm
AllocID=Q111
Side=Buy
OrderID=NONE
NoPartyIDs=1
PartyID=STABC
PartyIDSource=D
PartyRole=Contra firm

13.3.2 Confirmation of a Reported Trade

Both parties have reported their side of the trade (Firm STABC is the *buyer*, STXYZ is the *seller*). The marketplace sends out a confirmation to both parties. When the first party sends in his report, the second party gets a notification. When the reports have matched, the notification gets cancelled.

NOTE: in a typical setup, the confirmations are sent on a separate back-office FIX session to the client.
NOTE 2: Notice how the inbound trade reports only contain the contra side. Firm STABC has to set his Account, OrderCapacity etc on the counterparty side (on the sell side even though he is the *buyer*). The outbound confirmations contain all these fields on the correct side (buy for firm STABC).
NOTE 3: Only fields relevant to the example are shown in the diagram.



Trade Capture Report – Reporting a Trade
```
TradeReportID=A1
TradeReportTransType=New
NoSides=1
    Side=Sell
    NoPartyIDs
        PartyID=STXYZ
        PartyIDSource=Prop. Code
        PartyRole=Contra Firm
    Account=ABCD
    AllocID=1111
    OrderCapacity=Principal
    OrderReference=ABC123
```

Trade Capture Report – Notification to Counterparty
```
TradeReportID=M01
TradeReportType=Alleged
TradeReportTransType=New
NoSides=1
    Side=Sell
    PartyID=STXYZ
    PartyIDSource=Prop. Code
    PartyRole=Executing Firm
```

Trade Capture Report – Reporting a Trade
```
TradeReportID=X1
TradeReportTransType=New
NoSides=1
    Side=Buy
    PartyID=STABC
    PartyIDSource=Prop. Code
    PartyRole=Contra Firm
    AllocID=XYZ1
    OrderCapacity=Agency
```

Trade Capture Report Ack
```
TradeReportID=A1
SecondaryTradeReportID=FF3456A0
TradeRptStatus=Accepted
...
```

Trade Capture Report Ack
```
TradeReportID=X1
SecondaryTradeReportID=FF345679
TradeRptStatus=Accepted
...
```

Trade Capture Report – confirmation
```
TradeReportID=M12
TradeReportRefID=A1
TradeID=1005112:1
NoSides=2
    Side=Buy
    NoPartyIDs=1
        PartyID=STABC
        PartyIDSource=D
        PartyRole=Executing Firm
    Account=ABCD
    OrderCapacity=Principal
    OrderReference=ABC123
    AllocID=1111
    Side=Sell
    NoPartyIDs=1
        PartyID=STXYZ
        PartyIDSource=D
        PartyRole=Contra firm
```

Trade Capture Report – confirmation
```
TradeReportID=M13
TradeReportRefID=X1
TradeID=1005112:2
NoSides=2
    Side=Sell
    NoPartyIDs=1
        PartyID=STXYZ
        PartyIDSource=D
        PartyRole=Executing Firm
    AllocID=XYZ1
    OrderCapacity=Agency
    Side=Buy
    NoPartyIDs=1
        PartyID=STABC
        PartyIDSource=D
        PartyRole=Contra firm
```

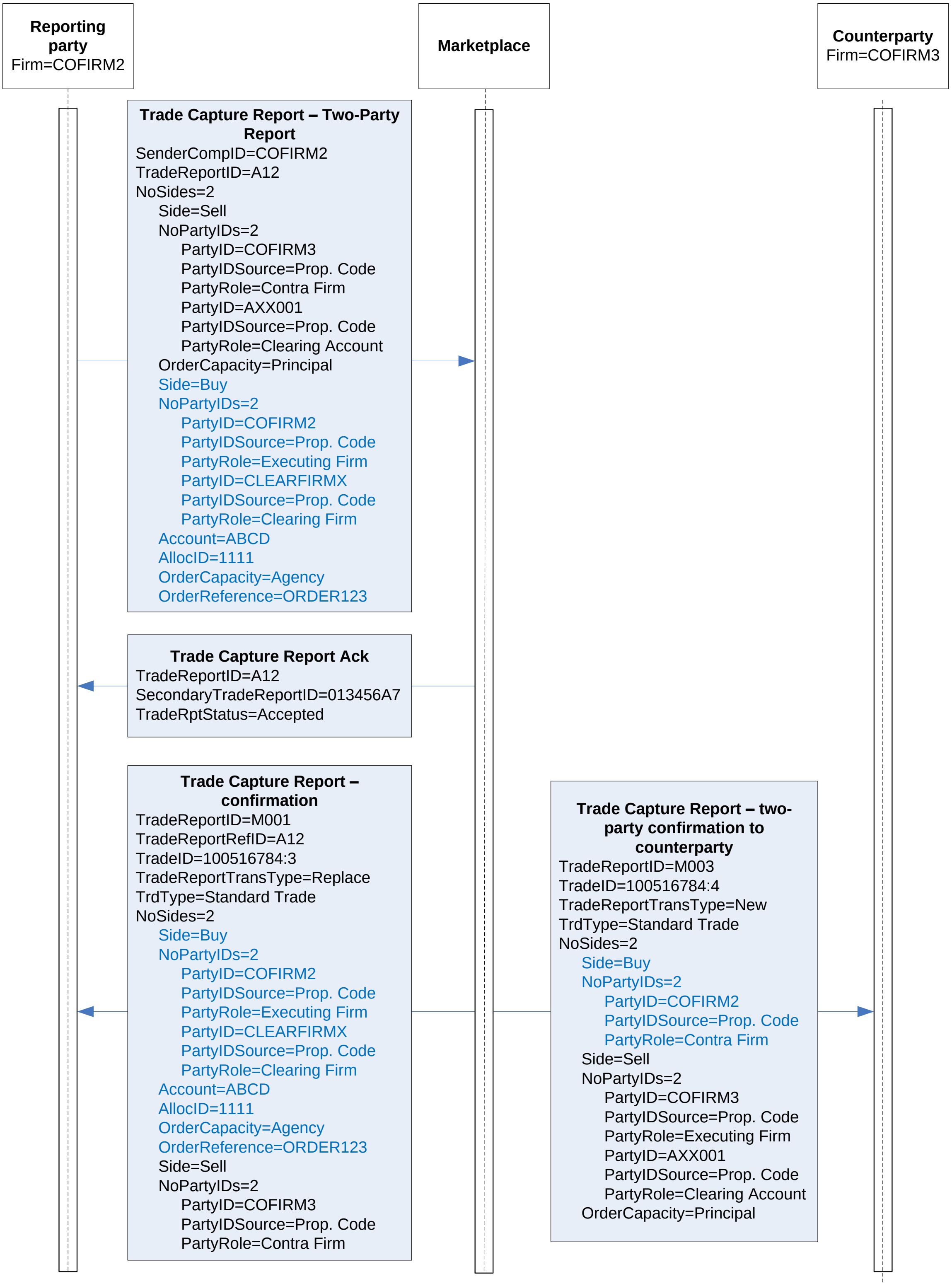
Trade Capture Report – Modify/ Delete Notification to Counterparty
```
TradeReportID=M02
TradeReportRefID=M01
TradeReportType=Alleged
TradeReportTransType=Cancel
NoSides=1
    Side=Sell
    PartyID=STXYZ
    PartyIDSource=Prop. Code
    PartyRole=Executing Firm
```

13.3.3 Confirmation of a two-party Trade Report

The Entering party, COFIRM2, enters a two-party trade report. The counterparty is COFIRM3.

NOTE: If instead COFIRM1 enters the trade on behalf of COFIRM2, the SenderCompID is changed to COFIRM1. All other fields remain the same.

NOTE 2: Only fields relevant to the example are shown in the diagram.



Trade Capture Report – Two-Party Report
SenderCompID=COFIRM2
TradeReportID=A12
NoSides=2
    Side=Sell
    NoPartyIDs=2
        PartyID=COFIRM3
        PartyIDSource=Prop. Code
        PartyRole=Contra Firm
        PartyID=AXX001
        PartyIDSource=Prop. Code
        PartyRole=Clearing Account
    OrderCapacity=Principal
    Side=Buy
    NoPartyIDs=2
        PartyID=COFIRM2
        PartyIDSource=Prop. Code
        PartyRole=Executing Firm
        PartyID=CLEARFIRMX
        PartyIDSource=Prop. Code
        PartyRole=Clearing Firm
    Account=ABCD
    AllocID=1111
    OrderCapacity=Agency
    OrderReference=ORDER123

Trade Capture Report Ack
TradeReportID=A12
SecondaryTradeReportID=013456A7
TradeRptStatus=Accepted

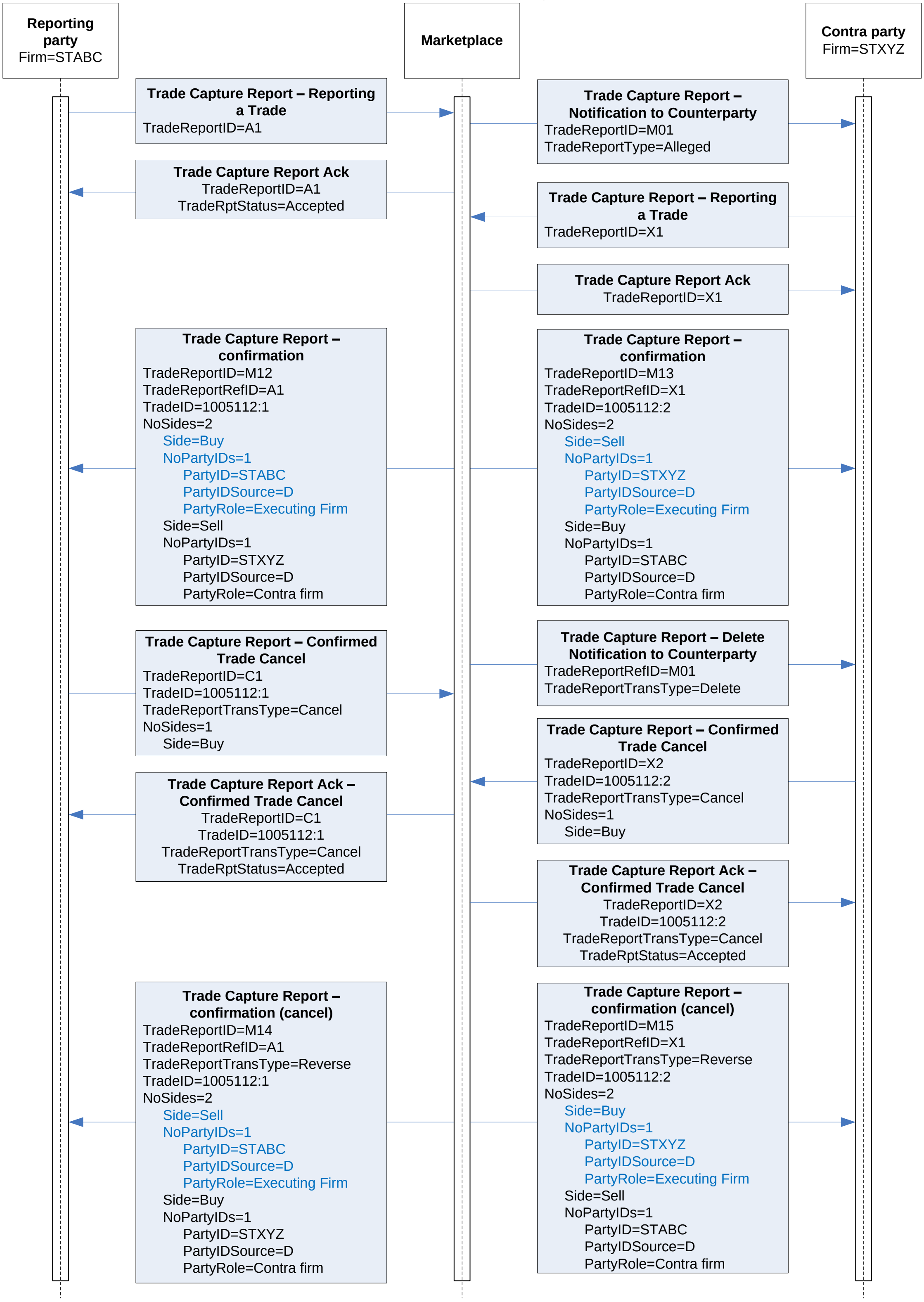
Trade Capture Report – confirmation
TradeReportID=M001
TradeReportRefID=A12
TradeID=100516784:3
TradeReportTransType=Replace
TrdType=Standard Trade
NoSides=2
    Side=Buy
    NoPartyIDs=2
        PartyID=COFIRM2
        PartyIDSource=Prop. Code
        PartyRole=Executing Firm
        PartyID=CLEARFIRMX
        PartyIDSource=Prop. Code
        PartyRole=Clearing Firm
    Account=ABCD
    AllocID=1111
    OrderCapacity=Agency
    OrderReference=ORDER123
    Side=Sell
    NoPartyIDs=2
        PartyID=COFIRM3
        PartyIDSource=Prop. Code
        PartyRole=Contra Firm

Trade Capture Report – two-party confirmation to counterparty
TradeReportID=M003
TradeID=100516784:4
TradeReportTransType=New
TrdType=Standard Trade
NoSides=2
    Side=Buy
    NoPartyIDs=2
        PartyID=COFIRM2
        PartyIDSource=Prop. Code
        PartyRole=Contra Firm
    Side=Sell
    NoPartyIDs=2
        PartyID=COFIRM3
        PartyIDSource=Prop. Code
        PartyRole=Executing Firm
        PartyID=AXX001
        PartyIDSource=Prop. Code
        PartyRole=Clearing Account
    OrderCapacity=Principal

13.3.4 Cancel of a confirmed Trade

If marketplace rules allow it, a confimed trade may be canceled.

NOTE: The last confirmation will not contain the TrdType field.



Trade Capture Report – Reporting a Trade
TradeReportID=A1

Trade Capture Report – Notification to Counterparty
TradeReportID=M01
TradeReportType=Alleged

Trade Capture Report Ack
TradeReportID=A1
TradeRptStatus=Accepted

Trade Capture Report – Reporting a Trade
TradeReportID=X1

Trade Capture Report Ack
TradeReportID=X1

Trade Capture Report – confirmation
TradeReportID=M12
TradeReportRefID=A1
TradeID=1005112:1
NoSides=2
Side=Buy
NoPartyIDs=1
PartyID=STABC
PartyIDSource=D
PartyRole=Executing Firm
Side=Sell
NoPartyIDs=1
PartyID=STXYZ
PartyIDSource=D
PartyRole=Contra firm

Trade Capture Report – confirmation
TradeReportID=M13
TradeReportRefID=X1
TradeID=1005112:2
NoSides=2
Side=Sell
NoPartyIDs=1
PartyID=STXYZ
PartyIDSource=D
PartyRole=Executing Firm
Side=Buy
NoPartyIDs=1
PartyID=STABC
PartyIDSource=D
PartyRole=Contra firm

Trade Capture Report – Delete Notification to Counterparty
TradeReportRefID=M01
TradeReportTransType=Delete

Trade Capture Report – Confirmed Trade Cancel
TradeReportID=C1
TradeID=1005112:1
TradeReportTransType=Cancel
NoSides=1
Side=Buy

Trade Capture Report – Confirmed Trade Cancel
TradeReportID=X2
TradeID=1005112:2
TradeReportTransType=Cancel
NoSides=1
Side=Buy

Trade Capture Report Ack – Confirmed Trade Cancel
TradeReportID=C1
TradeID=1005112:1
TradeReportTransType=Cancel
TradeRptStatus=Accepted

Trade Capture Report Ack – Confirmed Trade Cancel
TradeReportID=X2
TradeID=1005112:2
TradeReportTransType=Cancel
TradeRptStatus=Accepted

Trade Capture Report – confirmation (cancel)
TradeReportID=M14
TradeReportRefID=A1
TradeReportTransType=Reverse
TradeID=1005112:1
NoSides=2
Side=Sell
NoPartyIDs=1
PartyID=STABC
PartyIDSource=D
PartyRole=Executing Firm
Side=Buy
NoPartyIDs=1
PartyID=STXYZ
PartyIDSource=D
PartyRole=Contra firm

Trade Capture Report – confirmation (cancel)
TradeReportID=M15
TradeReportRefID=X1
TradeReportTransType=Reverse
TradeID=1005112:2
NoSides=2
Side=Buy
NoPartyIDs=1
PartyID=STXYZ
PartyIDSource=D
PartyRole=Executing Firm
Side=Sell
NoPartyIDs=1
PartyID=STABC
PartyIDSource=D
PartyRole=Contra firm

13.4 **Message Details**

13.4.1 Trade Capture Report – auto-matched trade (out)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = AE
571	TradeReportID	Y	Identifier assigned by marketplace
1003	TradeID	Q	NASDAQ OMX Extension: Unique identifier for trade
1040	SecondaryTradeID		NASDAQ OMX Extension: Trade id assigned by external system.
1126	OrigTradeID		NASDAQ OMX Extension: Used to refer to original trade in case of modifications
1127	OrigSecondaryTradeID		NASDAQ OMX Extension: Original trade id assigned by external system.
487	TradeReportTransType	Q	Valid values: 0 = New 2 = Replace 4 = Reverse
856	TradeReportType	Q	Valid values: 0 = Submit
829	TrdSubType	Q	Further qualification of the trade type (omnet trade_type_c). NASDAQ OMX Extension values: 1001 = Standard. The trade is a normally registered trade. 1002 = Transitory. The trade is placed on a transitory account. 1003 = Overtaking. The trade is a result of a rectify operation. 1004 = Reversing. The trade is a result of a rectify operation. 1005 = Transfer. The trade is a result of a transfer from a daily account. 1008 = Closing. The trade is a result of a closing series operation. 1009 = Issue 1010 = New contract. The trade is a result where delivery is new contract. 1011 = Delivery 1012 = Dummy trade 1013 = Alias 1014 = Offsetting 1015 = Superseeding 1016 = State change 1017 = Giveup 1018 = Takeup
573	MatchStatus	Q	Valid values: 0 = Compared, matched or affirmed
880	TrdMatchID	Q	Match ID assigned by the matching engine.

570	PreviouslyReported			Y	Indicates if the trade capture report was previously reported to the counterparty Valid values: N = No
55	Instrument/Symbol			Q	OMNet short name
48	Instrument/SecurityID			Q	Orderbook ID
22	Instrument/SecurityIDSource			Q	Valid values: M = Marketplace-assigned identifier
38	OrderQtyData/OrderQty				
32	LastQty			Y	Traded quantity
31	LastPx			Y	Trade Price
75	TradeDate			Y	Always set to date of trade.
60	TransactTime			Y	**NOTE:** Contains Time of Trade Execution
64	SettlDate				Settlement date
552	NoSides			Y	Either 1 (own side only) or 2 (both sides)
→	54	Side		Y	Side. Valid values: 1 = Buy 2 = Sell
→	37	OrderID		Y	OrderID on own Side. Set to “NONE” on counterparty side.
→	453	NoPartyIDs		Q	Number of party id entries
→	→	448	PartyID	Q	party identifier
→	→	447	PartyIDSource	Q	Valid values : D = Propr. Code
→	→	452	PartyRole	Q	Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 14 = Giveup Clearing Firm (Takeup Firm) 17 = Contra Firm 83 = Clearing Account 1001 = Confirmed by Firm (NASDAQ OMX Extension)
→	528	OrderCapacity			*NASDAQ OMX Extension:* Designates the capacity of the firm placing the order. Valid values: P = Principal A = Agency R = Riskless Principal **NOTE:** Required for Fixed Income. **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	529	OrderRestrictions			*NASDAQ OMX Extension:* Restrictions associated with an order. Valid values: B = Issuer Holding **(requires 528=A)** C = Issue Price Stabilization **(requires 528=P)** 5 = Acting as Market Maker or Specialist in the security **(requires 528=P)** **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	20009	OrderReference			*NASDAQ OMX Extension:* Order Reference

				pass-thru field. **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	1	Account		Optional pass-thru field set by client and echoed back by marketplace. Only set on the own Side (where PartyRole=Executing Firm). **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)

→	77	PositionEffect		Defines the position update for the account. Valid values:
				C = Close
				O = Open
				NOTE: for instruments **not** cleared within the system, this field contains the *requested* position effect. The following additional values then applies:
				D = Default
				M = Mandatory Close (NASDAQ OMX Extension)

→	70	AllocID		Optional pass-thru field set by client and echoed back by marketplace. Only set on the own Side (where PartyRole=Executing Firm). **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	1057	AggressorIndicator		Indicates who is the aggressive party in the trade. Valid values: Y = Party is the aggressor N = Party is passive **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	151	LeavesQty		Remaining quantity
	715	ClearingBusinessDate		
	855	SecondaryTrdType		Contains Genium INET deal_source value.
	793	SecondaryAllocID		*NASDAQ OMX Extension:* Contains Genium INET Give_up_number.
	20007	CorrespondingPrice		*NASDAQ OMX Extension:* Corresponding Price/Yield for fixed income related trades.
	20008	Consideration		*NASDAQ OMX Extension:* Consideration/Settlement Amount for fixed income related trades.
	21000	DealID		*NASDAQ OMX Extension:* Contains the numeric Genium INET deal_number.
	797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
		Standard Trailer	Y	

13.4.2 Trade Capture Report – confirmation (out)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = AE

571	TradeReportID	Y	Identifier assigned by marketplace
1003	TradeID	Q	NASDAQ OMX Extension: Unique identifier for trade
1040	SecondaryTradeID		NASDAQ OMX Extension: Trade id assigned by external system.
1126	OrigTradeID		NASDAQ OMX Extension: Used to refer to original trade in case of modifications
1127	OrigSecondaryTradeID		NASDAQ OMX Extension: Original trade id assigned by external system.
572	TradeReportRefID	Q	From inbound TCR
818	SecondaryTradeReportID	Q	Genium INET order_number. Also present in previous TCR Ack message.
487	TradeReportTransType	Q	Valid values: 2 = Replace 4 = Reverse
856	TradeReportType	Q	Valid values: 0 = Submit
828	TrdType		For valid values, please see Appendix C, Trade types.
829	TrdSubType	Q	Further qualification of the trade type (omnet trade_type_c). NASDAQ OMX Extension values: 1001 = Standard. The trade is a normally registered trade. 1002 = Transitory. The trade is placed on a transitory account. 1003 = Overtaking. The trade is a result of a rectify operation. 1004 = Reversing. The trade is a result of a rectify operation. 1005 = Transfer. The trade is a result of a transfer from a daily account. 1008 = Closing. The trade is a result of a closing series operation. 1009 = Issue 1010 = New contract. The trade is a result where delivery is new contract. 1011 = Delivery 1012 = Dummy trade 1013 = Alias 1014 = Offsetting 1015 = Superseeding 1016 = State change 1017 = Giveup 1018 = Takeup
573	MatchStatus	Q	Valid values: 0 = Compared, matched or affirmed
880	TrdMatchID	Q	Match ID assigned by the matching engine.
570	PreviouslyReported	Y	Indicates if the trade capture report was previously reported to the counterparty

					Valid values: N = No
55	Instrument/Symbol			Q	OMNet short name
48	Instrument/SecurityID			Q	Orderbook ID
22	Instrument/SecurityIDSource			Q	Valid values: M = Marketplace-assigned identifier
32	LastQty			Y	Traded quantity
31	LastPx			Y	Trade Price
75	TradeDate			Y	Always set to date of trade.
60	TransactTime			Y	**NOTE:** Contains Time of Trade Execution
64	SettlDate				Settlement date
552	NoSides			Y	Always 2 Sides
→	54	Side		Y	Side. Valid values: 1 = Buy 2 = Sell
→	37	OrderID		Y	Required in FIX. Set to “NONE”.
→	453	NoPartyIDs		Q	Number of party id entries
→	→	448	PartyID	Q	party identifier
→	→	447	PartyIDSource	Q	Valid values : D = Propr. Code
→	→	452	PartyRole	Q	Valid values: 1 = Executing Firm 4 = Clearing Firm 12 = Executing Trader 14 = Giveup Cleaaring Firm (Takeup Firm) 17 = Contra Firm 83 = Clearing Account 1001 = Confirmed by Firm (NASDAQ OMX Extension)
→	528	OrderCapacity			*NASDAQ OMX Extension:* Designates the capacity of the firm placing the order. Valid values: P = Principal A = Agency R = Riskless Principal **NOTE:** Required for Fixed Income. **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	529	OrderRestrictions			*NASDAQ OMX Extension:* Restrictions associated with an order. Valid values: B = Issuer Holding **(requires 528=A)** C = Issue Price Stabilization **(requires 528=P)** 5 = Acting as Market Maker or Specialist in the security **(requires 528=P)** **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	483	TransBkdTime			*NASDAQ OMX Extension:* Time of agreement. **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	20006	CleanPrice			*NASDAQ OMX Extension:* The price of a

				Bond excluding accrued interest. Only used when reporting REPO trades. **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	20009	OrderReference		*NASDAQ OMX Extension:* Order Reference pass-thru field. **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	1	Account		Optional pass-thru field set by client and echoed back by marketplace. Only set on the own Side (where PartyRole=Executing Firm) **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)

→	77	PositionEffect		Defines the position update for the account. Valid values:
				C = Close
				O = Open
				NOTE: for instruments **not** cleared within the system, this field contains the *requested* position effect. The following additional values then applies:
				D = Default
				M = Mandatory Close (NASDAQ OMX Extension)

→	70	AllocID		Optional pass-thru field set by client and echoed back by marketplace. Only set on the own Side (where PartyRole=Executing Firm) **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	151	LeavesQty		Remaining quantity
715	ClearingBusinessDate			
855	SecondaryTrdType			Contains Genium INET deal_source value.
793	SecondaryAllocID			*NASDAQ OMX Extension:* Contains Genium INET Give_up_number.
20007	CorrespondingPrice			*NASDAQ OMX Extension:* Corresponding Price/Yield for fixed income related trades.
20008	Consideration			*NASDAQ OMX Extension:* Consideration/Settlement Amount for fixed income related trades.
20013	DeferredPublicationTime			*NASDAQ OMX Extension:* The number of minutes the publication of this trade will be delayed (relative to time of agreement). **NOTE:** -1 means end of day.
21000	DealID			*NASDAQ OMX Extension:* Contains the numeric Genium INET deal_number.
797	CopyMsgIndicator			Set to ‘Y’ on Drop Copy messages
	Standard Trailer		Y	

13.4.3 Trade Capture Report – two-party confirmation to counterparty (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = AE
571	TradeReportID	Y	Identifier assigned by marketplace
1003	TradeID	Q	NASDAQ OMX Extension: Unique identifier for trade
1040	SecondaryTradeID		NASDAQ OMX Extension: Trade id assigned by external system.
1126	OrigTradeID		NASDAQ OMX Extension: Used to refer to original trade in case of modifications
1127	OrigSecondaryTradeID		NASDAQ OMX Extension: Original trade id assigned by external system.
818	SecondaryTradeReportID	Q	Genium INET order_number.
487	TradeReportTransType	Q	Valid values: 0 = New 2 = Replace 4 = Reverse
856	TradeReportType	Q	Valid values: 0 = Submit
828	TrdType		For valid values, please see Appendix C, Trade types.
829	TrdSubType	Q	Further qualification of the trade type (omnet trade_type_c). NASDAQ OMX Extension values: 1001 = Standard. The trade is a normally registered trade. 1002 = Transitory. The trade is placed on a transitory account. 1003 = Overtaking. The trade is a result of a rectify operation. 1004 = Reversing. The trade is a result of a rectify operation. 1005 = Transfer. The trade is a result of a transfer from a daily account. 1008 = Closing. The trade is a result of a closing series operation. 1009 = Issue 1010 = New contract. The trade is a result where delivery is new contract. 1011 = Delivery 1012 = Dummy trade 1013 = Alias 1014 = Offsetting 1015 = Superseeding 1016 = State change 1017 = Giveup 1018 = Takeup
573	MatchStatus	Q	Valid values: 0 = Compared, matched or affirmed
880	TrdMatchID	Q	Match ID assigned by the matching engine.

570	PreviouslyReported			Y	Indicates if the trade capture report was previously reported to the counterparty Valid values: N = No
55	Instrument/Symbol			Q	OMNet short name
48	Instrument/SecurityID			Q	Orderbook ID
22	Instrument/SecurityIDSource			Q	Valid values: M = Marketplace-assigned identifier
32	LastQty			Y	Traded quantity
31	LastPx			Y	Trade Price
75	TradeDate			Y	Always set to date of trade.
60	TransactTime			Y	**NOTE:** Contains Time of Trade Execution
64	SettlDate				Settlement date
552	NoSides			Y	Always 2 Sides
→	54	Side		Y	Side. Valid values: 1 = Buy 2 = Sell
→	37	OrderID		Y	Required in FIX. Set to “NONE”.
→	453	NoPartyIDs		Q	Number of party id entries
→	→	448	PartyID	Q	party identifier
→	→	447	PartyIDSource	Q	Valid values : D = Propr. Code
→	→	452	PartyRole	Q	Valid values: 1 = Executing Firm 4 = Clearing Firm 14 = Giveup Clearing Firm (Takeup Firm) 17 = Contra Firm 83 = Clearing Account 1001 = Confirmed by Firm (NASDAQ OMX Extension)
→	528	OrderCapacity			*NASDAQ OMX Extension:* Designates the capacity of the firm placing the order. Valid values: P = Principal A = Agency R = Riskless Principal **NOTE:** Required for Fixed Income. **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	529	OrderRestrictions			*NASDAQ OMX Extension:* Restrictions associated with an order. Valid values: B = Issuer Holding **(requires 528=A)** C = Issue Price Stabilization **(requires 528=P)** 5 = Acting as Market Maker or Specialist in the security **(requires 528=P)** **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	483	TransBkdTime			*NASDAQ OMX Extension:* Time of agreement. **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	20009	OrderReference			*NASDAQ OMX Extension:* Order Reference

				pass-thru field. **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	1	Account		Optional pass-thru field set by client and echoed back by marketplace. **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)

→	77	PositionEffect		Defines the position update for the account..
				Valid values:
				C = Close
				O = Open
				NOTE: for instruments **not** cleared within the system, this field contains the *requested* position effect. The following additional values then applies:D = Default
				M = Mandatory Close (NASDAQ OMX Extension)

→	70	AllocID		Optional pass-thru field set by client and echoed back by marketplace. **NOTE:** Only set on the own Side (where PartyRole=Executing Firm)
→	151	LeavesQty		Remaining quantity
715	ClearingBusinessDate			
855	SecondaryTrdType			Contains Genium INET deal_source value.
793	SecondaryAllocID			*NASDAQ OMX Extension:* Contains Genium INET Give_up_number.
20013	DeferredPublicationTime			*NASDAQ OMX Extension:* The number of minutes the publication of this trade will be delayed (relative to time of agreement). **NOTE:** -1 means end of day.
21000	DealID			*NASDAQ OMX Extension:* Contains the numeric Genium INET deal_number.
797	CopyMsgIndicator			Set to 'Y' on Drop Copy messages
	Standard Trailer		Y	

13.4.4 Trade Capture Report – Confirmed Trade Cancel (in)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = AE
571	TradeReportID	Y	Client-generated identifier
1003	TradeID	Q	NASDAQ OMX Extension: Unique identifier for trade
487	TradeReportTransType	Q	Valid values: 1 = Cancel
856	TradeReportType	Q	Valid values: 0 = Submit
570	PreviouslyReported	Y	Indicates if the trade capture report was previously reported to the counterparty Valid values: N = No

1116	NoRootPartyIDs			Number of party id entries (used for on-behalf-of transactions)
→	1117	RootPartyID	Q	Party identifier.
→	1118	RootPartyIDSource	Q	Valid values: D = Proprietary/Custom code
→	1119	RootPartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 1 = Executing Firm 12 = Executing Trader
55	Instrument/Symbol			OMNet short name. Either Symbol or SecurityID+SecurityIDSource needs to be set.
48	Instrument/SecurityID			Orderbook ID
22	Instrument/SecurityIDSource			Valid values: M = Marketplace-assigned identifier
32	LastQty		Y	Not validated
31	LastPx		Y	Not validated
75	TradeDate		Y	Trade Date. Must be set to a valid date. Required in FIX but ignored.
60	TransactTime		Y	
552	NoSides		Y	Set to 1, only own side given
→	54	Side	Y	Own Side. Valid values: 1 = Buy 2 = Sell
→	37	OrderID	Y	Required in FIX, but ignored
	Standard Trailer		Y	

13.4.5 Trade Capture Report Ack – Confirmed Trade Cancel (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = AR
571	TradeReportID	Y	The client-generated identifier
487	TradeReportTransType	Q	Valid values: 1 = Cancel
1003	TradeID	Q	NASDAQ OMX Extension: Unique identifier for trade
150	ExecType	Y	Type of Execution being reported. Valid values: 4 = Canceled
939	TradeRptStatus	Q	Valid values: 0 = Accepted
55	Instrument/Symbol	Q	Short name of security
48	Instrument/SecurityID	Q	
22	Instrument/SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
	Standard Trailer	Y	

13.4.6 Trade Capture Report Ack – Confirmed Trade Cancel Reject (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = AR
571	TradeReportID	Y	The client-generated identifier

487	TradeReportTransType	Q	Valid values: 1 = Cancel
150	ExecType	Y	Type of Execution being reported. Valid values: 8 = Rejected
939	TradeRptStatus	Q	Valid values: 1 = Rejected
751	TradeReportRejectReason	Q	Valid values: 99 = Other
55	Instrument/Symbol	Y	**NOTE:** Set to [N/A]
58	Text		Free text description of reject
	Standard Trailer	Y	

13.4.7 Trade Capture Report – Rectify Confirmed Trade (in)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = AE
571	TradeReportID	Y	Client-generated identifier
1003	TradeID	Q	NASDAQ OMX Extension: Unique identifier for trade. **NOTE:** The first part is ignored by the system, so a TradeID of 0:x, where x is the trade number, can be entered.
487	TradeReportTransType	Q	Valid values: 2 = Replace
856	TradeReportType	Q	Valid values: 0 = Submit
570	PreviouslyReported	Y	Indicates if the trade capture report was previously reported to the counterparty Valid values: N = No
55	Instrument/Symbol		OMNet short name. Either Symbol or SecurityID+SecurityIDSource needs to be set.
48	Instrument/SecurityID		Orderbook ID
22	Instrument/SecurityIDSource		Valid values: M = Marketplace-assigned identifier
32	LastQty	Y	Not validated
31	LastPx	Y	Not validated
75	TradeDate	Y	Trade Date. Must be set to a valid date. Required in FIX but ignored.
60	TransactTime	Y	
552	NoSides	Y	Set to 1, only own side given
→ 54	Side	Y	Own Side. Valid values: 1 = Buy 2 = Sell
→ 37	OrderID	Y	Required in FIX, but ignored
→ 77	*PositionEffect*		**NOTE: This option is currently unavailable. Setting tag 77 will cause a reject. The same result can be achieved**

						by using the AllocPositionEffect (tag 1047). *Can only be used to close the position. In this case NoAllocs must be 1, and the other fields must be identical to that of the trade to be rectified. Valid values:* *C = Close*
→	826	TradeAllocIndicator				Identifies how the trade is to be allocated. Valid values: 6 = Trade Posting
→	78	NoAllocs			Q	Can be set to more than 1 if the trade is to be split into multiple accounts.
→	→	79	AllocAccount		Y	Account the trade should be posted to. **NOTE:** This field is required in FIX. Must be set to “NONE” if no value is desired.
→	→	756	NoNested2PartyIDs		Q	Will always be set to 1.
→	→	→	757	Nested2PartyID		Clearing firm id (owner of account given in tag 79).
→	→	→	758	Nested2PartyIDSource		Valid values: D = Propr. Code
→	→	→	759	Nested2PartyRole		Valid values: 4 = Clearing Firm
→	→	467	IndividualAllocID			Free text field.
→	→	80	AllocQty			Quantity allocated to the current AllocAccount.
→	→	1047	AllocPositionEffect			NASDAQ OMX Extension. If this field is not set the default action for the account will occur. Valid values: O = Open C = Close
	Standard Trailer				Y	

13.4.8 Trade Capture Report Ack – Rectify Confirmed Trade (out)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = AR
571	TradeReportID	Y	The client-generated identifier
487	TradeReportTransType	Q	Valid values: 2 = Replace
1003	TradeID	Q	NASDAQ OMX Extension: Unique identifier for trade.
150	ExecType	Y	Type of Execution being reported. Valid values: G = Trade Correct
939	TradeRptStatus	Q	Valid values: 0 = Accepted
55	Instrument/Symbol	Q	Short name of security
48	Instrument/SecurityID	Q	
22	Instrument/SecurityIDSource	Q	Valid values:

			M = Marketplace-assigned identifier
797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
	Standard Trailer	Y	

13.4.9 Trade Capture Report Ack – Rectify Confirmed Trade Reject (out)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = AR
571	TradeReportID	Y	The client-generated identifier
487	TradeReportTransType	Q	Valid values: 2 = Replace
150	ExecType	Y	Type of Execution being reported. Valid values: 8 = Rejected
939	TradeRptStatus	Q	Valid values: 1 = Rejected
751	TradeReportRejectReason	Q	Valid values: 99 = Other
55	Instrument/Symbol	Y	**NOTE:** Set to [N/A]
58	Text		Free text description of reject
	Standard Trailer	Y	

13.4.10 Allocation Instruction – Give up Request (in)

Tag	FIX tag name	Req’d	Comment
	Standard Header	Y	MsgType = J
70	AllocID	Y	Client-generated identifier
71	AllocTransType	Q	Valid values: 0 = New
626	AllocType	Q	Valid values: 17 = Give-Up
124	NoExecs	Q	1
32	LastQty	Q	Required in FIX, ignored by the system.
1003	TradeID	Q	NASDAQ OMX Extension: Unique identifier for trade
54	Side	Y	Required in FIX. Ignored
55	Instrument/Symbol		OMNet short name. Either Symbol or SecurityID+SecurityIDSource needs to be set.
48	Instrument/SecurityID		Orderbook ID
22	Instrument/SecurityIDSource		Valid values: M = Marketplace-assigned identifier
53	Quantity	Y	Required in FIX but ignored.
75	TradeDate	Y	Trade Date. Must be set to a valid date. Required in FIX but ignored.
78	NoAllocs	Y	Set to 1
→ 79	AllocAccount	Y	Optional account (of the participant the trade is given up to). **NOTE:** This field is required in FIX. Must be set to “NONE” if no value is desired.
80	AllocQty	Q	Quantity of the trade to be given up. This does not have to be the full trade quantity.

→	539	NoNestedPartyIDs		Q	Must be set to 1.
→	→	524	NestedPartyID	Q	ID of the participant the trade should be given up to.
→	→	525	NestedPartyIDSource	Q	Valid values: D = Propr. Code
→	→	538	NestedPartyRole	Q	Valid values: 14 = Giveup Clearing Firm (firm to which the trade is given up)
→	12	Commission			
→	13	CommType			Valid values: 3 = Absolute
58	Text				Contains user supplied text as information to the receiver. Max 30 chars.
	Standard Trailer			Y	

13.4.11 Allocation Report Ack – Reject (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = AT
755	AllocReportID	Y	Unique identifier for this message
70	AllocID	Y	The AllocID set by the requestor.
71	AllocTransType	Q	Valid values: 0 = New
87	AllocStatus	Q	Valid values: 1 = Block level reject
88	AllocRejCode	Q	Valid values: 99 = Other
58	Text		Free text description of reject
	Standard Trailer	Y	

13.4.12 Allocation Report – Give up Notification (out)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = AS
755	AllocReportID		Unique identifier for this message
70	AllocID		To the give up requestor this will be set to the AllocID sent in. Will not be set for other recipients.
793	SecondaryAllocID	Q	Give_up_number.
71	AllocTransType	Q	Valid values: 0 = New
794	AllocReportType	Q	Valid values: 15 = Give-Up 16 = Take-Up
87	AllocStatus	Y	Identifies the status off allocation. Valid values: 6 = allocation pending 7 = reversed

					9 = claimed 10 = refused 14 = reversal pending
715	ClearingBusinessDate				
54	Side			Q	Valid values: 1 = Buy 2 = Sell
124	NoExecs			Q	1
→	32	LastQty		Y	Required, not used.
→	31	LastPx			Deal price
→	1003	TradeID		Q	NASDAQ OMX Extension: Unique identifier for trade
55	Instrument/Symbol				OMNet short name. Either Symbol or SecurityID+SecurityIDSource needs to be set.
48	Instrument/SecurityID				Orderbook ID
22	Instrument/SecurityIDSource				Valid values: M = Marketplace-assigned identifier
53	Quantity			Y	Required in FIX but ignored.
30	LastMkt				Defines the trade venue
6	AvgPx			Y	Set to 0
453	NoPartyIDs			Q	Set to 1
→	448	PartyID		Q	Participant ID for firm giving up the trade
→	447	PartyIDSource		Q	Valid values: D = Proprietary
→	452	PartyRole		Q	Valid values: 1 = Executing Firm
75	TradeDate			Y	Trade Date. Must be set to a valid date. Required in FIX but ignored.
60	TransactTime				
78	NoAllocs			Y	Set to 1
→	79	AllocAccount			**NOTE:** This field is required in FIX. Will be set to "NONE" if no value exists.
→	80	AllocQty		Q	Quantity to be given up. Need not be full
→	539	NoNestedPartyIDs		Q	Will be set to 1
→	→	524	NestedPartyID	Y	Participant ID for the take up firm
→	→	525	NestedPartyIDSource	Q	Valid values: D = Proprietary
→	→	538	NestedPartyRole	Q	Valid values: 14 = Giveup Clearing Firm (firm to which the trade is given up)
→	12	Commission			
→	13	CommType			Valid values: 3 = Absolute
→	20014	ExternalTradeFeeType			The name of the trade fee type used to calculate the trade fee.

58	Text		Contains user supplied text as information to the receiver. Max 30 chars.
1040	SecondaryTradeID		NASDAQ OMX Extension: External trade number
1127	OrigSecondaryTradeID		NASDAQ OMX Extension: External trade number
855	SecondaryTrdType		NASDAQ OMX Extension: Contains Genium INET deal_source value.
	Standard Trailer	Y	

13.4.13 Allocation Instruction – Accept Give up Request (in)

Tag	FIX tag name			Req'd	Comment
	Standard Header			Y	MsgType = J
70	AllocID			Y	ID from Allocation Report – Give Up Notification message.
71	AllocTransType			Q	Valid values: 0 = New
626	AllocType			Q	Valid values: 18 = Take-Up
54	Side			Y	Required in FIX. Ignored
55	Instrument/Symbol				OMNet short name. Either Symbol or SecurityID+SecurityIDSource needs to be set.
48	Instrument/SecurityID				Orderbook ID
22	Instrument/SecurityIDSource				Valid values: M = Marketplace-assigned identifier
53	Quantity			Y	Required in FIX but ignored.
75	TradeDate			Y	Trade Date. Must be set to a valid date. Required in FIX but ignored.
78	NoAllocs			Y	Can be multiple if take up is to be distributed over several accounts
→	79	AllocAccount		Y	**NOTE:** This field is required in FIX. Must be set to "NONE" if no value is desired.
→	80	AllocQty		Q	
→	1047	AllocPositionEffect			
→	539	NoNestedPartyIDs		Q	Set to 1
→	→	524	NestedPartyID	Q	Identifier of the owner of the AllocAccount
→	→	525	NestedPartyIDSource	Q	Valid values: D = Proprietary
→	→	538	NestedPartyRole	Q	
→	161	AllocText			Contains customer_info
	Standard Trailer			Y	

13.4.14 Allocation Instruction – Reject Give up Request (in)

Tag	FIX tag name	Req'd	Comment
	Standard Header	Y	MsgType = J
70	AllocID	Y	ID from Allocation Report – Give Up Notification

			message.
71	AllocTransType	Q	Valid values: 0 = New
626	AllocType	Q	Valid values: 19 = Refuse Take-Up
54	Side	Y	Required in FIX. Ignored
55	Instrument/Symbol		OMNet short name. Either Symbol or SecurityID+SecurityIDSource needs to be set.
48	Instrument/SecurityID		Orderbook ID
22	Instrument/SecurityIDSource		Valid values: M = Marketplace-assigned identifier
53	Quantity	Y	Required in FIX but ignored.
75	TradeDate	Y	Trade Date. Must be set to a valid date. Required in FIX but ignored.
58	Text		Contains user supplied text as information to the receiver. Max 30 chars.
	Standard Trailer	Y	

Appendix A, NASDAQ OMX Extensions

This chapter details how this solution deviates from standard FIX 4.4. While great care has been taken to conform to the standard, a number of deviations are unavoidable to support all mechanisms provided by the host. Wherever later versions of FIX (up to version 5.0 SP2) provide the missing functionality, we have chosen to use that.

There are four types of deviations from the standard:

- Messages added. All current additions come from the later standard versions of FIX.
- Fields added. Most added fields come from later FIX versions. A few user defined fields had to be added to accommodate back-end functionality not present in FIX 4.4.
- Enumerated values added. Some fields have added enums.
- Removed fields required in standard FIX.
- Other datatype used for existing field.

Added Messages

The following messages not present in standard FIX 4.4 have been added to this specification:

Message Name	In FIX 5.0 SP2	Comment
User Notification	Y	MsgType = CB
One Sided Auction Request	N	MsgType = UB
One Sided Cancel Auction Request	N	MsgType = UC
One-Sided Auction Request Ack	N	MsgType = UD

Added Fields

Tag Num	Field Name	In FIX 5.0 SP2	Comment
1003	TradeID	Y	
1126	OrigTradeID	Y	
1089	MatchIncrement	Y	
70	AllocID	Y	Existing FIX tags added to the Order Cancel Request and the Execution Report messages.
1385	ContingencyType	Y	
1390	TradePublishIndicator	Y	
925	NewPassword	Y	Added to inbound Logon
926	UserStatus	Y	User Notification
1100	TriggerType	Y	
1101	TriggerAction	Y	
1102	TriggerPrice	Y	
1113	TriggerTradingSessionID	Y	
1103	TriggerSymbol	Y	
1104	TriggerSecurityID	Y	
1105	TriggerSecurityIDSource	Y	
1107	TriggerPriceType	Y	
1109	TriggerPriceDirection	Y	
1116	NoRootPartyIDs	Y	

1117	RootPartyID	Y	
1118	RootPartyIDSource	Y	
1119	RootPartyRole	Y	
1386	ListRejectReason	Y	
1418	LegLastQty	Y	For multileg fills.
20001	AuctionRequestID	N	For One-sided auctions. Data type: String
20002	AuctionType	N	For One-sided auctions. Data type: char
20003	BookTransparency	N	For One-sided auctions. Data type: char
20004	AuctionUncrossTime	N	For One-sided auctions. Data type: UTCTimestamp
20005	AuctionRequestResult	N	For One-sided auctions. Data type: char
20006	CleanPrice	N	Data type: Price
20007	CorrespondingPrice	N	Data type: Price
20008	Consideration	N	Data type: Price
20009	OrderReference	N	Data type: String
20010	QuantityLimit	N	For One-sided auctions. Data type: Qty
20011	ReferencePrice	N	For One-sided auctions. Data type: Price
20012	OrigAuctionRequestID	N	For One-sided auctions. Data type: String
20013	DeferredPublicationTime	N	Data type: int
20014	ExternalTradeFeeType	N	Data type: String
21000	DealID	N	Data type: int
21013	TradeReportState	N	Data type: int
21014	TradeReportSubState	N	Data type: int
21015	TradeReportInstrType	N	Data type: int
21016	TraderReportReason	N	Data type: int
21017	AuthorizationState	N	Data type: int
21018	AffimationState	N	Data type: int
21019	DeliveryUnit	N	Data type: int
21020	OrigClearingBusinessDate	N	Data type: UTC Date only
21021	StrategyMarker	N	Data type: int
528	OrderCapacity	Y	Existing field added to Trade Capture Reports.
529	OrderRestrictions	Y	Existing field added to Trade Capture Reports.
483	TransBkdTime	Y	Existing field added to Trade Capture Reports.
793	SecondaryAllocID	Y	Existing field added to Trade Capture Reports.

Added Enumerations

Enumeration	Added to Field	In FIX 5.0 SP2	Comment
1001 = Standard Trade, Outside Spread 1002 = Standard Trade, on hours (fixed income) 1003 = Exchange Granted Trade, on hours (fixed income) 1004 = OTC non-standard Trade, on hours (fixed income) 1005 = Turnover Reporting (fixed income)	TrdType	N	

1006 = Exchange Granted Trade, exceeding Maximum Lot Size, Off Hours 1007 = Off Hours Trade 1008 = Block Trade 2103 = Exchange Granted Trade, Late Reported 2105 = Exchange Granted Trade, exceeding Maximum Lot Size 1102 = Standard Trade, off hours (fixed income) 1103 = Exchange Granted Trade, off hours (fixed income) 1104 = OTC non-standard Trade, off hours (fixed income) 1148 = Derivatives Related Trade, off hours (fixed income) 1149 = Non-Standard Settlement, off hours (fixed income) 1150 = Portfolio Trade , off hours (fixed income) 1151= Volume Weighted Average Price Trade, off hours (fixed income) 1153 = Repurchase Agreement, off hours (fixed income) 1154 = OTC Standard Trade, off hours (fixed income) 1201 = Standard (Commodities) 1202 = Standard Outside Spread (Commodities) 1203 = Combination (Commodities) 1204 = Old (Commodities) 1205 = Internal (Commodities) 1206 = Portfolio (Commodities) 1207 = Correction (Commodities)			
1001 = Standard 1002 = Transitory 1003 = Overtaking 1004 = Reversing 1005 = Transfer. 1008 = Closing. 1009 = Issue 1010 = New contract. 1011 = Delivery 1012 = Dummy trade 1013 = Alias 1014 = Offsetting 1015 = Superseeding	TrdSubType	N	

1016 = State change 1017 = Giveup 1018 = Takeup			
M = Marketplace-assigned identifier	SecurityIDSource	Y	
101 = Genium INET series definition (NASDAQ OMX Extension)	SecurityIDSource	N	
101 = Password Expired	UserStatus	N	
102 = New password does not comply with policy	UserStatus	N	
100 = Invalid body length in received message, session suspended 101 = Heartbeat interval too low.	SessionStatus	N	
L = Triggered or Activated by the system	ExecType	Y	
S = GTS	TimeInForce	N	
1001 = Confirmed by Firm 1002 = Confirmed by User 1003 Reported by Firm 1004 Reported by User 1005 = Affirmed by Firm 1006 = Affirmed by User 1007 = Give-up Account	PartyRole	N	

Removed required fields

Tag Num	Field Name	In Message	Comment
54	Side	Execution Report – Order Reject	

Appendix B, Field length limitations

The following fields have a max length limit:

Tag Num	Field Name	max length	Comment
11	ClOrdID	20	
41	OrigClOrdID	20	
117	QuoteID	20	
66	ListID	20	
320	SecurityRequestID	20	
70	AllocID	15	Existing FIX tags added to the Order Cancel Request and the Execution Report messages.
1	Account	10	
79	AllocAccount	10	
448	PartyID	4 or 7	When PartyRole=Clearing Firm
448	PartyID	12 or 10	When PartyRole=Clearing Account
448	PartyID	7	When PartyRole=Executing Firm or Contra Firm
524	NestedPartyID	7	

757	Nested2PartyID	7	
571	TradeReportID	20	
572	TradeReportRefID	20	
881	SecondaryTradeReportRefID	20	
923	UserRequestID	20	
925	NewPassword	32	
20001	AuctionRequestID	20	
20012	OrigAuctionRequestID	20	
20009	OrderReference	10	

Appendix C, Trade types

The following table contains the definitions of all values the TrdType (828) field can contain.

Value	Name	Description	Asset Class
0	ST	Standard Trade	Financial derivatives
52	EGT	Exchange Granted Trade	Financial derivatives
1001	STOS	Standard trade, Outside Spread	Financial derivatives
1007	OHT	Off Hours Trade	Financial derivatives
1008	BT	Block Trade	Financial derivatives
2105	BTX	Exchange Granted Trade, exceeding Maximum Lot Size	Financial derivatives
1006	BTXO	Exchange Granted Trade, exceeding Maximum Lot Size, Off Hours	Financial derivatives
2103	EGLT	Exchange Granted Trade, Late Reported	Financial derivatives
48		Non-Standard Settlement, on hours	Fixed Income
49		Derivatives Related Trade, on hours	Fixed Income
50		Portfolio Trade, on hours	Fixed Income
51		Volume Weighted Average Price Trade, on hours	Fixed Income
53		Repurchase Agreement, on hours	Fixed Income
54		OTC Standard Trade, on hours	Fixed Income
1002		Standard Trade, on hours	Fixed Income
1003		Exchange Granted Trade, on hours	Fixed Income
1004		OTC non-standard Trade, on hours	Fixed Income
1005		Turnover Reporting	Fixed Income
1102		Standard Trade, off hours	Fixed Income
1103		Exchange Granted Trade, off hours	Fixed Income
1104		OTC non-standard Trade, off hours	Fixed Income
1148		Derivatives Related Trade, off hours	Fixed Income
1149		Non-Standard Settlement, off hours	Fixed Income
1150		Portfolio Trade , off hours	Fixed Income
1151		Volume Weighted Average Price Trade, off hours	Fixed Income

1153		Repurchase Agreement, off hours	Fixed Income
1154		OTC Standard Trade, off hours	Fixed Income
1201		Standard	Commodity Derivatives
1202		Standard Outside Spread	Commodity Derivatives
1203		Combination	Commodity Derivatives
1204		Old	Commodity Derivatives
1205		Internal	Commodity Derivatives
1206		Portfolio	Commodity Derivatives
1207		Correction	Commodity Derivatives
1401	F01	Block - Standard	Commodities - Freight
1403	F03	Block - Combination	Commodities - Freight
1404	F04	EFS/EFP	Commodities - Freight
1405	F05	Block - Internal	Commodities - Freight
1406	F06	Portfolio	Commodities - Freight
1407	F07	Block - Correction	Commodities - Freight
1416	F16	Block Combination, buyer only	Commodities - Freight
1417	F17	Block Combination, seller only	Commodities - Freight
1418	F18	EFS/EFP Combination	Commodities - Freight
1419	F19	EFS/EFP Combination, buyer only	Commodities - Freight
1420	F20	EFS/EFP Combination, seller only	Commodities - Freight
1431	IO01	OTC, Standard Trade report	Commodities - Iron Ore
1432	IO03	OTC, Combination	Commodities - Iron Ore
1433	IO16	OTC, Combination buyer only	Commodities - Iron Ore
1434	IO17	OTC, Combination seller only	Commodities - Iron Ore
1421	SF01	Standard Trade Report, Fishpool	Commodities - Seafood
1422	SF02	Standard Trade Report, Broker	Commodities - Seafood
1423	SF03	Combination, Fishpool	Commodities - Seafood
1424	SF15	Combination, Broker	Commodities - Seafood
1425	SF16	Combination, FP buyer only	Commodities - Seafood
1426	SF17	Combination, FP seller only	Commodities - Seafood
1427	SF18	Combination, Broker buyer only	Commodities - Seafood
1428	SF19	Combination, Broker seller only	Commodities - Seafood

Revision History

Date	Revision	Change Description
January 13, 2011	1.00	Public release. Modified TradeReportTransType values for Notification Modify/Delete message to Cancel and Replace. Clarified that TradeDate must be set to a valid date. Added values 1001-1004 to TrdSubType. Now shows all omnet trade_type_c values (+1000). Clarified that tags 1 and 70 can be set on both sides of a two-party trade report. Removed PartyRole=Executing Trader on the two-party confirmation to counterparty.
January 21, 2011	1.01	Moved Time of Agreement (tag 483) within the Trade Capture Report messages to be compatible with FIX 5.0. It is now located within the NoSides repeating group. Renamed the Modify/Delete Notification message since modifications of trade reports are not supported.
March 31, 2011	1.02	CLARIFICATION: Removed Replace value for tag 487 of Delete Notification since replaces cannot occur. CLARIFICATION: Added note in chapter 13.1.3 that delete notifications will not contain TargetSubID. CLARIFICATION: Removed reference to trade report updates in chapter 11.3.1.5 as they cannot occur (only deletes). CLARIFICATION: Clarified that cancelling a trade report will cause a delete notification to be sent to the counterparty. CLARIFICATION: Clarified how TradeID is constructed. CLARIFICATION: Swapped the sides on the resulting TCR in example 13.3.4. Added missing TrdMatchID (880) to two-party confirmation to counterparty and Multileg Order fill messages. Removed NoTradingSessions (386) tag from Execution Reports since it was non-standard and doesn’t carry any information. *This affects trigger orders only.* Added values *Filled*, and *Canceled* to OrdStatus for Execution Report – Cancel Replace Ack. These

Date	Revision	Change Description
		values are needed when a Cancel Replace causes the open quantity to go to 0 (zero). See std FIX scenario C.3.b.
May 3, 2011	1.03	Added two new fixed income TrdTypes: • 1104 = OTC non-standard Trade, off hours • 1154 = OTC Standard Trade, off hours CLARIFICAT ION: Clarified that TradeID is encoded as a hex string. CLARIFICAT ION: Clarified that the Mass Quote Ack Text field can contain multiple error messages separated by a "#". CLARIFICAT ION: Clarified that on the Logon or User Request, the SenderSubID must be set to the user id the client intends to log on. CLARIFICAT ION: Clarified the limit on the number of indicative quotes that can be entered in a single Mass Quote transaction. The *Security Definition – TMC Registration Response* message has been simplified and some fields removed.
May 9, 2011	1.04	CLARIFICATION: Clarified that TrdTypes 54 and 1004 are *on hours* trade types. Removed appendix on non-standard data types. CLARIFICATION: Clarified that the AllocID field may only be overwritten on outbound TCRs in rare cases due to manual intervention by the marketplace.
May 27, 2011	1.05	CLARIFICATION: Clarified note on modifying the price of an order to zero. CLARIFICATION: Removed NoMarketSegments and MarketID fields from the list of added fields, since they are not present in the spec anymore.
June 9, 2011	1.10	Added initial support for commodities trading.
June 30, 2011	1.11	Replaced field MiscFeeType with ExternalTradeFeeType in Giveup Notification message. Added new TrdTypes for Commodities.
July 7, 2011	1.12	Added fields to support on-behalf-of indicative quotes. Added Quote Status Report message for indicative quote acknowledgements. Added FPL proposed values for AllocStatus Added FPL proposed values for AllocReportType Added FPL proposed values for AllocType
August 18, 2011	1.13	Added field descriptions to the rectify trade message. Added SecondaryAllocID to Giveup Notification message.

Date	Revision	Change Description
		Added that the deal number part of TradeID may be set to 0 in some cases. Text field removed from the Accept Give up Request message.
August 31, 2011	1.14	Added possibility to set Giveup Firm on the inbound trade reports.
September 26, 2011	1.15	Added note that AllocAccount must be set to NONE if no value exists. Added Allocation Report Ack – Reject message. Added REPO related instrument registration messages. Added SecondaryAllocID to trade confirmation messages. Clarified that IOC orders can be of other Lot Types (MatchIncrement) than Round Lot. Clarified that AON orders cannot have MatchIncrement set. Replaced Quote Status Report with Mass Quote Ack. Rewrote the description of Mass Quote Acks. Added OBO fields to the Mass Quote Acks. Clarified that DeferredPublicationTime = -1 means until end of day. Fixed tag number for LastQty Added Order book id and removed SecurityResponseType to Sec Def Update.
October 4, 2011	1.16	Clarified how QuantityLimit and ReferencePrice can be used. Clarified that the leg size / leg lot size ratio must be the same across all legs in a linked order. Added examples of TMC and REPO registrations.
October 19, 2011	1.17	Clarified use of the rectify trade message. Removed old Revision History items.
February 29, 2012	1.18	CLARIFICATION: Clarified that Username must be in capital letters. CLARIFICATION: Removed draft document status. CLARIFICATION: Fixed text formatting issue in section 12.1.4. CLARIFICATION: Added Commodities market code (NC) to market code list. CLARIFICATION: Clarified how the Pass-thru fields can be used. Added section to detail how clearing accounts are handled. CLARIFICATION: Changed the description of Countersign Firm to Confirmed by Firm. CLARIFICATION: Removed reference to missing failover document.

Date	Revision	Change Description
		CLARIFICATION: Clarified that for Commodities TradeID does not contain deal_number which makes it unique regardless of instrument. CLARIFICATION: Clarified that suspended/inactivated orders cannot be reactivated. CLARIFICATION: Added section on Clearing Accounts CLARIFICATION: Clarified that the option to use the Rectify Trade message close a position by setting tag 77 to C is unavailable. CLARIFICATION: Clarified that OrderQty is ignored on Order Cancel Requests. CLARIFICATION: Added field length limitations for NestedPartyID and Nested2PartyID CLARIFICATION: Removed note that said Nested2PartyID must contain own participant id. CLARIFICATION: Clarified section 5.1.1.1 that PartyID can contain other identifiers than participant ids.
September 4, 2012	1.19	Minor clarification on the contents of Nested2PartyIDs. Added DealID field to trade confirmations. Clarified that MaxFloor on Execution Reports will now show the *currently* visible quantity. Added leg information to the Security Definition Update Report. Added LastQty to Combination Order Fill. Clarified that the option to trigger on best bid or best offer is not available. Fixed the numbering of TriggerTradingSessionID enumerations. Fixed broken cross references.
September 18, 2012	1.20	Added new Trade types and moved them to a separate appendix.
October 25, 2012	1.21	Rewrote description of the trigger order workflow. Added PositionEffect values for non-cleared instruments.
January 16, 2013	1.22	Added SettlPrice to One-Sided Auction Request. CORRECTION: Added missing Executing Trader value in PartyRole field of trade confirmations (the value is already present in these messages). CORRECTION: Added SettlDate to auto-matched trade confirmation (the field is already present in these messages). CORRECTION: Removed the required flag on AllocID in Allocation Report messages.

Date	Revision	Change Description
March 4, 2013	1.23	Introduce the Commodities-format of the TradeID field across all markets. Clarified the limitations of how the AON flag can be used.
May 21, 2013	1.24	CLARIFICATION: Removed obsolete note on DealID. CORRECTION: Corrected the descriptions of TrdTypes 1148 and 1149.
June 13, 2013	1.30	Initial addition of OTC Trade Reporting. Added new custom fields and enum values.
June 28, 2013	1.31	Clarified how OTC Trade identifiers are formatted.
October 10, 2013	1.32	Added new Commodities trade types.
February 20, 2014	1.33	Added new text regarding Give-Ups, removed limitations of OrderReference field. Added more TraderReportReasons(21016). Removed unused TradeReportState(21013)
February 25, 2014	1.35	Added additional comments on give ups