NASDAQ Futures, Inc. FIX

Version 1.31 | 2017-11-13

Confidentiality/Disclaimer

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

1 CONTENTS

2 REFERENCES

[1]
FIX 5.0 SP2 Protocol Specification
http://fixprotocol.org/specifications/fix5.0sp2spec

[2]
RFC 2045, Base64 encoding for MIME
http://www.ietf.org/rfc/rfc2045.txt

3 OVERVIEW

This document contains the specification for the FIX interface to NASDAQ Futures, Inc. (NFX). The interface is based on the FIX Protocol 5.0 SP2 standard (Financial Information exchange). More detailed information about the standard can be found in FIX specification document see [1].

The interface implemented by NFX 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 NFX has found reason to clarify matters. NFX tries to be explicit on deviations from the FIX standard specification in order to avoid confusion.

3.1 SUPPORTED MESSAGES

3.1.1 Administrative messages

Logon
Logout
Sequence Reset
Resend Request
Reject
Heartbeat
Test Request

3.1.2 Inbound Application messages

User Request
New Order Single
Order Cancel Replace Request
Order Cancel Request
Mass Quote
Quote Request
Trade Capture Report
Multileg Trade Report (custom message)
Security Definition Request
New Order List
MMProtection Set Limit

3.1.3 Outbound Application messages

User Response
User Notification
Execution Report
Order Cancel Reject
Business Message Reject
Mass Quote Acknowledgement
Quote Request Reject
Quote
Trade Capture Report
Trade Capture Report Ack
Security Definition (TMC ack/reject)

3.2 NFX EXTENSIONS

In order to support specific functionality of the back-end not covered by the FIX 5.0 SP2 standard protocol, a number of extensions have been made. A few custom messages and fields have been added.

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

Custom fields added by NFX have a tag number higher than 20000.

4 THE FIX SESSION

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

4.1 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 “NFX” for production and “NFX_TEST” for test systems.
- The SenderCompID for transactions sent *outbound* from the Exchange will be “NFX” for production and “NFX_TEST” for test systems.

4.2 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 5 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.

4.3 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.

4.3.1 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.

4.3.2 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.

4.4 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.

4.5 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.

When either end of a FIX connection has not sent any data for HeartBtInt seconds, it will transmit a Heartbeat message. When either end of the connection has not received any data for (HeartBtInt + “some reasonable transmission time”) seconds, it will transmit a Test Request message. If there is still no Heartbeat message received after (HeartBtInt + “some reasonable transmission time”) seconds then the connection should be considered lost.

4.6 ENCRYPTION

The system does not support encryption.

4.7 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.

4.8 CHARACTER ENCODING

The FIX gateway uses standard US ASCII encoding.

4.9 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 at the start of each new trading day..

4.10 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.

4.10.1 Order Suspension 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.

4.11 FIX SESSION LEVEL TEST CASES

This implementation is fully compliant with the session-level test cases specified in the standard FIX 5.0 SP2 Specification, Volume 2, section “FIX Session-level Test Cases and Expected Behaviors”. The only exception is the encryption test cases.

4.12 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

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".

4.12.1 Drop Party Identifiers

The messages on a FIX Drop session will have the same TargetCompID (participant) and TargetSubID (trader) as the messages on the original session it copies. The messages will be copies of the original messages but with the CopyMsgIndicator flag set.

4.13 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: FIXT.1.1
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 be set to the user that own the order.
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.

4.13.1 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.

4.14 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	

4.15 MESSAGE DETAILS

4.15.1 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.

4.15.2 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*.

4.15.3 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 **NOTE:** Must be in CAPTIAL LETTERS.
554	Password	Q	password (unencrypted)
925	NewPassword		Specifies a new password for the FIX Logon. The new password is used for subsequent logons.
1137	DefaultApplVerID	Y	The default version of FIX messages used in this session. Valid values: 9 = FIX50SP2
	Standard Trailer	Y	

4.15.4 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	Status of the FIX session. Valid values: 0 = Session Active 1 = Session password changed 3 = New session password does not comply with policy
1137	DefaultApplVerID	Y	The default version of FIX messages used in this session. Valid values: 9 = FIX50SP2
	Standard Trailer	Y	

4.15.5 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 = *NFX Extension:* Invalid body length in received message, session suspended 101 = *NFX Extension:* Heartbeat interval too low.
58	Text		Free text
	Standard Trailer	Y	

4.15.6 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	

4.15.7 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	

4.15.8 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	

4.15.9 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	

4.15.10 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	

5 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.

5.1 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.

5.2 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.

5.3 USER NOTIFICATION

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

5.4 PASSWORD MANAGEMENT

5.4.1 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.

5.4.2 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.

5.5 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, this second connection will be logged out. A fix session will not accept any new connections while a user is connected to that session.

5.6 MESSAGE DETAILS

5.6.1 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	

5.6.2 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 (NFX Extension) 102 = New Password does not comply with policy (NFX Extension)
927	UserStatusText		A text description associated with a user status.
	Standard Trailer	Y	

5.6.3 User Notification (out)

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	

6 BUSINESS LEVEL PARTY IDENTIFIERS

6.1 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.

6.1.1 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) = actual identifier
- PartyIDSource (447)
  - D = Proprietary/Custom code
- PartyRole (452) = see below

6.1.1.1 Participant Identifier

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

- NF ??

NOTE: This identifier scheme is also used for SenderCompID.

6.1.2 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:

- Trade Capture Report

6.2 IMPLICIT PARTIES

All inbound business messages must contain:

- SenderCompID (49) = participant identifier of the firm entering the transaction (see section 6.1.1.1).
- SenderSubID (50) = set to a valid already authenticated username (see chapter 5 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.

6.3 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 trader	11 = Order Originating Trader	Specified the originating trader for FIX connections where one FIX user is shared between multiple traders.
Counterparty in Trade Capture Reports	Firm	17 = Contra Firm	Counterparty in Trade Capture Reports.
Give-Up Clearing Firm	Firm	14= ClearingFirm	Specifies the CMTA Number (OCC Member that will clear the trade) up to 5-character numeric.
Firm that entering on-behalf trade reports	Firm	7 = Entering Firm	Only distributed over Fix Drop Copy

NOTE : Executing Trader is identified by the implicit party identifier fields (SenderCompID and SenderSubID):

7 ORDER MANAGEMENT

7.1 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.

7.2 PASS-THRU FIELDS

The NFX Trading System offers the possibility for the clients to utilize two fields as pass-thru fields on incoming transactions. The values of those two fields are echoed back to the client in subsequent outgoing transactions. The fields are:

- Account (1) will be mapped to ex_client, and echoed back in Account in subsequent transactions. This is a mandatory field that must match the account detail provided.
- AllocID (70) will be mapped to customer_info, and echoed back in subsequent transactions.

NOTE: The two pass-thru fields are limited in length. The Account field is limited to 10 characters, and AllocID is limited to 15 characters.
NOTE 2: The pass-thru fields are *not* echoed back on rejects.
NOTE 3: If Account is to be used as an actual account, the account field is case sensitive.
NOTE 4: The AllocID field may be overwritten on outbound Trade Capture Reports *in rare cases* due to manual intervention by the marketplace.

7.3 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 short name 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 provided via FIX Reference Data.
  - The Orderbook ID *can* be different across trading days for the same security.

NOTE: In some messages the Symbol field is required (as a consequence of being the first field in a repeating group). If the Orderbook ID is used as the identifier, Symbol must be set to “[N/A]”.

7.4 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
Also known as *Standard Combinations*. Marketplace-defined multileg securities made available for trading. In The NFX Trading System 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 9, Multileg Orders for additional details.

7.5 MAIN WORKFLOW

7.5.1 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.

7.5.2 Fills

When an order is filled the Execution Report will contain details about the fill. See section 7.10.15 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.

7.5.2.1 Trade Match ID

The TrdMatchID (880) contains the match id generated by the system. TrdMatchID will hold the 64 bit binary match id encoded as a 16 byte hex string.

NOTE: TrdMatchID is also set in Trade Capture Report confirmation messages.

7.5.3 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*.
  NOTE: OrderQty set to 0 will leave OrderQty unchanged.
- The required fields must be set regardless if they are changed or not.

7.5.3.1 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 system allows. The following attributes are allowed to change:

- OrderQty (38)
- MaxFloor(111)
- TimeInForce (59) together with ExpireDate (432)
- Account (1), pass-thru field
- AllocID (70), pass-thru field

- Price (44)
- PositionEffect (77)

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: If 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.

7.5.3.2 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 7.10.14 for message details.

7.5.4 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.

7.5.4.1 Unsolicited cancellation of orders entered via FIX

Orders entered via FIX may be cancelled by the marketplace. In such an event an Execution Report – Unsolicited Cancel will be sent out over FIX. See section 7.10.12 for message details.

7.5.5 Order suspension 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 7.10.16 for message details.

Suspended orders may be cancelled using ordinary Order Cancel Request messages.

NOTE: The Execution Report –Order Suspended will not contain TargetSubID (57).
NOTE 2: Suspended orders will be cancelled at end-of-day.

7.6 ORDER FEATURES

7.6.1 Order Identification

7.6.1.1 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.

7.6.1.2 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: 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.

7.6.1.3 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.

7.6.1.4 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.

7.6.2 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.

7.6.3 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).

7.6.3.1 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.

7.6.3.2 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.

7.6.4 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 0 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)	
6	Good Till Date (GTD)	GTD orders must have ExpireDate (432) set.

7.6.5 Quantity Conditions

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.

7.6.6 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: 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.
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.

7.6.6.1 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.
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*
- TriggerPriceDirection (1109) to indicate price movement direction

7.6.6.2 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 immediately triggered but does not immediately trade.

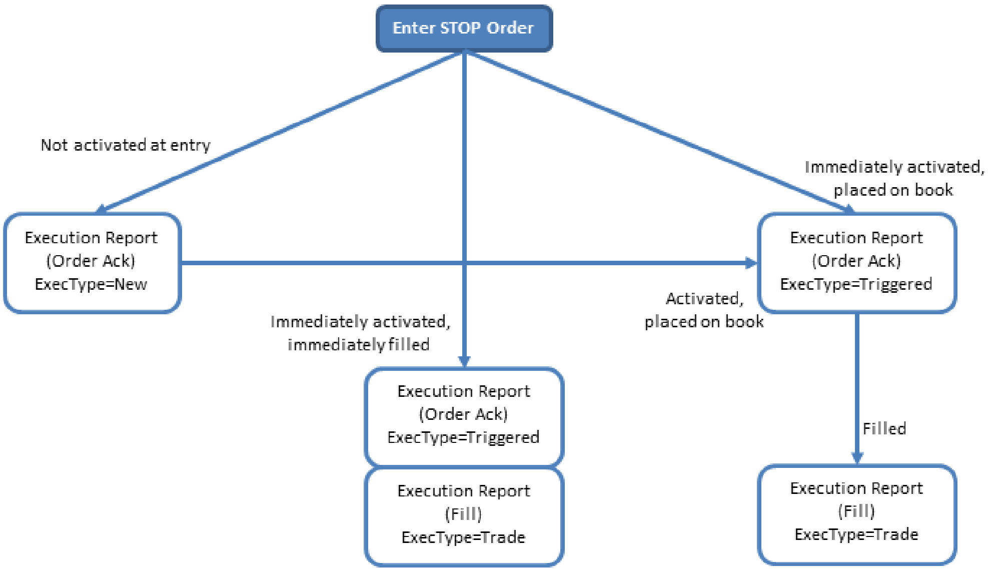

Enter STOP Order
Not activated at entry
Immediately activated, placed on book
Execution Report (Order Ack) ExecType=New
Execution Report (Order Ack) ExecType=Triggered
Activated, placed on book
Immediately activated, immediately filled
Execution Report (Order Ack) ExecType=Triggered
Execution Report (Fill) ExecType=Trade
Filled
Execution Report (Fill) ExecType=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.

7.6.6.3 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.*

7.7 MISSING REQUIRED FIELDS IN REJECTS

Due to the way the back-end works, certain fields required in standard FIX 5.0 SP2 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]".

7.8 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 7.10.18 for message details.
NOTE: The user must be prepared to receive this message as an alternative response to all other business messages.

7.9 HOW TO INTERPRET THE MESSAGE DETAILS LISTINGS

7.9.1 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.

7.9.2 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*.

7.10 MESSAGE DETAILS

7.10.1 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		Q	
→	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: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
1	Account		Q	Mandatory account identifier set by client
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)
55	Instrument/Symbol			Instrument 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: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.
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)
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD

528	OrderCapacity	Q	The type of business conducted.
			Valid values:
			A = Agency
			P = Principal

582	CustOrderCapacity	Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other

77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
			C = Close
			O = Open

70	AllocID		Optional pass-thru field set by client and echoed back by marketplace.
	Standard Trailer	Y	

7.10.2 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
55	Instrument/Symbol			Instrument 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	
	Standard Trailer	Y	

7.10.3 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		Q	
→	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: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
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			Mandatory account identifier set by client
70	AllocID			Optional pass-thru field set by client and echoed back by marketplace.
21	HandlInst			Instructions for order handling on Broker trading floor. Valid values: 1 = Automated execution order, private, no Broker intervention (default value)
111	MaxFloor			For hidden orders.
55	Instrument/Symbol			Instrument 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)
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD

528	OrderCapacity	Q	The type of business conducted.
			Valid values:
			A = Agency
			P = Principal

582	CustOrderCapacity	Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other

77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
			C = Close
			O = Open

	Standard Trailer	Y	

7.10.4 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	

7.10.5 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	

7.10.6 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
x	OrderID		Y	
11	ClOrdID		Q	Unique identifier set by the client.
453	NoPartyIDs			
→	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: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
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	Q	Mandatory account identifier set by client
55	Instrument/Symbol	Q	Instrument 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: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.
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)
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD
111	MaxFloor		For hidden orders.

528	OrderCapacity	Q	The type of business conducted.
			Valid values:
			A = Agency
			P = Principal

582	CustOrderCapacity	Q	Capacity of customer placing the order

			Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
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		NFX Extension: Optional pass-thru field set by client and echoed back by marketplace.

77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
			C = Close
			O = Open

797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
	Standard Trailer	Y	

7.10.7 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			
→	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: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
17	ExecID		Y	
150	ExecType		Y	Valid values: 4 = Canceled
39	OrdStatus		Y	Valid values: 4 = Canceled
1	Account		Q	Mandatory account identifier set by client
55	Instrument/Symbol		Q	Instrument 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	Q	The type of business conducted.
			Valid values:
			A = Agency
			P = Principal

582	CustOrderCapacity	Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
151	LeavesQty	Y	Will be 0.
14	CumQty	Y	
6	AvgPx	Y	Note: Always set to 0.0
60	TransactTime	Q	
70	AllocID		NFX Extension: Optional pass-thru field set by client and echoed back by marketplace.

77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
			C = Close
			O = Open

797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
	Standard Trailer	Y	

7.10.8 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.
→	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: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
17	ExecID	Y	
150	ExecType	Y	Valid values: 4 = Canceled
39	OrdStatus	Y	Valid values: 4 = Canceled
1	Account	Y	Mandatory account identifier set by client
55	Instrument/Symbol	Q	Instrument 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)
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD

528	OrderCapacity	Q	The type of business conducted.
			Valid values:
			A = Agency
			P = Principal

582	CustOrderCapacity	Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
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		NFX Extension: Optional pass-thru field set by client and echoed back by marketplace.

77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
			C = Close
			O = Open

797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
	Standard Trailer	Y	

7.10.9 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	

7.10.10 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	System order number
11	ClOrdID	Q	Unique identifier set by the client.
41	OrigClOrdID		ClOrdID of the order to modify/cancel.
453	NoPartyIDs		
→	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: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
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	Q	Mandatory account identifier set by client
55	Instrument/Symbol	Q	Instrument 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: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.
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)
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD
111	MaxFloor		For hidden orders.

528	OrderCapacity	Q	The type of business conducted.
			Valid values:
			A = Agency
			P = Principal

582	CustOrderCapacity	Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
151	LeavesQty	Y	
14	CumQty	Y	
6	AvgPx	Y	Note: Always set to 0.0
60	TransactTime	Q	
70	AllocID		NFX Extension: Optional pass-thru field set by client and echoed back by marketplace.

77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
			C = Close
			O = Open

797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages

	Standard Trailer	Y	

7.10.11 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	System 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		
→	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: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
17	ExecID	Y	
150	ExecType	Y	Valid values: 4 = Canceled
39	OrdStatus	Y	Valid values: 4 = Canceled
1	Account	Q	Mandatory account identifier set by client
55	Instrument/Symbol	Q	Instrument 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	Q	The type of business conducted.
			Valid values:
			A = Agency
			P = Principal

582	CustOrderCapacity	Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
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		NFX Extension: Optional pass-thru field set by client and echoed back by marketplace.
77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:

			C = Close
			O = Open

797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
	Standard Trailer	Y	

7.10.12 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	System 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: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
17	ExecID	Y	
150	ExecType	Y	Valid values: 4 = Canceled
39	OrdStatus	Y	Valid values: 4 = Canceled
1	Account	Y	Mandatory account identifier set by client
55	Instrument/Symbol	Q	Instrument 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	Q	The type of business conducted.
			Valid values:
			A = Agency
			P = Principal

582	CustOrderCapacity	Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
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		NFX Extension: Optional pass-thru field set by client and echoed back by marketplace.

77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
			C = Close
			O = Open

797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
58	Text		Information on why the order was canceled. Order detail change reason code will match these values: 1 = Order cancelled 9 = Order cancelled by system 10 = Order cancelled on-behalf 15 = Order cancelled by the system due to price limit change 17 = Linked order leg cancelled 19 = Order expired due to last trading day for the order 20 = Order cancelled due to that trading is halted 34 = Market order cancelled during auction 41 = Order cancelled due to market maker delta protection limit breached 42 = Order cancelled due to market maker quantity protection limit breached 43 = Order cancelled due to self match prevention Other = Unsolicited Cancel
	Standard Trailer	Y	

7.10.13 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 = 5 or 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.
→	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: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
17	ExecID		Y	
150	ExecType		Y	Valid values: D = Restated
39	OrdStatus		Y	Valid values: 0 = New 1 = Partially Filled
1	Account		Q	Mandatory account identifier set by client
55	Instrument/Symbol		Q	Instrument 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: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.
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)
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD
111	MaxFloor		For hidden orders.

528	OrderCapacity	Q	The type of business conducted.
			Valid values:
			A = Agency
			P = Principal

582	CustOrderCapacity	Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
151	LeavesQty	Y	
14	CumQty	Y	

6	AvgPx	Y	Note: Always set to 0.0
60	TransactTime	Q	
378	ExecRestatementReason	Q	Valid values: 5 = Partial decline of OrderQty 8 = Market (Exchange) option
70	AllocID		NFX Extension: Optional pass-thru field set by client and echoed back by marketplace.

77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
			C = Close
			O = Open

797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
	Standard Trailer	Y	

7.10.14 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	System 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: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
17	ExecID	Y	
150	ExecType	Y	Valid values: D = Restated
39	OrdStatus	Y	Valid values: 0 = New 1 = Partially Filled
1	Account	Q	Mandatory account identifier set by client
55	Instrument/Symbol	Q	Instrument 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		
1100	TriggeringInstruction/ TriggerType		Determines what should trigger an order modification. Valid values: 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.
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: 1 = Good Till Cancel (GTC) 6 = Good Till Date (GTD)
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD
111	MaxFloor		For hidden orders.

528	OrderCapacity	Q	The type of business conducted.
			Valid values:
			A = Agency
			P = Principal

582	CustOrderCapacity	Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
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		NFX Extension: Optional pass-thru field set by client and echoed back by marketplace.

77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
			C = Close
			O = Open

797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
	Standard Trailer	Y	

7.10.15 Execution Report – Fill (out)

Purpose: Order Fill.

NOTE: For Multileg (Combination) order fills, see section 9.5.5.

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.
→	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:11 = Order Originating Trader14 = Give-Up Clearing Firm
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		Q	Optional pass-thru field set by client.
55	Instrument/Symbol		Q	Instrument 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)
432	ExpireDate			Date of order expiration. Conditionally required if TimeInForce = GTD
111	MaxFloor			For hidden orders.

528	OrderCapacity		Q	The type of business conducted.
				Valid values:
				A = Agency
				P = Principal

582	CustOrderCapacity	Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
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		NFX Extension: Optional pass-thru field set by client and echoed back by marketplace. Back Office Account.

77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
			C = Close
			O = Open

797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
	Standard Trailer	Y	

7.10.16 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.
→	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: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
17	ExecID		Y	
150	ExecType		Y	Valid values: 9 = Suspended
39	OrdStatus		Y	Valid values: 9 = Suspended
1	Account		Q	Mandatory account identifier set by client
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: 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.
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)
432	ExpireDate		Date of order expiration. Conditionally required if TimeInForce = GTD
111	MaxFloor		For hidden orders.

528	OrderCapacity	Q	The type of business conducted.
			Valid values:
			A = Agency
			P = Principal

582	CustOrderCapacity	Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
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		NFX Extension: Optional pass-thru field set by client and echoed back by marketplace.

77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
			C = Close
			O = Open

797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
	Standard Trailer	Y	

7.10.17 Execution Report – Expired (out)

Purpose: GTD **or GTC** Order Expired. For GTD orders the Expired transaction will be sent the day after the order expired. 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	System 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: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
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	Q	Mandatory account identifier set by client
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)
111	MaxFloor		For hidden orders.

528	OrderCapacity	Q	The type of business conducted.
			Valid values:
			A = Agency
			P = Principal

582	CustOrderCapacity	Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
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		NFX Extension: Optional pass-thru field set by client and echoed back by marketplace.

77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
			C = Close
			O = Open

797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
	Standard Trailer	Y	

7.10.18 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	

8 CONTINGENT (LINKED) ORDERS

8.1 OVERVIEW

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.

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 6), 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

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)

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.

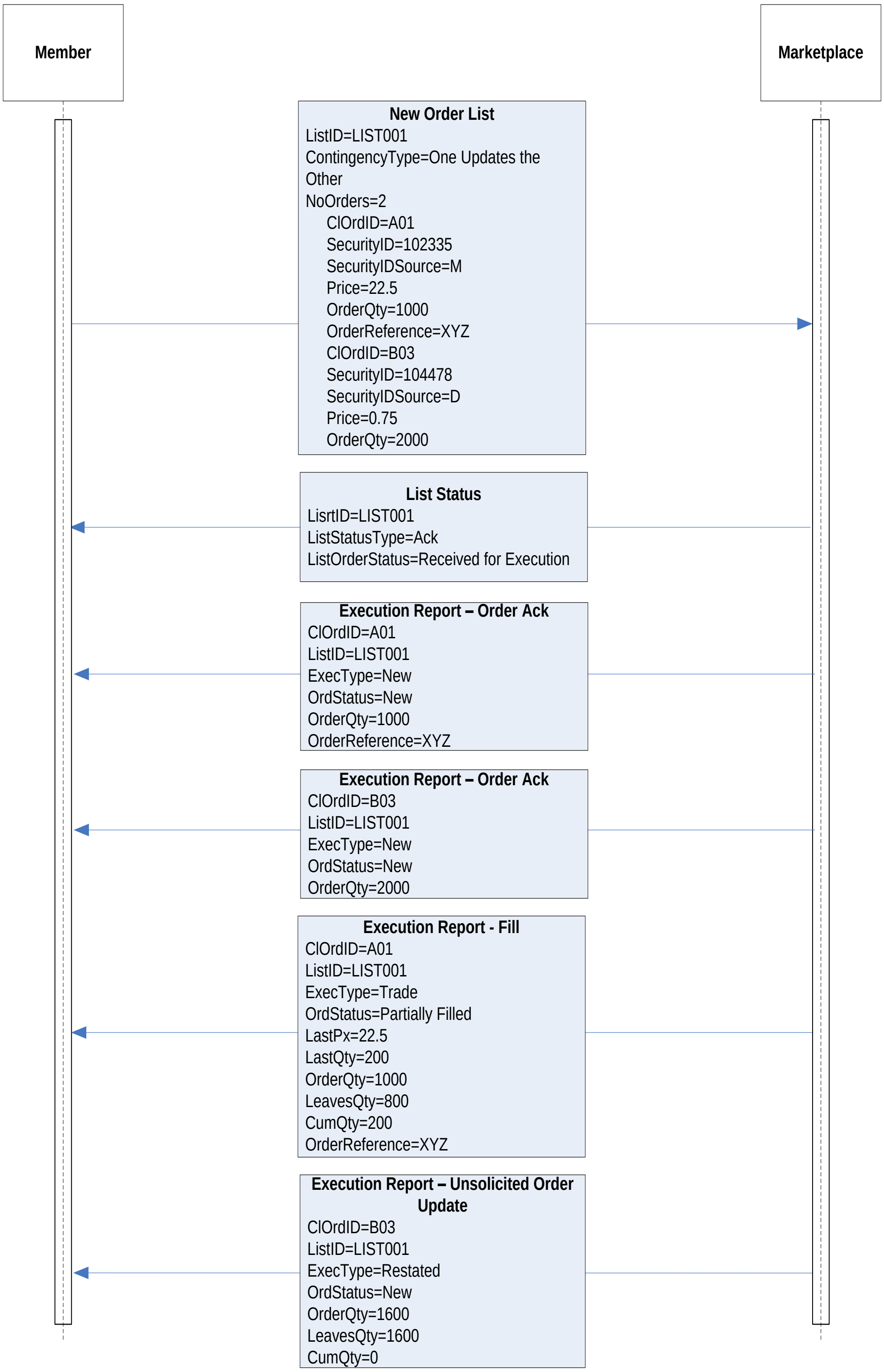
A Reserve size (hidden order) is not allowed.

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

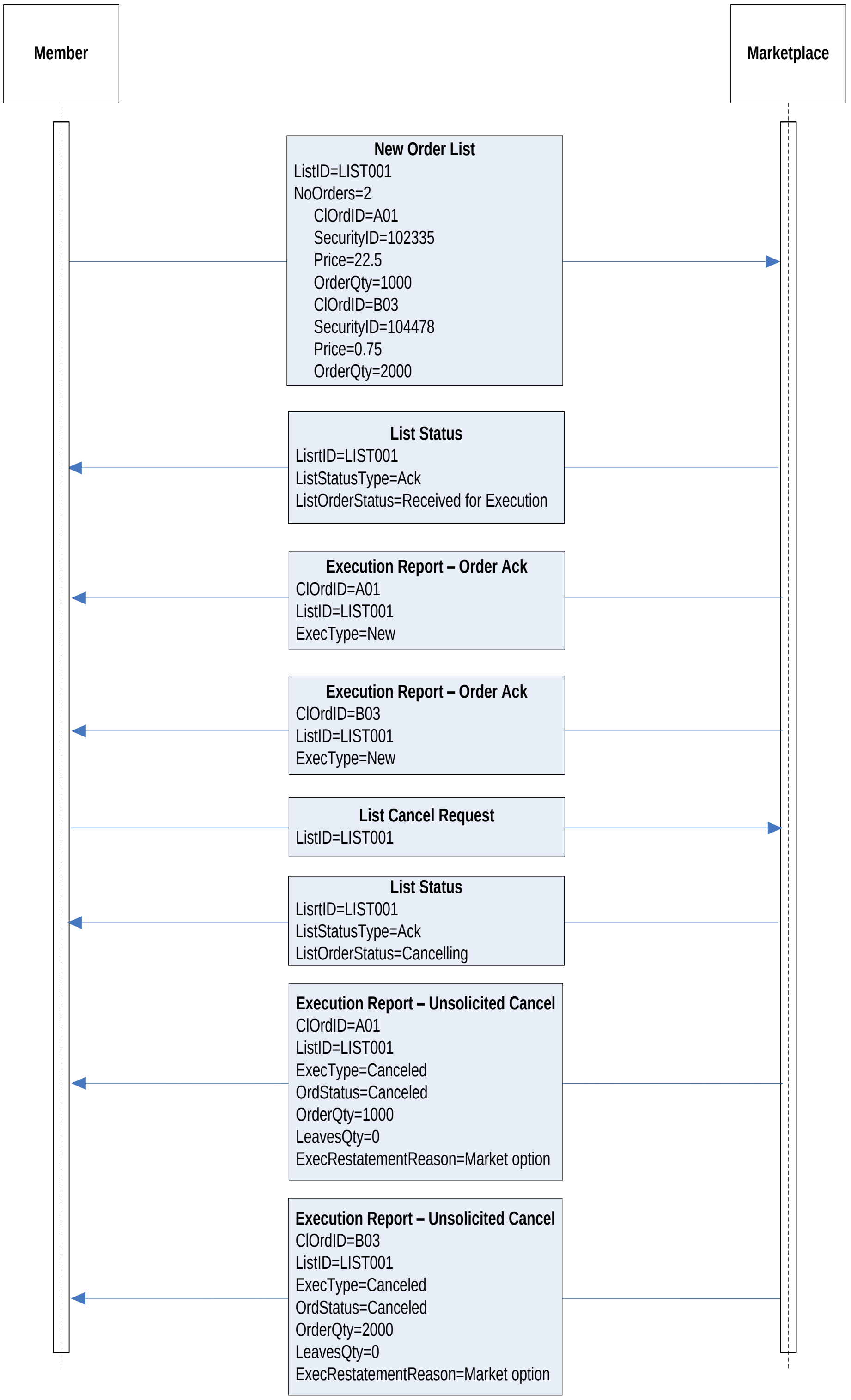
Execution Report – Order Ack
ClOrdID=B03
ListID=LIST001
ExecType=New
OrdStatus=New
OrderQty=2000

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	Y	NASDAQ 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		Q	Required repeating group used for order originating traderand for optional Clearing Firm.
→	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: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
73	NoOrders		Y	Number of orders in this message.
→	11	ClOrdID	Y	Client Order ID
→	67	ListSeqNo	Y	Required in FIX, but ignored
→	1	Account	Q	Mandatory account identifier set by client. Needs to be the same for all orders in the transaction.
→	1089	MatchIncrement		NASDAQ Extension
→	55	Symbol		OMNet short name. Symbol or SecurityID+SecurityIDSource must be set.
→	48	SecurityID		Orderbook ID
→	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	Valid values: 0 = Day 1 = Good Till Cancel (GTC) 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FoK) 6 = Good Till Date (GTD)

→	528	OrderCapacity	Q	Designates the capacity of the firm placing the order. Valid values:
				P = Principal
				A = Agency

→	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)

→	582	CustOrderCapacity	Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member

				4 = Other
	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 Extension: Defines the type of contingency. Valid values: 0 = Valid
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		
→	1117 RootPartyID	Q	Party identifier.
→	1118 RootPartyIDSource	Q	Valid values:
→	1119 RootPartyRole	Q	Identifies the type of role for the PartyID specified. Valid values: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
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: 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 Extension: Defines the type of contingency. Valid values: 0 = Valid
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	

9 MULTILEG ORDERS

9.1 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 The NFX Trading System, 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.

9.2 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 7, Order Management for information on aspects shared with single order messages.

9.2.1 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.

9.2.2 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:

- Triggers are not allowed.

9.3 MAIN WORKFLOW

9.3.1 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 9.5.2 for message details).

9.3.1.1 TMC Registration Response

The response to a submitted TMC registration request is a Security Definition message. This message will not contain a usable instrument identifier. The SecurityResponseType (tag 323) will indicate whether the request was successful or not. This message only serves as an acknowledgement (or reject) that the TMC registration request has been received. *It will not contain the appropriate instrument identifiers that can be used to enter orders and trades over FIX.*
The actual instrument definition will be returned in the subsequent Security Definition Update Report (see 9.5.4). See also workflow diagram 9.4.1 for message flows.

9.3.2 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.

9.3.3 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.

9.3.4 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 7 for details. The only difference is with fills, which are sent per leg (see section 9.3.5).

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.

9.3.5 Fills

When multileg orders are filled, Execution Reports are issued. The Execution Report – Combination Order Fill is used for multileg fills. See section 9.5.5 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.

9.3.5.1 Message Fragmentation

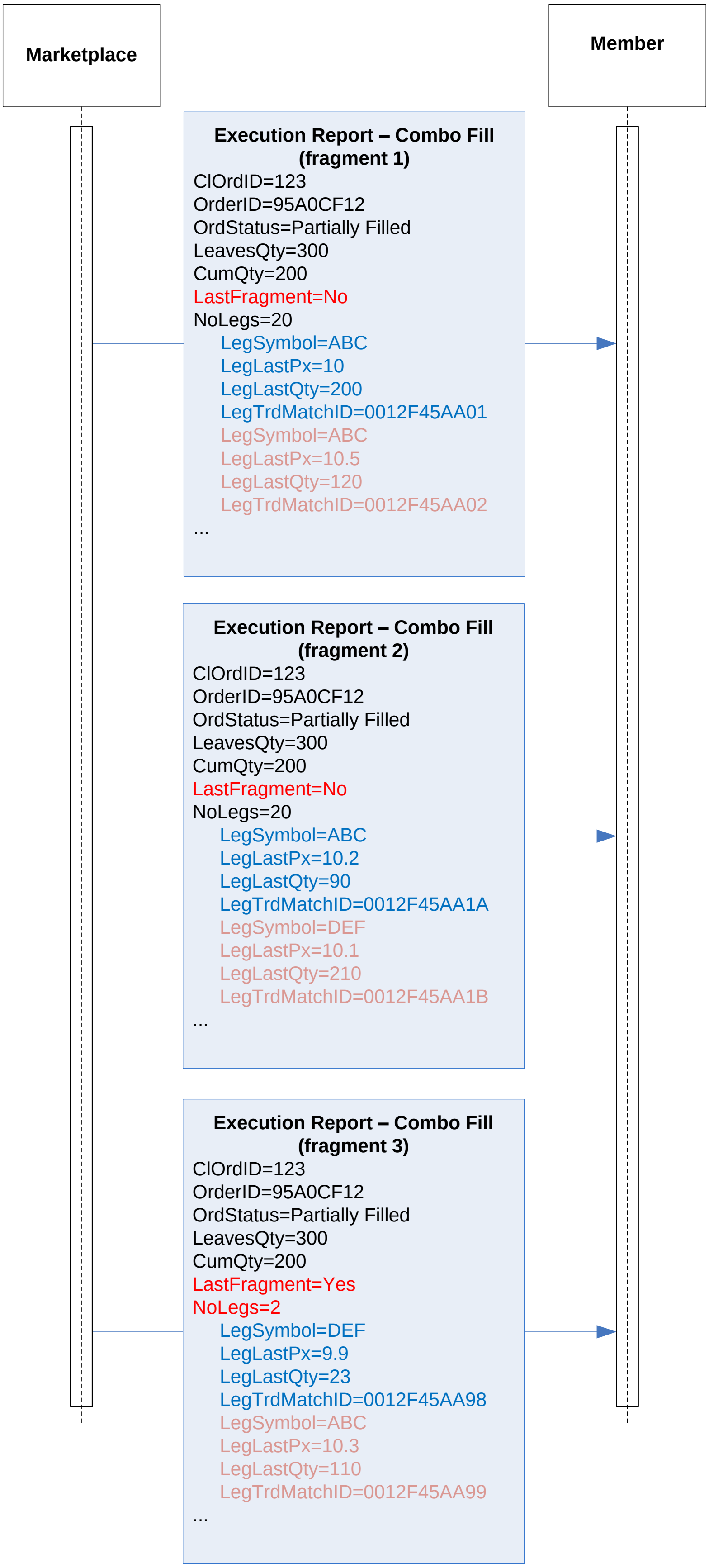
The multileg order fills (Execution Reports) may contain many leg trade entries. If the message is deemed too large to send it will be split up into multiple fragments. Each fragment is a valid Execution Report following the format of the Execution Report – Combination Order Fill. Each of the fragments will contain the exact same contents except:

- Some Standard Header fields will be different (such as MsgSeqNum)
- Standard Trailer will be different (different CheckSum)
- The NoLegs (555) repeating group will contain the leg trades for the current fragment

All other fields like ExecID, ClOrdID, and LeavesQty etc will be set to the same value in all fragments.

The LastFragment (893) field is the indicator for a fragmented message. If it is not in the message, there is no fragmentation. If the client detects LastFragment=N, this is the indicator that the message is fragmented and more fragments will be sent. The last fragment will have LastFragment=Y.

Example:
An aggressive mutlileg order executes against a large number of outright orders. The result is an execution with 42 trades in the legs. The system decides that only 20 leg trades fit into a single Execution Report and generates the following sequence of messages:
NOTE: Only fields relevant to the example are present.



Execution Report – Combo Fill (fragment 1)

ClOrdID=123
OrderID=95A0CF12
OrdStatus=Partially Filled
LeavesQty=300
CumQty=200
LastFragment=No
NoLegs=20
    LegSymbol=ABC
    LegLastPx=10
    LegLastQty=200
    LegTrdMatchID=0012F45AA01
    LegSymbol=ABC
    LegLastPx=10.5
    LegLastQty=120
    LegTrdMatchID=0012F45AA02
...

Execution Report – Combo Fill (fragment 2)

ClOrdID=123
OrderID=95A0CF12
OrdStatus=Partially Filled
LeavesQty=300
CumQty=200
LastFragment=No
NoLegs=20
    LegSymbol=ABC
    LegLastPx=10.2
    LegLastQty=90
    LegTrdMatchID=0012F45AA1A
    LegSymbol=DEF
    LegLastPx=10.1
    LegLastQty=210
    LegTrdMatchID=0012F45AA1B
...

Execution Report – Combo Fill (fragment 3)

ClOrdID=123
OrderID=95A0CF12
OrdStatus=Partially Filled
LeavesQty=300
CumQty=200
LastFragment=Yes
NoLegs=2
    LegSymbol=DEF
    LegLastPx=9.9
    LegLastQty=23
    LegTrdMatchID=0012F45AA98
    LegSymbol=ABC
    LegLastPx=10.3
    LegLastQty=110
    LegTrdMatchID=0012F45AA99
...

9.4 WORKFLOWS

9.4.1 Registering a new TMC instrument

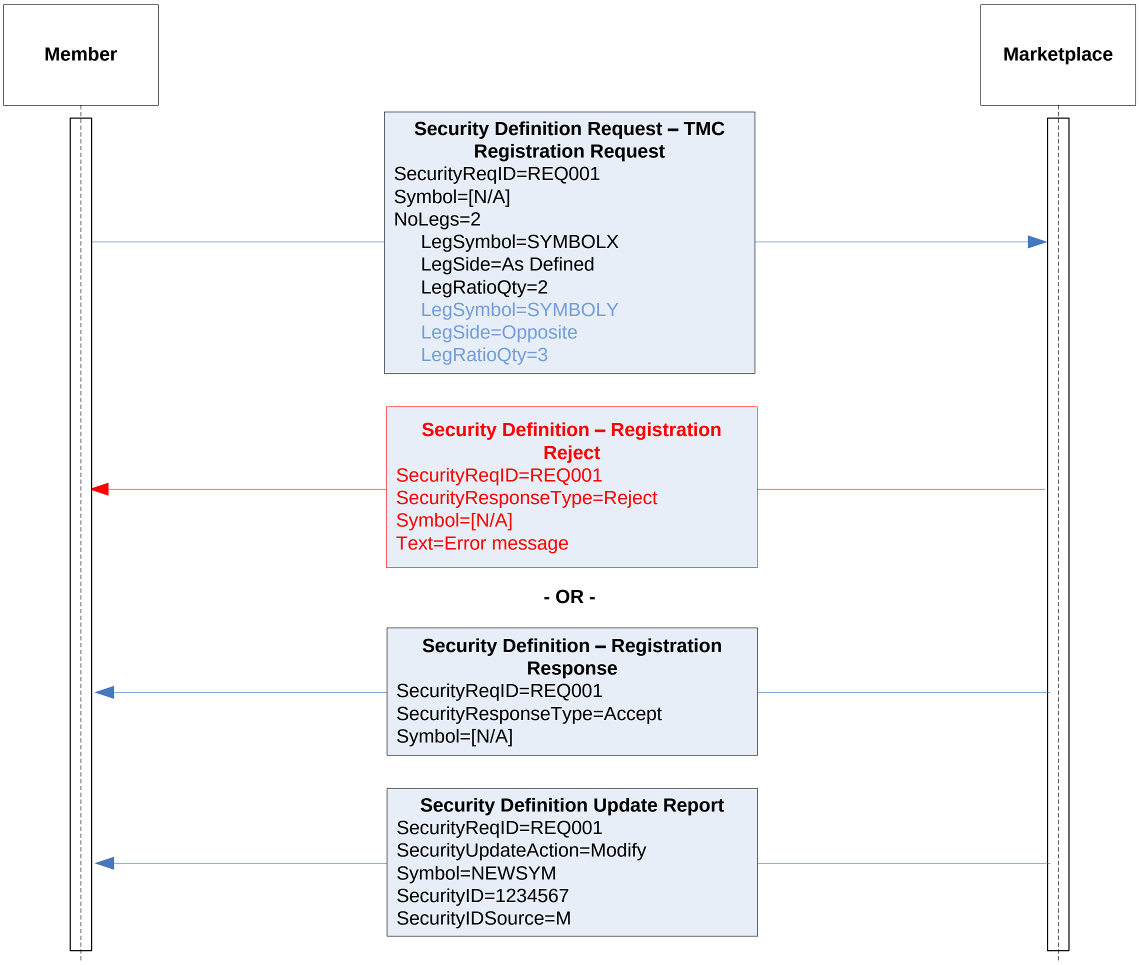
This example shows how to create a new Tailor-Made Combination with two legs. The two legs of requested TMC have the following properties:

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).



9.5 MESSAGE DETAILS

9.5.1 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]
20017	OriginatingTrader		Originating Trader

762	SecuritySubType		Q	Indicates Strategy subtype. Valid values if defined: COV = Covered Options, for the future leg 637, 1017 must be specified.
555	NoLegs		Q	Number of legs
→	600	LegSymbol	Q	Instrument 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.
→	637	LegLastPx		The price of the Future leg in TMC for Covered Options
→	1017	LegOptionRatio		The leg delta of the Future leg in TMC for Covered Options.
	Standard Trailer		Y	

9.5.2 Security Definition – TMC Registration Response (out)

Purpose: Accept of a TMC 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 = NFX Trading System series definition (NFX Extension)
	Standard Trailer	Y	

9.5.3 Security Definition – TMC Registration Reject (out)

Purpose: Reject of a TMC 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	

9.5.4 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 Extension)
762	SecuritySubType			Indicates Strategy subtype. Valid values if defined: COV = Covered Options, for the future leg 637, 1017, 20018 must be specified.
555	NoLegs			Number of legs (for strategy/combination) instruments. **NOTE**: Only used for strategies.
→	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

→	637	LegPastPx		The price of the Future leg in TMC for Covered Options.
→	1017	LegOptionRatio		The leg delta of the Future leg in TMC for Covered Options.
→	20018	LegQuantityFuture		Leg Future Quantity Multiple of TMC.
	Standard Trailer		Y	

9.5.5 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.
→	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: 11 = Order Originating Trader 14 = Give-Up Clearing Firm
880	TrdMatchID		Q	Match ID assigned by the matching engine.
20034	ComboGroupID			NFX Extension: Can be used to group trades in the legs of a strategy (will be the same for all legs).
17	ExecID		Y	
150	ExecType		Y	Valid values: F = Trade
39	OrdStatus		Y	Valid values: 1 = Partially filled 2 = Filled
1	Account		Q	Mandatory account identifier set by client
55	Instrument/Symbol		Q	Combination orderbook Instrument 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
38	OrderQtyData/OrderQty		Q	Combination (strategy) order quantity.
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		Q	Net price of the combination as entered in the order.
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)
151	LeavesQty		Y	
14	CumQty		Y	
6	AvgPx		Y	Note: Always set to 0.0
893	LastFragment			If set, indicates that the message has been fragmented. The last fragment will have LastFragment set to Y. Valid values: Y = Yes N = No (this indicates that the message is fragmented – more fragments will be sent)
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	NFX Extension: Quantity traded in this leg
→	20200	LegTrdMatchID	Q	NFX Extension: Match ID for the current leg.

528	OrderCapacity		Q	The type of business conducted.
				Valid values:
				A = Agency
				P = Principal

582	CustOrderCapacity		Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
32	LastQty		Q	Quantity (e.g. shares) bought/sold on this (last) fill.
31	LastPx			Price of this (last) fill. This tag will not be set when there is multiple prices in the same combo match.
70	AllocID			NFX Extension: Optional pass-thru field set by client and echoed back by marketplace.

77	PositionEffect			*NFX Extension.* Defines the requested position update for the account. Valid values:
				C = Close
				O = Open

797	CopyMsgIndicator			Set to ‘Y’ on Drop Copy messages
	Standard Trailer			

10 GENERAL QUOTE HANDLING

10.1 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.

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

10.2 SOLUTION RESTRICTIONS

Quoting in FIX has the following restrictions:

- Only one two-sided quote per participant per instrument is allowed. This simplifies quote cancellation and generation of quote identifiers, see below.
- 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 either by sending a quote cancel message or 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** included in quoting transactions. Trades originating from quotes will have firm specific default values assigned as defined during configuration of each user.

10.3 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.

It is possible to replace a double sided quote and leave one side unchanged (for example to avoid losing priority).
To leave one side of the quote unchanged, set the quantity (Bid or Offer size) on that side set to 0.

NOTE: To differentiate between a single sided quote cancel (see Quote Cancellation below) and an unchanged quote, the price (BidPx or OfferPx) on the side in question **must be set to a non-zero value**. This is true even if the currently quoted price is zero (zero is a valid price for certain instruments).

10.4 QUOTE CANCELLATION

A quote can be canceled (or withdrawn) either by sending a Quote Cancel message or 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

It is possible to cancel only one side of a double-sided quote by setting the price and the quantity on that side to zero.

It is possible to cancel all the participants' quotes for a specific MarketID and UnderlyingSercurityID using the Quote Cancel message, i.e. mass cancel of quotes. If there are some rejected cancelations in the mass cancel a Mass Quote Acknowledgement – some quotes cancelled (out) message will be sent in response with the amount of successfully canceled quote sides in tag 1168 and the amount of rejected quote side cancelations in tag 1170. Each rejected quote-side canel will generate a Mass Quote Acknowledgement – quote cancel rejected (out) message.

10.5 REQUEST FOR QUOTE

Any participant can issue a quote request message to request other members to enter quotes in an order book. The Quote Request is broadcasted to all members via public Market Data. The Quote Request must contain:

- The order book (instrument) a quote is requested for.

It may also optionally contain:

- A minimum quantity
- A side (if not set a request for a double-sided quote is assumed)
- A Lot Size to request a certain Lot Type
- Order Originating Trader (will not be published in Market Data)

Any participant may act upon the Quote Request by entering regular quotes in the requested order book.

10.6 MAIN WORKFLOW

10.6.1 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. The limit is currently set to **29** double-sided 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.

10.6.1.1 Limitations

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

10.6.2 Mass Quote Acknowledgement

Mass Quote Acknowledgement is used as the application level response to a Mass Quote and Quote Cancel messages.

The Mass Quote Acknowledgement as a response to a Mass Quote 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 Acknowledgement as a response to a Quote Cancel contains a field for reporting the number of quotes that were successfully cancelled. It also contains information regarding if there were quotes that were not possible to cancel.

10.6.3 Quote Rejects

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

10.6.4 Quote Request

The Quote Request message is used to request for quotes in a specific order book.

A Quote message containing unique QuoteID is returned if the Quote Request is accepted.

A Quote Request Reject is returned by the system if the Quote Request is rejected.

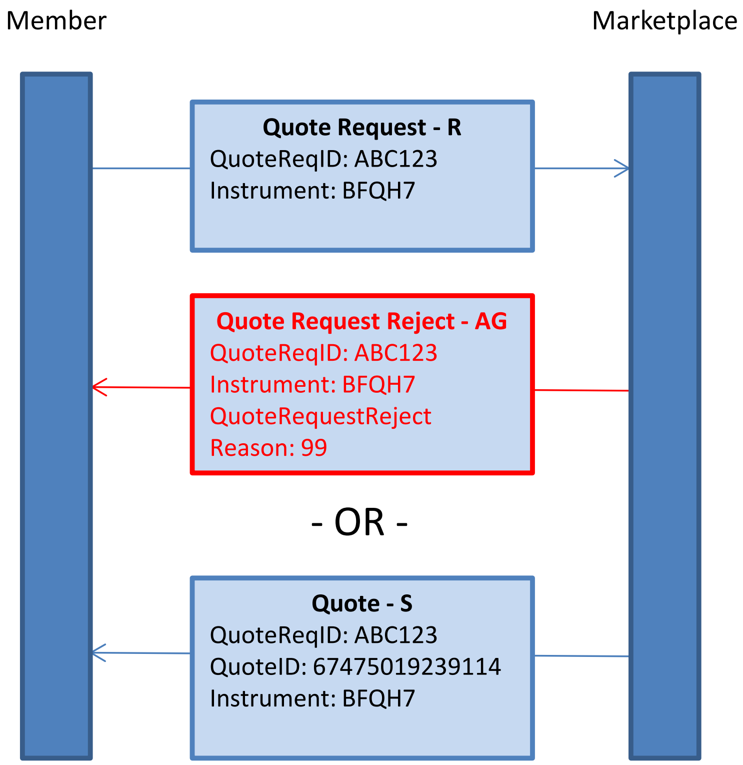


Figure 2 Quote request workflow

10.6.5 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.

10.6.6 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.

- 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.

10.7 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 *Mass Quotes* the supported values are:

- 1 = Requests acknowledgement of invalid or erroneous quote messages only (negative)
- 3 = Request acknowledgement of all quotes accepted or all quotes canceled

10.8 QUOTE STATE CHANGES

Quote state changes are divulged by:

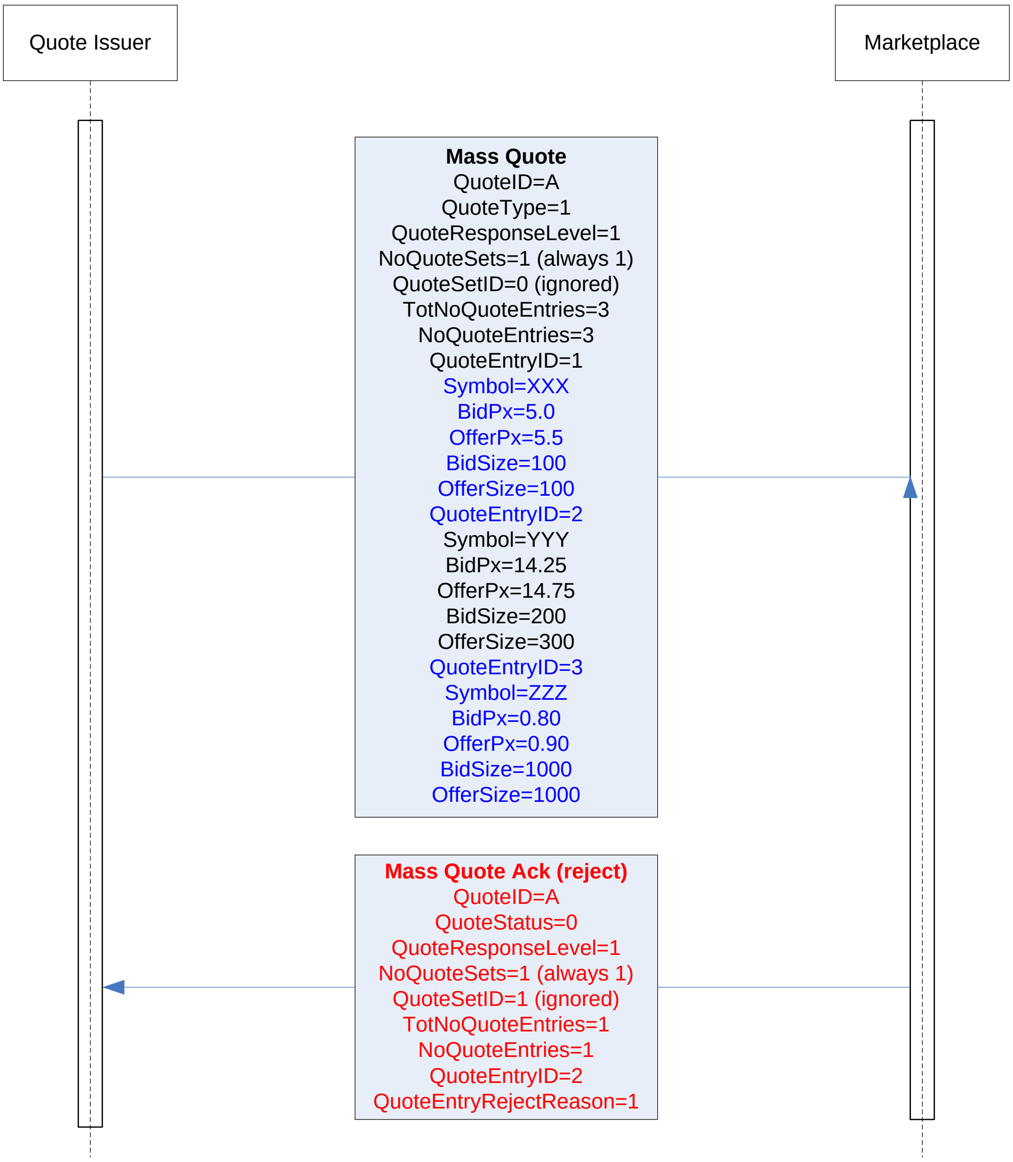
- The Mass Quote Acknowledgement message after a quote update (excluding fills) and subject to the specified or bilaterally agreed QuoteReponseLevel

- Execution Reports after fills

10.9 WORKFLOWS

10.9.1 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 similarly on 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

10.10 MESSAGE DETAILS

10.10.1 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 3 = Acknowledge all quotes accepted or canceled
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		Instrument 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	

10.10.2 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: 1 = Tradeable
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	

10.10.3 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: 1 = Tradeable
58	Text		
	Standard Trailer	Y	

10.10.4 Mass Quote Acknowledgement – All Quotes Accepted or Canceled (out)

TAG	FIX TAG NAME	REQ'D	COMMENT
	Standard Header	Y	MsgType = b
117	QuoteID	Q	From Mss Quote
297	QuoteStatus	Y	Identifies the status of the mass quote acknowledgement. Valid values: 0 = Accept
301	QuoteResponsLevel	Q	Level of Response requested from receiver of quote messages. Valid values: 3 = Summary Acknowledgement
537	Quote Type	Q	Identifies the type of quote. Valid values: 1 = Tradeable
58	Text		
37	Order ID		
	Standard Trailer	Y	

10.10.5 Quote Cancel (in)

TAG	FIX TAG NAME	REQ'D	COMMENT
	Standard Header	Y	MsgType = Z

1166	QuoteMsgID			Q	Quote cancel assigned message identifier. Must be unique
298	QuoteCancelType			Y	Specified level of quote cancel: Valid values: 1 = Cancel per Orderbook ID 3 = Cancel per market and underlying
1301	MarketID				Market code. Required if QuoteCancelType = 3 Valid Values: 3 = Energy
295	NoQuoteEntries				Multiple quote entries not supported. Will always be 1. Only supported to specify one instrument or one underlying
→	55	Instrument/Symbol		Q	Instrument short name. Symbol or SecurityID+SecurityIDSource must be set. If SecurityID is set or if QuoteCancelType = 1 else must be set to “[N/A]”.
→	48	Instrument/SecurityID			Orderbook ID.
→	22	Instrument/SecurityIDSource			Valid values: M = Marketplace-assigned identifier
→	711	NoUnderlyings			Optional repeating group. Required if QuoteCancelType = 3.
→	→	311	UnderlyingSymbol	Q	Underlying identity. Required in FIX but ignored.
→	→	309	UnderlyingSecurityID	Q	Underlying Code.
→	→	305	UnderlyingSecurityIDSource	Q	Valid values: u = Underlying Code
	Standard Trailer			Y	

10.10.6 Mass Quote Acknowledgement – some quotes cancelled (out)

TAG	FIX TAG NAME			REQ’D	COMMENT
	Standard Header			Y	MsgType = b
117	QuoteID				
297	QuoteStatus			Y	1 = Canceled per Orderbook ID accepted 3 = Canceled for Underlying accepted 5 = Rejected
301	QuoteResponseLevel			Q	Level of Response requested from receiver of quote messages. Valid values: 3 = Acknowledge all quotes accepted or canceled
296	QuoteSetAckGrp/NoQuoteSets			Y	
→	302	QuoteSetID		Y	Will always be set to 1
→	1168	TotNoCxldQuotes			Number of cancelled quote sides
→	1170	TotNoRejQuotes			Number of rejected quote sides
→	295	NoQuoteEntries		Y	Number of quotes cancels that was rejected
→	→	299	QuoteEntryID		Blank for quote cancel acknowledgements
→	→	55	Instrument/Symbol	Y	Instrument short name.
→	→	48	Instrument/SecurityID	Y	Orderbook ID
→	→	22	Instrument/SecurityIDSo	Y	Valid values:

			urce		M = Marketplace-assigned identifier
→	→	300	QuoteEntryRejectReson	Y	
58	Text				
	Standard Trailer				

10.10.7 Mass Quote Acknowledgement – quote cancel rejected (out)

TAG	FIX TAG NAME			REQ'D	COMMENT
	Standard Header			Y	MsgType = b
117	QuoteID				
297	QuoteStatus			Y	5 = Rejected
301	QuoteResponseLevel			Q	Level of Response requested from receiver of quote messages. Valid values: 1 = Acknowledge only negative or erroneous quotes
296	QuoteSetAckGrp/NoQuoteSets			Y	Will always be 1
→	302	QuoteSetID		Y	Will always be set to 1
→	295	NoQuoteEntries		Y	Number of quotes cancels that was rejected
→	→	299	QuoteEntryID	Q	Will be set to [N/A]
→	→	55	Instrument/Symbol	Y	Instrument short name.
→	→	48	Instrument/SecurityID	Y	Orderbook ID
→	→	22	Instrument/SecurityIDSource	Y	Valid values: M = Marketplace-assigned identifier
58	Text				
	Standard Trailer				

10.10.8 Mass Quote Acknowledgement – all quotes cancelled (out)

TAG	FIX TAG NAME		REQ'D	COMMENT
	Standard Header		Y	MsgType = b
117	QuoteID			
297	QuoteStatus		Y	1 = Canceled per Orderbook ID accepted 3 = Canceled for Underlying accepted 5 = Rejected
301	QuoteResponseLevel		Q	Level of Response requested from receiver of quote messages. Valid values: 3 = Acknowledge all quotes accepted or canceled
296	QuoteSetAckGrp/NoQuoteSets		Y	Will always be 1
→	302	QuoteSetID	Y	Will always be set to 1
→	1168	TotNoCxldQuotes	4	Number of cancelled quote sides
	Standard Trailer			

10.10.9 Quote Request (in)

TAG	FIX TAG NAME	REQ'D	COMMENT
	Standard Header	Y	MsgType = R
131	QuoteReqID	Q	Unique identifier assigned by the requestor. Will be returned in responses.
146	NoRelatedSym	Q	Will be set to 1

→	55	Instrument/Symbol	Y	Short name of instrument. **NOTE:** Must be set to [N/A] if SecurityID is used as instrument identifier.
→	48	Instrument/SecurityID	Q	Orderbook ID
→	22	Instrument/SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
→	54	Side		Valid values: 1 = Buy 2 = Sell **NOTE:** The absence of a side implies that a two-sided quote is being requested.
→	110	MinQty		Minimum quantity requested. **NOTE:** The absence of MinQty implies that any size is acceptable.
1116	NoRootPartyIDs			Used to enter Order Originating Trader. Valid value: 1
→	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: 11 = Order Originating Trader
	Standard Trailer		Y	

10.10.10 Quote Request Reject (out)

TAG	FIX TAG NAME		REQ'D	COMMENT
	Standard Header		Y	MsgType = AG
131	QuoteReqID		Y	Unique identifier assigned by the requestor.
658	QuoteRequestRejectReason		Y	Valid values: 99 = Other
146	NoRelatedSym		Y	Will be set to 1
→	55	Instrument/Symbol	Y	Short name of instrument.
→	48	Instrument/SecurityID	Q	Orderbook ID
→	22	Instrument/SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
58	Text			Free text description of the reject.
	Standard Trailer		Y	

10.10.11 Quote (out)

TAG	FIX TAG NAME	REQ'D	COMMENT
	Standard Header	Y	MsgType = S
131	QuoteReqID	Y	Unique identifier of the submitted quote request assigned by the requestor
117	QuoteID	Y	Unique identifier of the quote request assigned by the system
55	Instrument/Symbol	Y	Short name of instrument.
48	Instrument/SecurityID	Q	Orderbook ID
22	Instrument/SecurityIDSource	Q	Valid values: M = Marketplace-assigned identifier
	Standard Trailer	Y	

10.10.12 MMProtection Set Limit (in)

TAG	FIX TAG NAME	REQ'D	COMMENT
	Standard Header	Y	MsgType = MM
309	UnderlyingSecurityID	Y	Specify the underlying code the market maker protection is valid for. As Commodity Code
311	UnderlyingSymbol	Y	Specify the underlying the market maker protection is valid for. Must be set to NONE
20114	ProtectionRequestID	Y	Unique identifier assigned by the requestor. Will be returned in responses.
20110	Protection Qty Limit	Y	Specify the limit of the total traded contracts per underlying within the exposure time interval when market maker protection is triggered. When this value is met or exceeded the system automatically removes the quotes for the instruments connected to the underlying. A value of 0 means that no quantity protection exists.
20111	Protection Delta Limit	Y	Specify the limit of the delta value per underlying within the exposure time interval when market maker protection is triggered. When this value is met or exceeded the system automatically removes the quotes for the instruments connected to the underlying. A value of 0 means that no delta protection exists.
20112	Protection Frozen Limit	Y	Specify the time interval in seconds when quotes are rejected after market maker protection has been triggered.
20113	Protection Exposure Limit	Y	Specify the rolling time interval in seconds used in quantity/delta protection calculations.
20115	Protection Include Futures	Y	Selected if futures and forwards are to be included in the delta calculation. Valid values: 1 = Include 2 = Do not Include
	Standard Trailer	Y	

10.10.13 MMProtection Set Limit - Reject (out)

TAG	FIX TAG NAME	REQ'D	COMMENT
	Standard Header	Y	MsgType = MR
20114	ProtectionRequestID	Q	From MMProtection
309	UnderlyingSecurityID	Y	As Commodity Code
311	UnderlyingSymbol	Y	Should be set to NONE
58	Text	Y	Reject Reason
	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:
The marketplace can allow trades to be reported using a set of different mechanisms, the mechanisms currently supported over FIX are:

Two-Party Reports
Used when one of the parties or an authorized third party 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.

Multileg Reports
Used to report up to twelve trades in different instruments in one transaction.
NOTE: all legs in a multi-leg transaction will be validated in the PTRM tool before a deal is approved.

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. 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. 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 System order number.

11.2.4 Timestamps

11.2.4.1 Time of Agreement

Time of agreement is shown by the TransBkdTime (483) field. This should be a UTC Timestamp

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 (two-party or multileg)
- To update a Trade Report (not supported in this solution)
- To cancel a Trade Report (not supported in this solution)
- For the marketplace to publish trade confirmations (see chapter 0)
- For the marketplace to publish updates to previous trade confirmations (not supported in this solution)
- To cancel a confirmed trade (not supported in this solution)
- For the marketplace to notify the contra party when a one-party report has been sent in. (not supported in this solution)

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 *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. Multileg Trade Reports

A multileg trade report can contain up to 12 legs. For each leg the instrument, price, quantity, AccountCode and the two parties involved must be set.

The multileg trade report message is a custom message (MsgType=UF) that mimics the standard Trade Capture Report, but adds the ability to include multiple legs, where each leg can have different parties and other parameters.
The system will respond with Trade Capture Report Acks, following the procedure described in section 11.3.1.2. If the multileg report is acknowledged, one trade confirmation message will be sent for each leg to both parties.

NOTE: The Trade Capture Report Ack will not contain Account or AllocID in response to a Multileg Trade Report.

11.3.1.2 Trade Reports for Strategy instruments

For trades reported in strategy instruments, all messaging returned from the marketplace after submitting the initial report, will be per leg (outright). You will receive Trade Capture Report Ack messages per leg (outright) the strategy is made up of. Trade confirmations are also sent per leg (outright). See section 11.4.2 for an example of a trade report message flow for a strategy instrument.

11.3.1.3 Update a Reported Trade

Request to modify a reported trade is not supported by the system.

11.3.1.4 Trade Types

The TrdType tag (828) is used to specify the type of trade being reported to the marketplace.
The following values are supported:

- 1 = Block Trade
- 2 = Exchange for Physical (EFP)
- 11 = Exchange for Risk (EFR)
- 14 = Exchange of Options for Options (EOO)

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.

11.3.1.5 Marketplace publication of Confirmed Trades

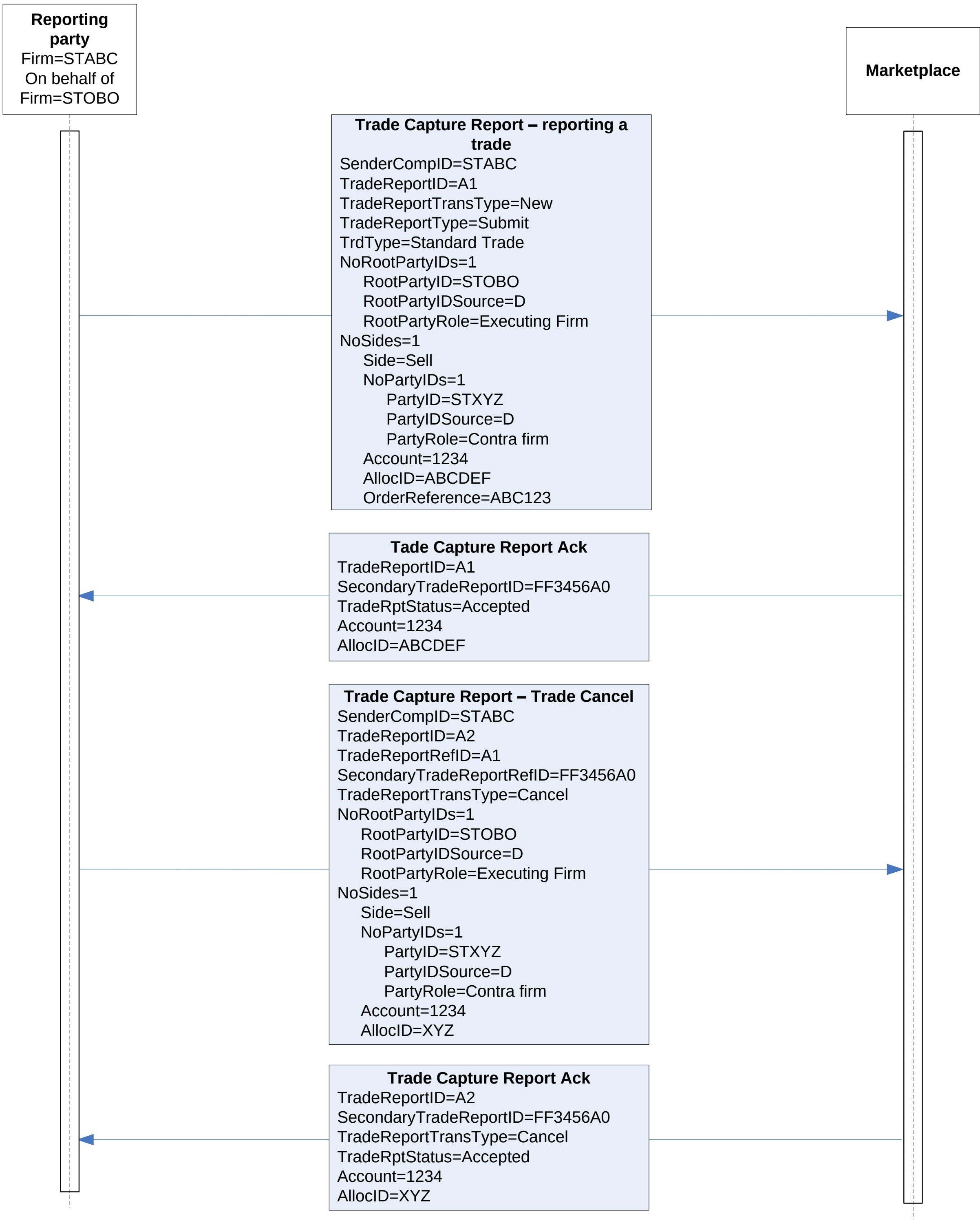
The marketplace uses the TCR to publish confirmed trades, whether auto-matched or reported by clients. See chapter 0 for details.

11.3.2 Trade Capture Report Acknowledgement

The TCR Ack is used to respond to a Trade Capture Report submitted to the exchange. You will always receive at least one Trace Capture Report Ack when reporting a trade.
NOTE: For strategy instruments, you will receive one TCR Ack per leg (outright) the strategy consists of.

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 SP2 [2], Volume 5, Appendix B. They have been modified for this solution.



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

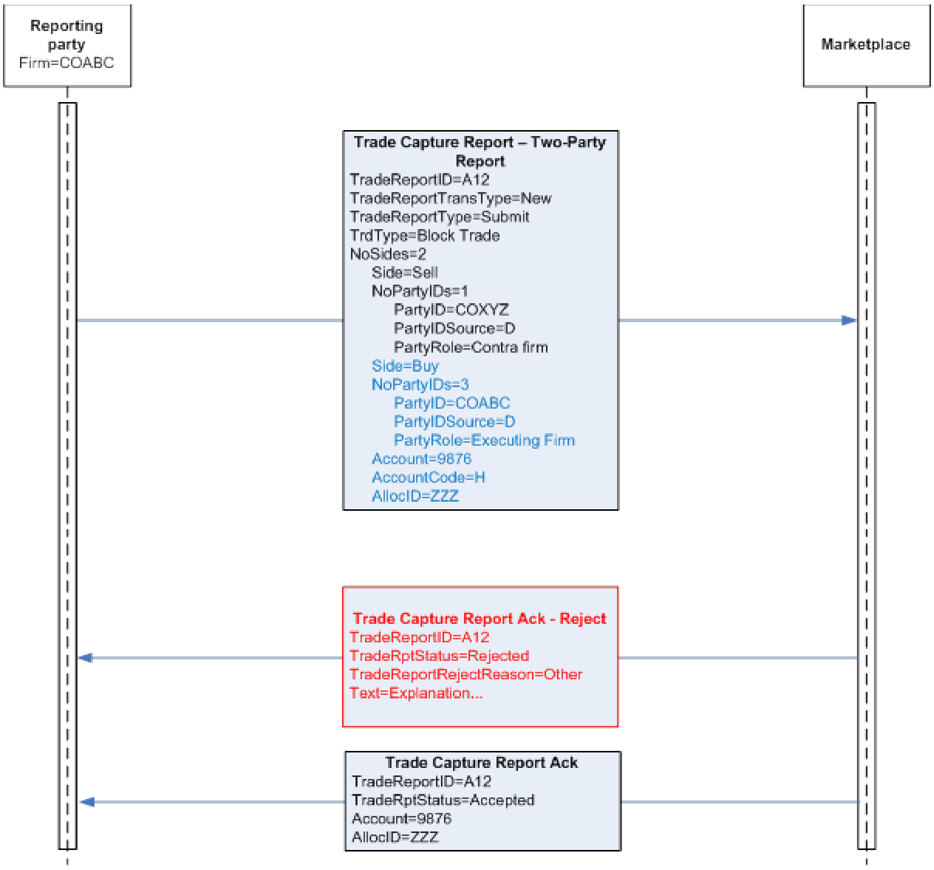
```
SenderCompID=STABC
TradeReportID=A2
TradeReportRefID=A1
SecondaryTradeReportRefID=FF3456A0
TradeReportTransType=Cancel
NoRootPartyIDs=1
    RootPartyID=STOBO
    RootPartyIDSource=D
    RootPartyRole=Executing Firm
NoSides=1
    Side=Sell
    NoPartyIDs=1
        PartyID=STXYZ
        PartyIDSource=D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

Trade Capture Report Ack

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

11.4.1 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 0 for details.



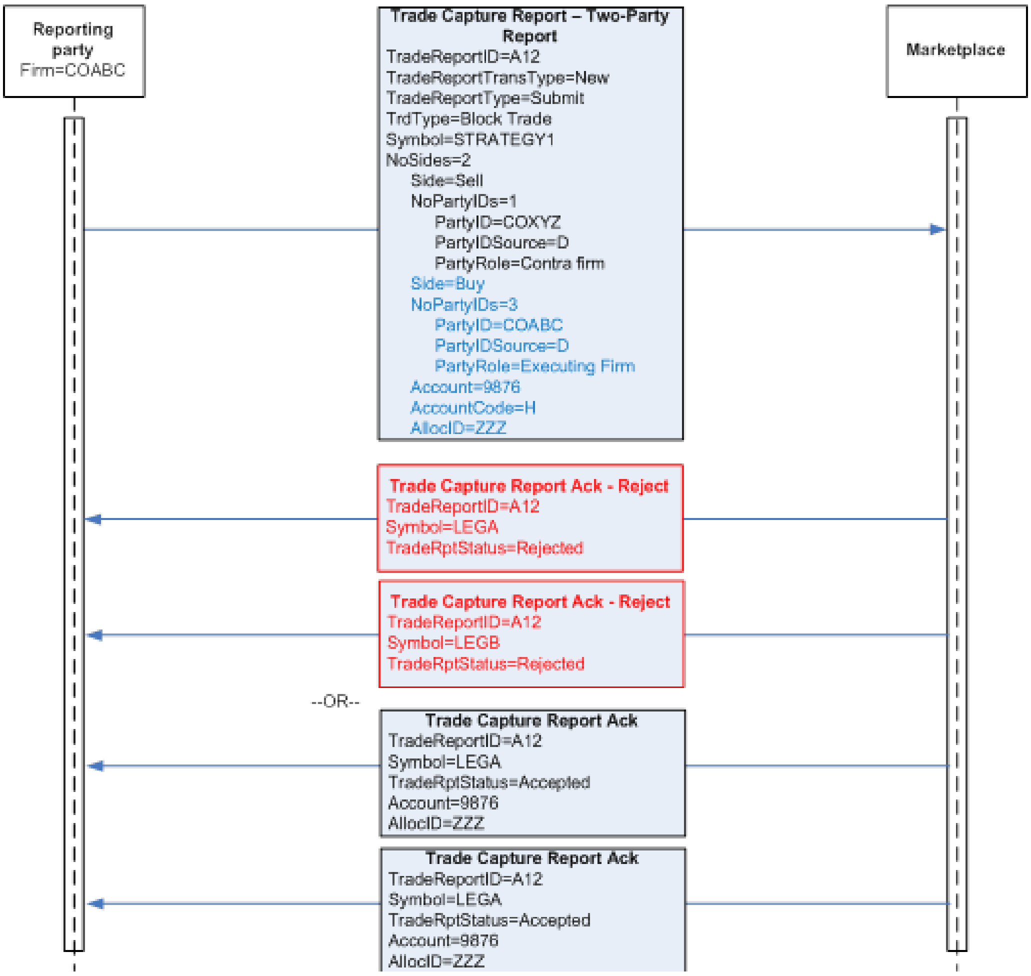
Trade Capture Report – Two-Party Report
TradeReportID=A12
TradeReportTransType=New
TradeReportType=Submit
TrdType=Block Trade
NoSides=2
    Side=Sell
    NoPartyIDs=1
        PartyID=COXYZ
        PartyIDSource=D
        PartyRole=Contra firm
    Side=Buy
    NoPartyIDs=3
        PartyID=COABC
        PartyIDSource=D
        PartyRole=Executing Firm
    Account=9876
    AccountCode=H
    AllocID=ZZZ

Trade Capture Report Ack - Reject
TradeReportID=A12
TradeRptStatus=Rejected
TradeReportRejectReason=Other
Text=Explanation...

Trade Capture Report Ack
TradeReportID=A12
TradeRptStatus=Accepted
Account=9876
AllocID=ZZZ

11.4.2 Two-party Strategy instrument Trade Report

In this scenario a two-party Trade Report in a strategy instrument, STRATEGY1 is sent into the system. STRATEGY1 consists of two legs (outrights), LEGA and LEGB. This scenario illustrates that a client will receive *one Trade Capture Report Ack per leg (outright)*.



Trade Capture Report – Two-Party Report
TradeReportID=A12
TradeReportTransType=New
TradeReportType=Submit
TrdType=Block Trade
Symbol=STRATEGY1
NoSides=2
    Side=Sell
    NoPartyIDs=1
        PartyID=COXYZ
        PartyIDSource=D
        PartyRole=Contra firm
    Side=Buy
    NoPartyIDs=3
        PartyID=COABC
        PartyIDSource=D
        PartyRole=Executing Firm
    Account=9876
    AccountCode=H
    AllocID=ZZZ

Trade Capture Report Ack - Reject
TradeReportID=A12
Symbol=LEGA
TradeRptStatus=Rejected

Trade Capture Report Ack - Reject
TradeReportID=A12
Symbol=LEGB
TradeRptStatus=Rejected

Trade Capture Report Ack
TradeReportID=A12
Symbol=LEGA
TradeRptStatus=Accepted
Account=9876
AllocID=ZZZ

Trade Capture Report Ack
TradeReportID=A12
Symbol=LEGA
TradeRptStatus=Accepted
Account=9876
AllocID=ZZZ

11.5 MESSAGE DETAILS

11.5.1 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	Valid values: 1 = Block Trade 2 = Exchange for Physical 11 = EFR 14 = EOO
570	PreviouslyReported			Y	Indicates if the trade capture report was previously reported to the counterparty Valid values: N = No
55	Instrument/Symbol				Instrument 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
20016	Future Reference Price				Passthrough field for Options Trades, contain a reference price for the futures reference leg.
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	ID of Party
→	→	447	PartyIDSource	Q	Valid values : D = Propr. Code
→	→	452	PartyRole	Q	Valid values: 1 = Executing Firm 11 = Order Originating Trader 17 = Contra Firm 14 = Give-Up Clearing Firm
→	528	OrderCapacity			The type of business conducted. Valid values: A = Agency P = Principal

→	582	CustOrderCapacity		Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
→	483	TransBkdTime	Q	Time of agreement as a UTC timestamp
→	1	Account	Q	Mandatory account identifier set by client
→	70	AllocID		Optional pass-thru field set by client and echoed back by marketplace.

→	77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
				C = Close
				O = Open

	Standard Trailer		Y	

11.5.2 Trade Report – Multileg (in)

TAG	FIX TAG NAME				REQ'D	COMMENT
	Standard Header				Q	MsgType = UF
571	TradeReportID				Q	Client-generated identifier
828	TrdType				Q	Valid values: 1 = Block Trade 2 = EFP 11 = EFR 14 = EOO
555	NoLegs				Q	Number of Legs (max 12 are allowed)
→	600	LegSymbol			Q	Leg Instrument short name. Either LegSymbol or LegSecurityID+LegSecurityIDSource must be set. If LegSecurityID is set, LegSymbol must be set to "[N/A]", since it's the first field in the repeating group.
→	602	LegSecurityID				Leg Orderbook ID
→	603	LegSecurityIDSource				Valid values: M = Marketplace-assigned identifier
→	637	LegLastPx			Q	Leg trade price
→	1418	LegLastQty			Q	Leg trade quantity
→	552	NoSides			Q	Must be set to 2 (both sides for each leg)
→	→	54	Side		Q	Valid values: 1 = Buy 2 = Sell
→	→	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 11 = Order Originating Trader 14 = Give-Up Clearing Firm 17 = Contra Firm

→	→	528	OrderCapacity		The type of business conducted. Valid values: A = Agency P = Principal
→	→	582	CustOrderCapacity		Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
→	→	1	Account	Q	Mandatory account identifier set by client
→	→	70	AllocID		Optional pass-thru field set by client and echoed back by marketplace
→	→	77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values: C = Close O = Open
483	TransBkdTime			Q	Time of agreement as a UTC timestamp
60	TransactTime			Q	Time of execution/order creation as a UTC timestamp
	Standard Trailer			Q	

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		System order number.
939	TradeRptStatus	Q	Valid values: 0 = Accepted 101 = Holding (NFX Extension)
55	Instrument/Symbol	Q	Instrument short name. **NOTE:** Will be set to "[N/A]" for multileg trade reports.
48	Instrument/SecurityID		Orderbook ID. **NOTE:** Will be set on any non-multileg trade report acks.
22	Instrument/SecurityIDSource		Valid values: M = Marketplace-assigned identifier **NOTE:** Will be set on any non-multileg trade report acks.
70	AllocID		Optional pass-thru field set by client and echoed back by marketplace.
1	Account		Optional pass-thru field set by client,

77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
			C = Close
			O = Open

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
818	SecondaryTradeReportID		System order number.
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	

12 TRADE 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 previously 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.

12.1 TRADE CONFIRMATION FEATURES

12.1.1 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 12.4.1 and 12.4.2 respectively for message details.

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

12.1.2 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 12.4.3 for message details.

12.1.3 Timestamps

12.1.3.1 Time of Agreement

Time of agreement is shown by the TransBkdTime (483) field. This should be a UTC Timestamp

12.1.3.2 Time of Execution

Time of Execution is shown by the TransactTime (60) field. This should be a UTC Timestamp

12.1.4 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.

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.
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 confirmations to reported trades.

12.2.3 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 System order number.
Secondary Trade Report ID is also set in confirmations.

12.2.4 Trade Match ID

The TrdMatchID (880) contains the match id generated by the system. TrdMatchID will hold the 64 bit binary match id encoded as a 16 byte hex string.

12.2.5 Trade ID

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

12.2.6 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.

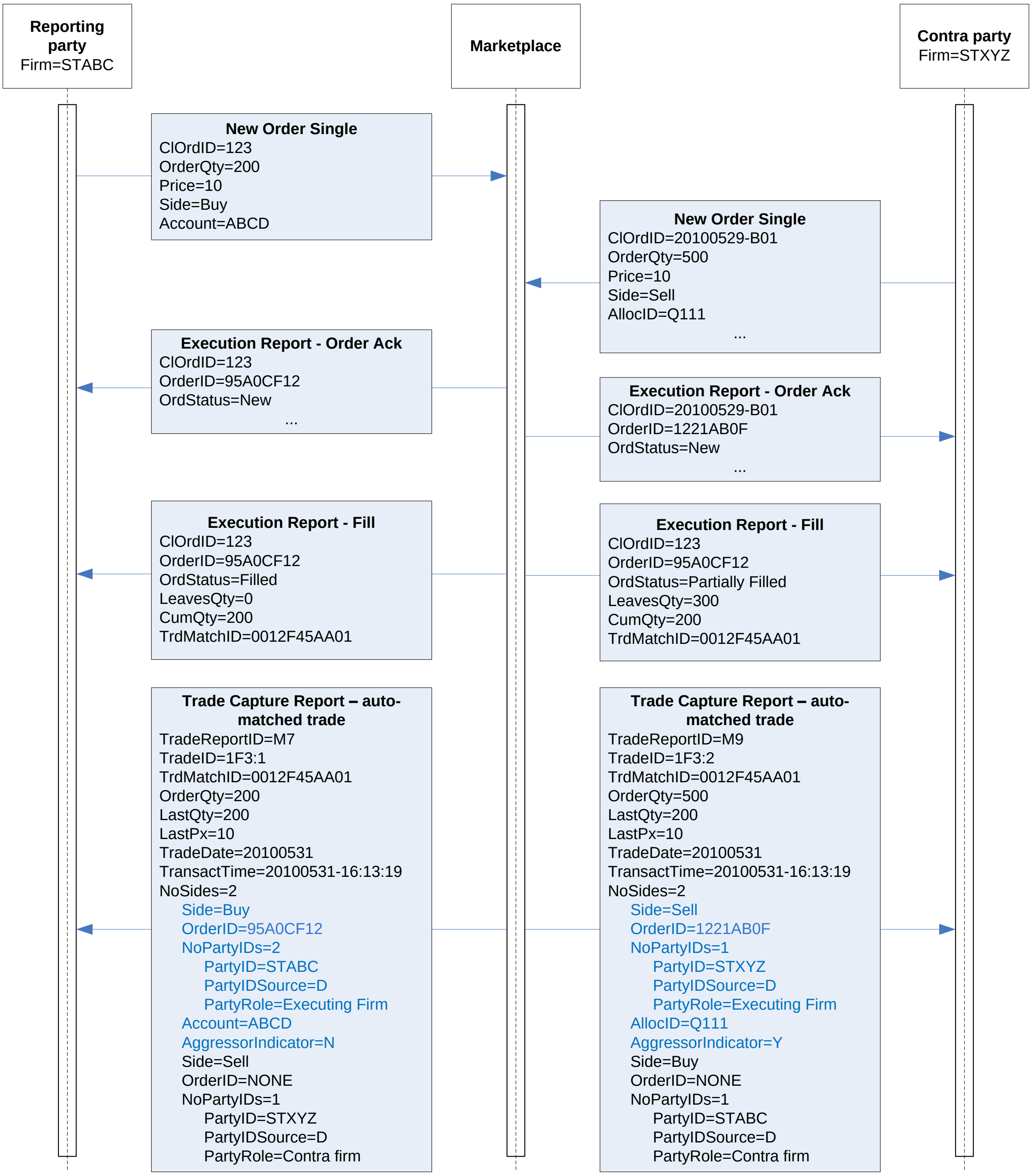
12.3 WORKFLOWS

12.3.1 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 an auto-matched trade, the counterparty will not be shown.



New Order Single
ClOrdID=123
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=0012F45AA01

Execution Report - Fill
ClOrdID=123
OrderID=95A0CF12
OrdStatus=Partially Filled
LeavesQty=300
CumQty=200
TrdMatchID=0012F45AA01

Trade Capture Report – auto-matched trade
TradeReportID=M7
TradeID=1F3:1
TrdMatchID=0012F45AA01
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
    Account=ABCD
    AggressorIndicator=N
    Side=Sell
    OrderID=NONE
    NoPartyIDs=1
        PartyID=STXYZ
        PartyIDSource=D
        PartyRole=Contra firm

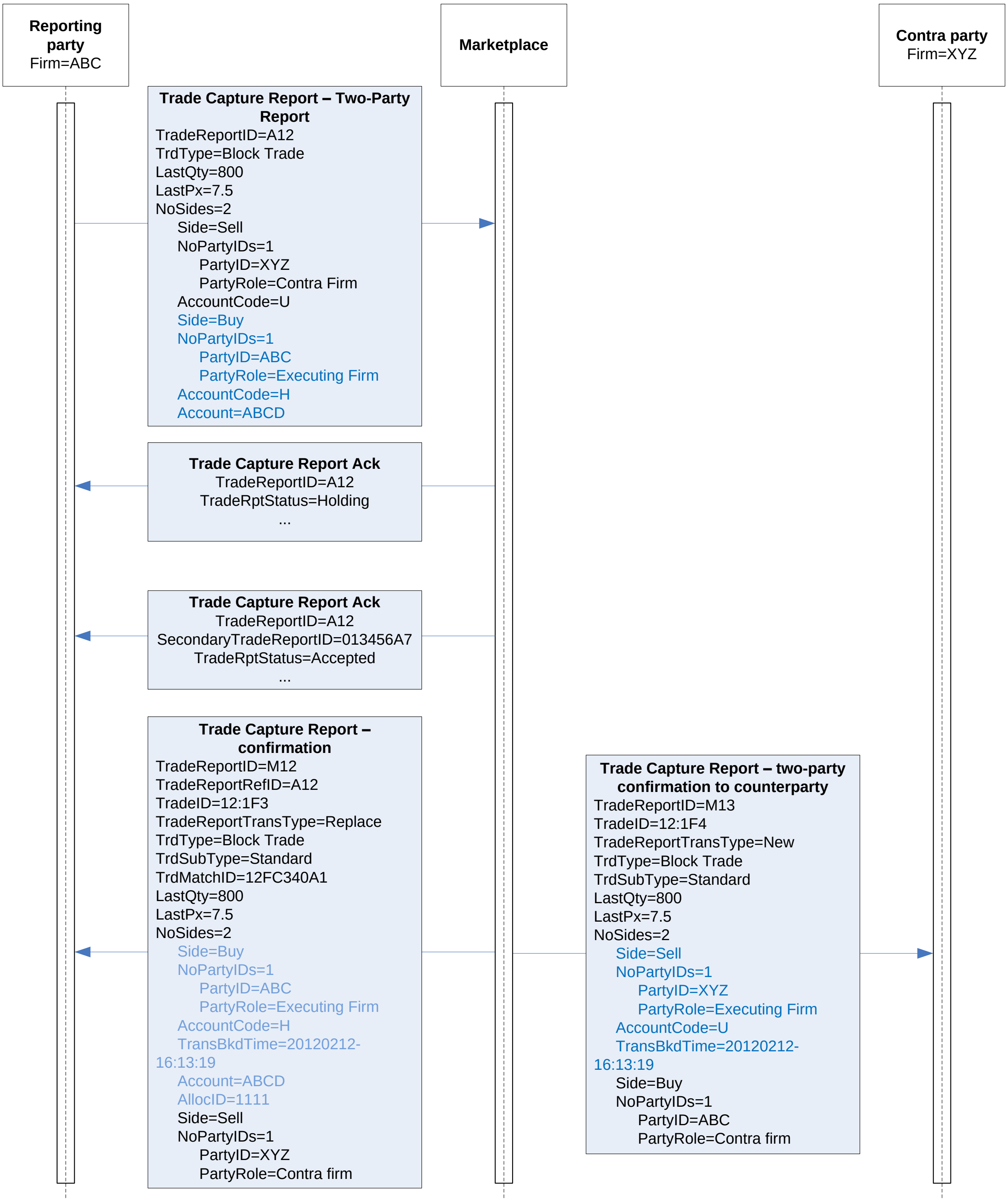
Trade Capture Report – auto-matched trade
TradeReportID=M9
TradeID=1F3:2
TrdMatchID=0012F45AA01
OrderQty=500
LastQty=200
LastPx=10
TradeDate=20100531
TransactTime=20100531-16:13:19
NoSides=2
    Side=Sell
    OrderID=1221AB0F
    NoPartyIDs=1
        PartyID=STXYZ
        PartyIDSource=D
        PartyRole=Executing Firm
    AllocID=Q111
    AggressorIndicator=Y
    Side=Buy
    OrderID=NONE
    NoPartyIDs=1
        PartyID=STABC
        PartyIDSource=D
        PartyRole=Contra firm

12.3.2 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
TradeReportID=A12
TrdType=Block Trade
LastQty=800
LastPx=7.5
NoSides=2
Side=Sell
NoPartyIDs=1
PartyID=XYZ
PartyRole=Contra Firm
AccountCode=U
Side=Buy
NoPartyIDs=1
PartyID=ABC
PartyRole=Executing Firm
AccountCode=H
Account=ABCD

Trade Capture Report Ack
TradeReportID=A12
TradeRptStatus=Holding
...

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

Trade Capture Report – confirmation
TradeReportID=M12
TradeReportRefID=A12
TradeID=12:1F3
TradeReportTransType=Replace
TrdType=Block Trade
TrdSubType=Standard
TrdMatchID=12FC340A1
LastQty=800
LastPx=7.5
NoSides=2
Side=Buy
NoPartyIDs=1
PartyID=ABC
PartyRole=Executing Firm
AccountCode=H
TransBkdTime=20120212-16:13:19
Account=ABCD
AllocID=1111
Side=Sell
NoPartyIDs=1
PartyID=XYZ
PartyRole=Contra firm

Trade Capture Report – two-party confirmation to counterparty
TradeReportID=M13
TradeID=12:1F4
TradeReportTransType=New
TrdType=Block Trade
TrdSubType=Standard
LastQty=800
LastPx=7.5
NoSides=2
Side=Sell
NoPartyIDs=1
PartyID=XYZ
PartyRole=Executing Firm
AccountCode=U
TransBkdTime=20120212-16:13:19
Side=Buy
NoPartyIDs=1
PartyID=ABC
PartyRole=Contra firm

12.4 MESSAGE DETAILS

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

TAG	FIX TAG NAME	REQ’D	COMMENT
	Standard Header	Y	MsgType = AE
20034	ComboGroupID		NFX Extension: Can be used to group trades in the legs of a strategy (will be the same for all legs).
571	TradeReportID	Y	Identifier assigned by marketplace
1003	TradeID	Q	Unique identifier for trade

1040	SecondaryTradeID		Trade id assigned by external system.
1126	OrigTradeID		Used to refer to original trade in case of modifications
1127	OrigSecondaryTradeID		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. NFX 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 = Superseding 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	Instrument 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.
715	ClearingBusinessDate		
60	TransactTime	Y	**NOTE:** Contains Time of Trade Execution
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 11 = Order Originating Trader 12 = Executing Trader 17 = Contra Firm 14 = Give-Up Clearing Firm
→	528	OrderCapacity			The type of business conducted, will not be included in trades originating from mass quotes. Valid values: A = Agency P = Principal
→	582	CustOrderCapacity			Capacity of customer placing the order, will not be included in trades originating from mass quotes. Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
→	1	Account			Mandatory account identifier set by client, will not be included in trades originating from mass quotes
→	70	AllocID			Optional pass-thru field set by client and echoed back by marketplace.
→	77	PositionEffect			*NFX Extension.* Defines the requested position update for the account. Valid values: C = Close O = Open
→	1057	AggressorIndicator			Indicates who is the aggressive party in the trade. Valid values: Y = Party is the aggressor N = Party is passive
855	SecondaryTrdType				Contains system deal source value.
797	CopyMsgIndicator				Set to ‘Y’ on Drop Copy messages
	Standard Trailer			Y	

12.4.2 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	Unique identifier for trade
1040	SecondaryTradeID		Trade id assigned by external system.
1126	OrigTradeID		Used to refer to original trade in case of modifications
1127	OrigSecondaryTradeID		Original trade id assigned by external system.
572	TradeReportRefID	Q	From inbound TCR
818	SecondaryTradeReportID	Q	System 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		Valid values: 1 = Block Trade 2 = EFP 11 = EFR 14 = EOO
829	TrdSubType	Q	Further qualification of the trade type. NFX 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 = Superseding 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.
20034	ComboGroupID		NFX Extension: Can be used to group trades in the legs of a strategy (will be the same for all legs).

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
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 11 = Order Originating Trader 12 = Executing Trader 17 = Contra Firm 14 = Give-Up Clearing Firm
→	528	OrderCapacity		Q	The type of business conducted. Valid values: A = Agency P = Principal
→	582	CustOrderCapacity		Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
→	483	TransBkdTime			Time of agreement as a UTC timestamp
→	1	Account		Q	Mandatory account identifier set by client. Only set on the own Side
→	70	AllocID			Optional pass-thru field set by client and echoed back by marketplace.
→	77	PositionEffect			*NFX Extension.* Defines the requested position update for the account. Valid values: C = Close O = Open
715	ClearingBusinessDate				

855	SecondaryTrdType		Contains System deal source value.
797	CopyMsgIndicator		Set to ‘Y’ on Drop Copy messages
	Standard Trailer	Y	

12.4.3 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	Unique identifier for trade
1040	SecondaryTradeID		Trade id assigned by external system.
1126	OrigTradeID		Used to refer to original trade in case of modifications
1127	OrigSecondaryTradeID		Original trade id assigned by external system.
818	SecondaryTradeReportID	Q	System order number.
487	TradeReportTransType	Q	Valid values: 0 = New 2 = Replace 4 = Reverse
856	TradeReportType	Q	Valid values: 0 = Submit
828	TrdType		Valid values: 1 = Block Trade 2 = EFP 11 = EFR 14 = EOO
829	TrdSubType	Q	Further qualification of the trade type. NFX 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 = Superseding 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.
20034	ComboGroupID				NFX Extension: Can be used to group trades in the legs of a strategy (will be the same for all legs).
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
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 11 = Order Originating Trader 12 = Executing Trader 17 = Contra Firm 14 = Give-Up Clearing Firm
→	528	OrderCapacity		Q	The type of business conducted. Valid values: A = Agency P = Principal
→	582	CustOrderCapacity		Q	Capacity of customer placing the order Valid Values: 1 = Member trading for their own account 2 = Clearing Firm trading for its proprietary account 3 = Member trading for another member 4 = Other
→	483	TransBkdTime			Time of agreement as a UTC timestamp
→	1	Account		Q	Mandatory account identifier set by client

→ 70	AllocID		Optional pass-thru field set by client and echoed back by marketplace.

→ 77	PositionEffect		*NFX Extension.* Defines the requested position update for the account. Valid values:
			C = Close
			O = Open

715	ClearingBusinessDate		
855	SecondaryTrdType		Contains System deal source value.
797	CopyMsgIndicator		Set to 'Y' on Drop Copy messages
	Standard Trailer	Y	

13 APPENDIX A, NFX EXTENSIONS

This chapter details how this solution deviates from standard FIX 5.0 SP2. 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.

There are five types of deviations from the standard:

- Messages added. All current additions come from the later standard versions of FIX.
- Fields added. A few user defined fields had to be added to accommodate back-end functionality not present in FIX 5.0 SP2.
- Enumerated values added. Some fields have added enums.
- Removed fields required in standard FIX.
- Other datatype used for existing field.

13.1 ADDED MESSAGES

The following messages not present in standard FIX 5.0 SP2 have been added to this specification:

MESSAGE NAME	IN FIX 5.0 SP2	COMMENT
Multileg Trade Report	N	MsgType=UF

13.2 ADDED FIELDS

TAG NUM	FIELD NAME	IN FIX 5.0 SP2	COMMENT
70	AllocID	Y	Existing FIX tags added to the Execution Report messages.
77	PositionEffect	Y	Existing FIX tag added to the Trade Capture Report Ack message
20034	ComboGroupID	N	Can be used to group trades in the legs of a strategy (will be the same for all legs). Data type: string
20200	LegTrdMatchID	N	The match id for the given leg in a strategy execution. Data type: string
20017	OriginatingTrader	N	Used to report the originating trader id when creating a tailor made combination (TMC). Data type: string

13.3 ADDED ENUMERATIONS

ENUMERATION	ADDED TO FIELD	IN FIX 5.0 SP2	COMMENT
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 = Superseding 1016 = State change 1017 = Giveup	TrdSubType	N	

1018 = Takeup			
101 = NFX Trading System series definition (NFX 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	
101 = Holding	TradeRptStatus	N	

13.4 REMOVED REQUIRED FIELDS

TAG NUM	FIELD NAME	IN MESSAGE	COMMENT
54	Side	Execution Report – Order Reject	

14 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	
320	SecurityRequestID	20	
70	AllocID	15	Existing FIX tags added to the Order Cancel Request and the Execution Report messages.
1	Account	10	
448	PartyID	5	When PartyRole=Executing Firm ,Contra Firm or Client
571	TradeReportID	20	
572	TradeReportRefID	20	
881	SecondaryTradeReportRefID	20	
923	UserRequestID	20	
925	NewPassword	32	

15 REVISION HISTORY

DATE	REVISION	CHANGE DESCRIPTION
July 04, 2014	1.0	Initial version. Added CustOrderCapacity (582) Added CustomerOrFirm (204) Added new values to Account Type (20015) Added MaxFloor(111) Added single sided trade report Added linked order Added Cross Transaction
July 17, 2014	1.01	Removed one sided trade report, Removed Entering and Executing special tags on trade capture report, both

		sides should fill same information, no less should be filled for contra firm party role. Added party role Client
September 29, 2014	1.02	Added party role information to handle give-ups Account (1) was changed from optional to mandatory
December 4, 2012	1.03	Added PositionEffect (77) Multileg Reports now support up to 12 (from 6) trades in different instruments in one transaction. NOTE: all legs in a multi-leg transaction will be validated in the PTRM tool before a deal is approved.
December 14, 2014	1.04	Added Transactions: New Order Cross MMProtection (Market Maker Protection Limits) MMProtection Reject Mass Quote Acknowledgement – All Quotes Accepted or Canceled (out) CustOrderCapacity (582) was changed from optional to mandatory CustomerOrFirm (204) was changed from optional to mandatory
January 23, 2015	1.05	Added Transaction: Quote Cancel
February 04, 2015	1.06	Removed AccountCode (20015) Replaced CustomerOrFirm (204) with OrderCapacity (528)
February 16, 2015	1.07	Minor corrections
March 10, 2015	1.08	Remove party role 83 = Clearing Account Added MaxFloor (111) to New Order Sinlge message Minor corrections
April 14, 2015	1.09	Changes to Quote Cancel (in) message, removed Executing Trader and on behalf of text for ordersadded OrderOriginating Trader and Give-up clearing firm to some messages. TAG 483 is now reuired for Two Party Trade report (in). Changed TrdType EFS(12) to EFR(11).
April 22, 2015	1.10	Removed Client ID, added Order Originating Trader to trade capture reports and tag 1 is no longer required for Order Cancel and Order Cancel Replace Request. Changed the description of section 5.5 USERS ACROSS MULTIPLE SESSIONS.
May 20, 2015	1.11	Made the Partygroup required for New Order Single, Order Cancel Replace and New Order List since Order originating trader is required. Added tag 301 to mass quote ack, changed MMP reject and clarified the description for tag 483.
June 2, 2015	1.12	Updated Quote Cancel responses and some other minor corrections.
Jun 5, 2015	1.13	Minor corrections.
June 30, 2015	1.14	Partyrole clarification in New Order Cross. Fixed error in party roles for New Order List. LastPx is no longer required for combo fills. Updated the disclaimer.
July 7, 2015	1.15	Minor corrections.
September 30, 2015	1.16	Added new MPID information to the TradeCaptureReport for drop processes. Added new Nasdaq disclaimer.
October 15, 2015	1.17	Updated Heartbeat Intervals.
March 4, 2016	1.18	Added information about quoting in Combinations.
June 1, 2016	1.19	Update Execution Report – Unsolicited Order Update (out) message
June 3, 2016	1.20	Added detailed description of Execution Report – Unsolicited Cancel messages field Text
July 5, 2016	1.21	Added tag 60 as a required field of New Order Cross
August 26, 2016	1.22	Changed description of header tag 57 (TargetSubID)
October 20, 2016	1.23	Adding QuoteID (117) in New Order Cross Adding Quote as response to Quote Request Adding Order Originating Trader (1116-1119) to Quote Request

November 16, 2016	1.24	Copyright and disclaimer change
November 18, 2016	1.25	Set tag 528 as a required field for legs of a List Order message.
December 02, 2016	1.26	Added PositionEffect (77) to the list of amendable fields.
December 22, 2016	1.27	Added OriginatingTrader (20017) to the SecurityDefinitionRequest message.
February 08, 2017	1.28	Updated Disclaimer and Mass Quote in Combinations.
April 25, 2017	1.29	Removed NewOrderCross and updated Request for Quote.
November 08, 2017	1.30	Added NoLegs(555), LegSymbol(600), LegSecurityID(602), LegSecurityIDSource(603), LegRatioQty(623), LegSide(624) to SecurityDefinitionUpdateReport. Added SecuritySubType(762), LegLastPx(637) and LegOptionRatio(1017) to SecurityDefinitionRequest, SecurityDefinitionUpdateReport.
November 13, 2017	1.31	LegQuantityFuture(20018) to SecurityDefinitionUpdateReport

Nasdaq