

Refinitiv FXall

FXall FX Trading FIX API

Resting, Algos and Fixing Orders

Options RFQ

Cash RFQ (replacing the proprietary FXall Java TCPI API)

Document Version: 5.1
Date of issue: 28 March 2022

An LSEG Business

© Refinitiv 2021. All rights reserved.

Refinitiv does not guarantee that any information contained in this document is and will remain accurate or that use of the information will ensure correct and faultless operation of the relevant service or equipment. Refinitiv, its agents and employees, accepts no liability for any loss or damage resulting from reliance on the information contained in this document.

This document contains information proprietary to Refinitiv and may not be reproduced, disclosed, or used in whole or part without the express written permission of Refinitiv.

Any software, including but not limited to, the code, screen, structure, sequence, and organization thereof, and documentation are protected by national copyright laws and international treaty provisions. This document is subject to U.S. and other national export regulations.

Nothing in this document is intended, nor does it, alter the legal obligations, responsibilities or relationship between yourself and Refinitiv as set out in the contract existing between us.

Contents

About this document.....	6
Chapter 1 FIX Session and Administrative messages.....	7
Protocol Support.....	7
Transport Support	7
Session Schedule.....	7
Sequence Numbers.....	8
Session Level	8
Standard FIX Header	9
Standard FIX Trailer	10
Checksum Calculation	10
Session Level Messages.....	11
Logon (A).....	11
Logout (5)	11
Heartbeat (0)	11
TestRequest (1).....	12
ResendRequest (2)	12
SequenceReset (4)	12
Reject (3).....	13
BusinessMessageReject (j).....	14
Chapter 2 Resting, Algos and Fixing Orders.....	15
Orders - Initiator Messages	15
Standard FIX Header – Additional Fields	15
NewOrderSingle (D).....	15
NewOrderMultiLeg (AB)	20
OrderCancelReplaceRequest (G)	27
OrderCancelRequest (F)	33
OrderStatusRequest (H).....	34
DontKnowTrade (Q)	35
Execution Acknowledgement (BN).....	36
Orders - Acceptor Messages.....	37
OrderCancelReplaceReject (9)	37
ExecutionReport (8)	38
Chapter 3 Options - RFQ	43
RFQ Flow Diagram.....	43
Options – Initiator Messages.....	43
Standard FIX Header – Additional Fields	43

Quote Request (R)	44
Quote Cancel (Z)	51
Quote Acknowledgement (b)	51
MarketDataRequest (V)	52
NewOrderMultiLeg (AB)	53
ExecutionReport (8)	61
DontKnowTrade (Q)	66
Options - Acceptor Messages	67
Quote (S)	67
Quote Cancel (Z)	69
Quote Status Report (AI)	69
Quote Request Reject (AG)	70
Market Data Snapshot Full Refresh (W)	71
Market Data Snapshot Increment Refresh (R)	72
Market Data Request Reject (Y)	73
ExecutionReport (8)	73
Chapter 4 QuickTrade – Cash RFQ	79
RFQ Flow Diagram	79
QuickTrade – Initiator Messages	79
Quote Request (R)	79
Quote Response (AJ)	86
Quote Cancel (Z)	87
NewOrderSingle (D) – Single Leg Order	88
NewOrderMultileg (AB) – Multi-Leg Order	93
Execution Acknowledgement (BN)	99
OrderStatusRequest (H)	100
DontKnowTrade (Q)	101
QuickTrade – Acceptor Messages	101
QuoteRequestReject (AG)	101
Quote Cancel (Z)	102
Quote (S)	103
Execution Report (8)	108
Appendix 113	
QuickTrade	113
Security Types	113
Request for Quote States	113
Deprecated Features and Tags	114
Options 115	
Supported values for Expiry Cut	115

Regulatory	116
Supported values for NoPartyIDs.....	116
Additional Supported Workflows	117
Forward First RFQ.....	117

About this document

IN THIS DOCUMENT

This document describes the implementation of the Financial Information eXchange (FIX) protocol to trade Foreign Exchange on FXall:

- Resting, Algos and Fixing Orders
- Options RFQ
- Cash RFQ (replacing the proprietary FXall Java API TCPI)
- Post Trade Amendments

The intended readership of this document is for liquidity providers who wish to trade on FXall via FIX.

The following sections will be covered in this document:

- FIX Session and Administrative messages
- For each of the above FX trading workflows:
 - Supported business messages
 - Field descriptions
 - Sample message types

FXall use FIX Engines that support the FIX protocol version 5.0 SP2.

Previous FIX versions might be supported, please contact the global Refinitiv help desk for more details.

For further information about the FIX protocol please visit <https://www.fixtrading.org/>

The FIX 5.0 SP2 data dictionary https://www.onixs.biz/fix-dictionary/5.0.sp2/fields_by_tag.html can be used as a point of reference.

FOR FURTHER INFORMATION

For questions or further information related to FXall, please contact the global Refinitiv help desk using the link below.

[MyRefinitiv – Contact Us](#)

Chapter 1 FIX Session and Administrative messages

The [Initiator](#) is responsible for establishing the telecommunications link to the [Acceptor](#) and initiating the session by sending the initial **Logon** message.

The [Acceptor](#) is the receiving party of the connection. The [Acceptor](#) is responsible for performing the logon authentication and formally declaring the connection request as “accepted” by sending an acknowledgment **Logon** message to the [Initiator](#) of the session when authentication is complete.

FXall support both Outbound FIX connections (FXall is the [Initiator](#)) and Inbound FIX connections (Provider is the [Initiator](#), except on **Resting Orders and Algos where FXall only support Outbound FIX connections**).

It is assumed that the Initiator will be responsible for implementing the Initiator side of the specification and the Acceptor will be responsible for implementing the Acceptor side of the specification. The Initiator will support all of the Acceptors messages and the Acceptor will support all of the Initiators messages.

PROTOCOL SUPPORT

FXall supports the following versions of the FIX protocol for connectivity. The preferred implementation is FIX 5.0 SP2.

Protocol Version	Supported by FXALL
FIX 4.2	Y
FIX 4.3	Y
FIX 4.4	Y
FIX 5.0 SP2	Y

TRANSPORT SUPPORT

Environment	Network	Source IP	Connection Transport
Testing (INT)	Internet	75.98.57.4	TCP/IP
Production	Internet	209.191.134.245	TCP/IP
	Co-Location	209.191.134.242	TCP/IP

FXall supports two environments: User Acceptance Testing (INT) and Production. After network connectivity has been established, the Provider will receive FIX session and COMPID logon information.

Providers must have an UAT environment they can use for functional testing, which must be connected to the FXall UAT INT environment. Such UAT network connectivity is preferably done be over the internet.

SESSION SCHEDULE

FXall sessions run Sunday to Friday, commencing on Sunday at 17:30 EST and terminating on Friday at 17:00 EST.

FXall disconnects the FIX sessions daily around 5:00 EST, and reconnects after a configurable timespan agreed with each providers in accordance with their daily maintenance window.

SEQUENCE NUMBERS

Messages are identified by a unique sequence number and incremented throughout a session. Monitoring the sequence numbers enable both parties to identify and take action upon missed messages. This functionality enables each party to gracefully synchronize its application when reconnecting during a session. Each session will establish and maintain independent incoming and outgoing sequence number series for the purpose of gap detection, synchronization and recovery.

The incoming and outgoing sequence numbers are automatically initialized to 1 (one) based on one of the following options:

- The start of each day (1/24)
- The start of each week on Sunday (5/24)

The ResetSeqNumFlag (141) may optionally be sent by FXall in the **Logon** to notify the acceptor when to reset the sequence numbers.

SESSION LEVEL

This section outlines the implementation of the FIX session layer and methods to address message sequence gaps.

Establishing a FIX connection involves three distinct operations:

1. Creation of a telecommunications link
2. Authentication/acceptance of the initiator by the acceptor
3. Message synchronization

The sequence of a connection is as follows:

- The session initiator establishes a telecommunication link with the session acceptor.
- The initiator sends a **Logon** message to the acceptor. The acceptor authenticates the identity of the initiator by examining the **Logon** message. If the initiator is successfully authenticated, the acceptor responds with a **Logon** message. If authentication fails, the session acceptor terminates the connection after optionally sending a **Logout** message to indicate the reason of failure. The initiator must obtain the Confirming **Logon** message from the acceptor before the session fully established.
- Once authentication has been completed, prior to establishing a session the initiator and acceptor synchronize their messages through interrogation of the **MsgSeqNum (34)** field. This function must be performed prior to the exchange of any new messages by either party. A comparison of the **MsgSeqNum (34)** in the **Logon** message to the internally monitored next expected sequence number identifies any message gaps.
- If a gap is detected by either the initiator or the acceptor, a **ResendRequest** is issued to request the retransmission of the missing messages by the counterparty. The counterparty sends a **SequenceReset** to fill the message gap for one or more Session Level messages. Business Level messages are sent with the **PossDup (43)** flag set to "Y".
- After completion of the initialization process, normal message exchanges begin. Normal termination of the message exchanges are completed via the exchange of **Logout** messages. Termination by other means should be considered an abnormal condition and addressed as errors. Session termination without the receipt of a **Logout** should be treated by the counterparty as **logged out**. A new session begins when the incoming and outgoing sequence numbers by both parties are reset to 1(one) and 1(one).

STANDARD FIX HEADER

Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
8	BeginString	R	String	The protocol being used to communicate. (Always unencrypted, must be 1st field in message)	FIX.4.2 FIX.4.3 FIX.4.4 FIXT.1.1 (used for FIX 5.0 SP2)
9	BodyLength	R	Length	The length of the message. (Always unencrypted, must be 2nd field in message)	Integer > 0
35	MsgType	R	String	The type of message being sent. (Always unencrypted, must be 3rd field in message)	Refer to section below for supported message types
1128	ApplVerID		String	Specifies the service pack release being applied at message level. Enumerated field with values assigned at time of service pack release. 9 = FIX50SP2 [the only supported version]	9
49	SenderCompID	R	String	The session ID of the sending application. (Always unencrypted)	May be any alphanumeric combination that is agreed upon by initiator and acceptor at time of session setup. NOTE: This is the same as the acceptor's TargetCompID value
56	TargetCompID	R	String	The session ID of the receiving application. (Always unencrypted)	May be any alphanumeric combination that is agreed upon by initiator and acceptor at time of session setup. NOTE: This is the same as the acceptor's SenderCompID value
34	MsgSeqNum	R	SeqNum	The sequence number of the current message	Integer, must be positive starting from one
115	OnBehalfOfCompID		String	Trading partner company ID used when sending messages via a third party (Can be embedded within encrypted data section)	May be any alphanumeric combination that is agreed upon by initiator and acceptor at time of session setup
52	SendingTime	R	UTC Timestamp	The time the message was sent. Always in GMT.	Formatted as yyyyMMdd-HH:mm:ss.SSS

122	OrigSendingTime		UTC Timestamp	Required when message is a PossDup. This represents the original sending time of the message.	Formatted as yyyyMMdd- HH:mm:ss.SSS
43	PossDupFlag		Boolean	Required when message is transmitted in response to a Resend Request. Indicates possible retransmission of message with this sequence number.	N = Original transmission Y = Possible duplicate

STANDARD FIX TRAILER

Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
<u>10</u>	Checksum	R	String	Three byte, simple checksum. ALWAYS LAST FIELD IN MESSAGE; i.e. serves, with the trailing <SOH>, as the end-of-message delimiter. Always defined as three characters. (Always unencrypted)	000-999

CHECKSUM CALCULATION

The checksum of a FIX message is calculated by summing every byte of the message up to, but not including, the checksum field itself. This checksum is then transformed into a modulo 256 number for transmission and comparison. The checksum is calculated after all encryption is completed, i.e. the message as transmitted between parties is processed. For transmission, the checksum must be sent as printable characters, so the checksum is transformed into three ASCII digits.

For example, if the checksum has been calculated to be 274 then the Modulo 256 value is 18 ($256 + 18 = 274$). This value is transmitted as |10=018| where "10=" is the tag for the checksum field.

SESSION LEVEL MESSAGES

Logon (A)

The **Logon** message is sent by the initiator to establish a FIX connection with the acceptor. Once the acceptor authenticates the initiator's logon, it will send a confirmation **Logon** in response.

FXall does not support encryption at the FIX application layer, however SSL encryption may be provided at the transport layer, if requested.

Logon (35 = A)					
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
98	EncryptMethod	R	Integer	The encryption method to be used for the session	0 = None
108	HeartBeatInt	R	Integer	The interval in seconds between each heartbeat message during periods of inactivity	FXall defaults to 30 seconds
141	ResetSeqNumFlag		Boolean	Indicates that both sides of the FIX session should reset sequence numbers	N = do not reset Y = reset to (1,1)

Logout (5)

The **Logout** message is sent by either the initiator or the acceptor to terminate a session.

Logout (35 = 5)					
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
58	Text		String	Free form text for explanation of logout	May be any alphanumeric combination

Heartbeat (0)

Heartbeat messages are sent by both the initiator and the acceptor during periods of inactivity based upon the heartbeat interval specified in the logon. It is used as a means to notify the counterparty that the session is still alive.

Heartbeat (35 = 0)					
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
112	TestRequestID		String	Required when Heartbeat is in response to a TestRequest from a counter party. ID must match the one sent on the TestRequest request	May be any alphanumeric combination

TestRequest (1)

TestRequest messages are sent by either the initiator or the acceptor when a heartbeat interval is exceeded without receipt of a message.

TestRequest (35 = 1)					
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
112	TestRequestID	R	String	The request identifier	May be any alphanumeric combination

ResendRequest (2)

ResendRequest messages are sent by either the initiator or the acceptor when a sequence number gap is detected, and retransmission of missed messages is required.

ResendRequest (35 = 2)					
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
7	BeginSeqNo	R	SeqNum	Message sequence number of the first message in range to be resent.	Integer > 0
16	EndSeqNo	R	SeqNum	Message sequence number of last message in range to be resent. If request is for a single message BeginSeqNo (7) = EndSeqNo. If request is for all messages subsequent to a particular message, EndSeqNo = "0" (representing infinity)	Integer > 0. Or 0 = representing infinity

SequenceReset (4)

SequenceReset messages are sent by either the initiator or the acceptor in response to a **Resend Request** that has administrative or business level messages within the gap range.

SequenceReset (35 = 4)					
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
123	GapFill		Boolean	Indicates that the Sequence Reset message is replacing administrative or application messages which will not be resent	Y = Gap Fill
36	NewSeqNum	R	SeqNum	The sequence number to expect on the next message	Integer > 0

Reject (3)

Reject messages are sent by either the initiator or the acceptor to the counterparty in response to a message that is received which cannot be minimally parsed and handled at the session level.

Reject (35 = 3)					
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
45	RefSeqNum	R	SeqNum	MsgSeqNum of rejected message	Integer > 0
372	RefMsgType		String	The MsgType(35) of the FIX message being rejected	Refer to section on supported message types
373	SessionRejectReason		Char	The code for the session level reject	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 7 = Decryption problem 8 = Signature problem 9 = Comp ID problem 10 = Sending time accuracy problem 11 = Invalid MsgType
58	Text	R	String	Free form text for reason that session level reject occurred	May be any alphanumeric combination

BusinessMessageReject (j)

BusinessMessageReject messages are sent by either the initiator or the acceptor in response to a business message that cannot be minimally parsed at the business level to determine the proper business reject message to send.

BusinessMessageReject (35 = j)					
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
45	RefSeqNum		SeqNum	MsgSeqNum of rejected message	Integer > 0
372	RefMsgType	R	String	The MsgType of the FIX message being referenced	Refer to section on supported message types
379	BusRejectRefID	R	String	The value of the business-level "ID" field on the message being referenced. (i.e. ClOrdID, ExecID, etc)	Required unless the corresponding ID field was not specified.
380	BusRejectReason	R	String	Code to identify reason for a Business Message Reject message	0 = Other 1 = Unknown ID 2 = Unknown Security 3 = Unsupported Message Type 4 = Application not available 5 = Conditionally Required Field Missing 6 = Not authorized 7 = DeliverTo Firm not available
58	Text		String	Required, where available. Free form text for explanation of reject	May be any alphanumeric combination

Chapter 2 Resting, Algos and Fixing Orders

ORDERS - INITIATOR MESSAGES

Standard FIX Header – Additional Fields

Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
50	SenderSubID		String	Assigned value used to identify the message originator (trader, desk, etc). This represents the FXall user mnemonic that is assigned by the acceptor via FXall's Admin Interface	The default format is "<user>@<customer>" unless the value is overridden by an assigned mnemonic in FXall's Admin Tool

NewOrderSingle (D)

The **NewOrderSingle** message is sent by the initiator to the acceptor and represents a new order from the client/user.

Please note the NewOrderSingle message supports multi-allocation orders in the Allocation repeating group (NoLegAllocs) however this is currently limited to Fixing Orders only, i.e. TargetStrategy(847)=1003

NewOrderSingle (35 = D)					
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
11	ClOrdId	R	String	Unique identifier for the Order, as assigned by the buy-side (institution, broker, intermediary etc.) Identified by SenderCompID(49) or OnBehalfOfCompID(115), as appropriate Uniqueness must be guaranteed within a single trading day. Firms, particularly those which electronically submit multi-day orders, trade globally or throughout market close periods, should ensure uniqueness across days; for example, by embedding a date within the ClOrdID field	May be any alphanumeric combination
31984	BatchId		Integer	Unique identifier for MultiBank Batch	Used if the Order is part of a MultiBank Batch

31985	BatchCount		Integer	Number of RFQ's in the MultiBank Batch	Used if the Order is part of a MultiBank Batch
131	QuoteReqID		String	Unique identifier for the RFQ negotiation	May be any alphanumeric combination
117	QuoteID		String	Unique identifier for this quote as sent by the Provider	Used in Forward First when sending the Fixing Order to the winning bank, to reference back to the RFQ message
40	OrdType	R	Char	The order type	1 = Market 2 = Limit 3 = Stop 4 = Stop Limit E = Previously Indicated
59	TimeInForce		Char	Specifies the amount of time that the order remains in effect	0 = Day (default) 1 = GTC 6 = GTD
126	ExpireTime	C	UTCTimestamp	Required when TimeInForce(59) = GTD(6). Time/Date of order expiration (always expressed in UTC, also known as "GMT")	Formatted as yyyyMMdd-HH:mm:ss.SSS
21	HandlingInstruction	R	Char	Instructions for order handling. Required for FIX 4.2 and 4.3 implementations	1 = Automated - No Intervention 2 = Automated - Intervention Ok 3 = Manual
55	Symbol	R	String	Currency pair to be traded	Formatted as RRR/YYY
54	Side	R	Char	Side of order	1 = Client Buy 2 = Client Sell
15	Currency	R	String	Dealt currency of the order	Either RRR or YYY
31027	SpotDate		LocalMktDate	The Spot date	
31027	SplitValueDate		LocalMktDate	Stores the Split Value date if used	When settlement is split across two dates, one is stored in valueDate, and the other in splitValueDate
44	Price	C	Price	Required when OrdType(40) = Limit(2) or Stop Limit(4)	The limit rate at which to execute the order
38	OrdQty	R	Amount	The amount of the dealt currency to be traded	
99	StopPrice	C	Price	Required when OrdType(40) = Stop(3) or StopLimit(4)	The Stop price at which to trigger the order
63	SettlType	C	Char	Required for Forwards. The settlement type	6 = Future
64	SettlDate	C	LocalMktDate	Required for Forwards. The future settlement date	Formatted as YYYYMMDD
4783	SettlLocation	C	String	Precious Metals Settlement Location - Informational field	Set to LON only. Used for Precious Metal trades only, applies to spot or offvenue

30471	ForwardFirst Flag		Char	Indicates the Order is part of Forward First workflow	Y,N
31995	PointDPS		Integer	FXall Forward points precision	
30389	SpotRate		Price	Field added to support Forward First Workflow: This is an indicative reference Spot rate when the previous AJ message was sent	
30390	ForwardPoints		Price	Field added to support Forward First Workflow: Buy-side trader will use this field to send the agreed forward points to the sell-side provider as established during the RFQ process	
1	Account	R	String	The account mnemonic	The customer's allocation account or the break account. Trading on the break account requires Post Trade Allocation support
31345	Npft	R	Boolean	Required for MTF Trade. Non-Price Forming Trade Flag	Y/N
31344	TradeCapacity	R	String	Required for MTF Trade. Customer Trade Capacity	DEAL
30210	SecFin	R	Boolean	Required for MTF Trade. Security Financial Instrument Flag	Y/N
25044	CustomerLEI	R	String	Required for MTF Trade. Customer LEI	
30413	AcctLEI	R	String	Required for MTF Trade. Account LEI	
30211	LegUSI	R	String	Leg USI	Will be populated for all venues
30391	AllocUSI	R	String	Allocation USI	Will be populated for all venues
453	NoPartyIDs	R	Integer	Required for MTF Orders Repeating group below should contain unique combinations of PartyID, PartyIDSource and PartyRole	Integer > 0
->	448	PartyID	C	String	Required if NoPartyIDs(453) > 0 Party identifier code. See PartyIDSource (447) and PartyRole (452). See Appendix for supported values
->	447	PartyIDSource	C	Char	Required if NoPartyIDs(453) > 0 Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Note: applicable values depend upon PartyRole (452) specified.

						See Appendix for supported values
->	452	PartyRole	C	Integer	Identifies the type of PartyID(448)	Identifies the type or role of PartyID (448) specified. See Appendix for supported values
	670	NoLegAllocs		Integer	The number of allocations for this leg	Fixing Orders only Integer > 0
->	672	LegIndividualAllocID		String	Unique identifier for this leg allocation	Fixing Orders only
->	671	LegAllocAccount		String	The customer account mnemonic	Fixing Orders only
->	9252	LegAllocUSIValue		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall value with their own value when they return an Execution Report	Fixing Orders only
->	9255	LegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed	Fixing Orders only
->	30462	LegAllocNpft		Boolean	Non-Forming Price Tag at allocation level	Fixing Orders only
->	30423	LegAllocSecFin		Boolean	Security Financing flag of the allocation. Only used on forwards allocations	Fixing Orders only 'N' = Default 'Y'
->	673	LegAllocQty		Amount	The amount to be allocated for this leg allocation	Fixing Orders only
->	9212	LegAllocSide		Char	The side for this allocation	Fixing Orders only 1 = Client Buy 2 = Client Sell
->	1367	LegAllocCurrency		String	The allocation dealt currency	Fixing Orders only
->	9211	LegAllocContraAmount	C	String	Contra amount calculated using Provider Quoted rate	Fixing Orders only
	847	TargetStrategy		Integer	Target strategy of the order	1001 = TWAP 1002 = VWAP 1003 = Fixing See TargetStrategy section for supported values and description
	957	NoStrategyParameters		Integer	Indicates the number of strategy parameters	
->	958	StrategyParameterName	C	String	The name of the strategy parameter. Start of repeating group	
->	959	StrategyParameterType	C	Integer	The type of the strategy parameter	1 = Int 2 = Length 3 = NumInGroup 4 = SeqNum 5 = TagNum 6 = Float 7 = Qty 8 = Price 9 = PriceOffset 10 = Amt

						11 = Percentage 12 = Char 13 = Boolean 14 = String 15 = MultipleCharValue 16 = Currency 17 = Exchange 18 = MonthYear 19 = UTCTimestamp 20 = UTCTimeOnly 21 = LocalMktDate 22 = UTCDateOnly 23 = Data
->	960	StrategyParameterValue	C	String	The value for the strategy parameter, as defined below	Conditional to TargetStrategy (847)
	60	TransactTime	R	UTCTimestamp	Time this order request was initiated/released by the trader, trading system, or intermediary	Formatted as yyyyMMdd-HH:mm:ss.SSS

NewOrderMultiLeg (AB)

The **NewOrderMultiLeg** message is sent by the initiator to the acceptor and represents a new order from the client/user.

Please note, multi-leg and multi-allocation orders are currently limited to Fixing Orders only, i.e. TargetStrategy(847)=1003

NewOrderMultileg (35 = AB)					
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
11	ClOrdId	R	String	Unique identifier for the Order, as assigned by the buy-side (institution, broker, intermediary etc.) Identified by SenderCompID(49) or OnBehalfOfCompID(115), as appropriate. Uniqueness must be guaranteed within a single trading day. Firms, particularly those which electronically submit multi-day orders, trade globally or throughout market close periods, should ensure uniqueness across days; for example, by embedding a date within the ClOrdID field	May be any alphanumeric combination
31984	BatchId		Integer	Unique identifier for MultiBank Batch	Used if Order is part of a MultiBank Batch
31985	BatchCount		Integer	Number of RFQ's in the MultiBank Batch	Used if Order is part of a MultiBank Batch
131	QuoteReqID		String	Unique identifier for the RFQ negotiation	May be any alphanumeric combination
117	QuoteID		String	Unique identifier for this quote as sent by the Provider	Used in Forward First when sending the Fixing Order to the winning bank, to reference back to the RFQ message
40	OrdType	R	Char	The order type	1 = Market (as NewOrderMultileg only applies to Fixing, only 1 (Market) will be supported initially)
59	TimeInForce		Char	Specifies the amount of time that the order remains in effect	0 = Day (default) 1 = GTC 6 = GTD
126	ExpireTime	C	UTCTimestamp	Required when TimeInForce(59) = GTD(6). Time/Date of order expiration (always expressed in UTC, also known as "GMT")	Formatted as yyyyMMdd-HH:mm:ss.SSS
21	HandlingInstruction	R	Char	Instructions for order handling. Required for FIX 4.2 and 4.3 implementations	1 = Automated - No Intervention 2 = Automated - Intervention Ok 3 = Manual
55	Symbol	R	String	Currency pair to be traded	RRR/YYY format

54	Side	R	Char	Side of order	1 = Client Buy 2 = Client Sell
15	Currency	R	String	Dealt currency of the order	Either RRR or YYY
44	Price	C	Price	Required when OrdType(40) = Limit(2) or Stop Limit(4)	The limit rate at which to execute the order
38	OrdQty	R	Amount	The amount of the dealt currency to be traded	
99	StopPrice	C	Price	Required when OrdType(40) = Stop(3) or StopLimit(4)	The Stop price at which to trigger the order
63	SettlType	C	Char	Required for Swap Trades. The settlement type	6 = Future
64	SettlDate	C	LocalMktDate	Required for Swap Trades. The future settlement date	Formatted as YYYYMMDD
4783	SettlLocation	C	String	Precious Metals Settlement Location - Informational field	Used for Precious Metal trades, applies to spot or offvenue
30471	ForwardFirstFlag		Char	Indicates the Order is part of Forward First workflow	Y,N
31995	PointDPS		Integer	FXall Forward points precision	
30389	SpotRate		Price	Field added to support Forward First Workflow: This is an indicative reference Spot rate when the previous AJ message was sent	
30390	ForwardPoints		Price	Field added to support Forward First Workflow: Buy-side trader will use this field to send the agreed forward points to the sell-side provider as established during the RFQ process	
1	Account	R	String	The account mnemonic	Customer's allocation account or the break account. Trading on the break account requires Post Trade Allocation support
30413	AcctLEI	R	String	Required for MTF Trade. Account LEI	
25044	CustomerLEI	R	String	Required for MTF Trade. Customer LEI	
31345	Npft	R	Boolean	Required for MTF Trade. Non-Price Forming Trade Flag	Y/N
31344	TradeCapacity	R	String	Required for MTF Trade. Customer Trade Capacity	DEAL
30210	SecFin	R	Boolean	Required for MTF Trade. Security Financial Instrument Flag	Y/N
453	NoPartyIDs	R	Integer	Required for MTF Orders Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole	Integer > 0

->	448	PartyID	C	String	Required if NoPartyIDs(453) > 0	Party identifier code. See PartyIDSource (447) and PartyRole (452). See Appendix for supported values
->	447	PartyIDSource	C	Char	Required if NoPartyIDs(453) > 0	Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Note: applicable values depend upon PartyRole (452) specified. See Appendix for supported values
->	452	PartyRole	C	Integer	Required if NoPartyIDs(453) > 0. Identifies the type of PartyID(448)	Identifies the type or role of PartyID (448) specified. See Appendix for supported values
847		TargetStrategy		Integer	Target strategy of the order	1001 = TWAP 1002 = VWAP 1003 = Fixing See TargetStrategy section for supported values and description
957		NoStrategyParameters		Integer	Indicates the number of strategy parameters	
->	958	StrategyParameterName	C	String	The name of the strategy parameter. Start of repeating group	
->	959	StrategyParameterType	C	Integer	The type of the strategy parameter	1 = Int 2 = Length 3 = NumInGroup 4 = SeqNum 5 = TagNum 6 = Float 7 = Qty 8 = Price 9 = PriceOffset 10 = Amt 11 = Percentage 12 = Char 13 = Boolean 14 = String 15 = MultipleCharValue 16 = Currency 17 = Exchange 18 = MonthYear 19 = UTCTimestamp 20 = UTCTimeOnly 21 = LocalMktDate 22 = UTCDateOnly 23 = Data
->	960	StrategyParameterValue	C	String	The value for the strategy parameter, as defined below	Conditional to TargetStrategy (847)

555		NumberOfLegs	R	Integer	Number of legs	Fixing Orders only
->	654	LegRefID	R	String	The unique identifier of this leg	Fixing Orders only
->	600	LegSymbol	R	String	Security Symbol for the individual leg. See Symbol (55) field for description	Fixing Orders only
->	609	LegSecurity Type		String	Indicates type of security	Fixing Orders only FXSPOT = FX Spot (default) FXFWD = FX Forward
->	30250	LegAccount	C	String	Leg account	Fixing Orders only
->	30252	LegAcctLEI	C	String	LEI for the leg account	Fixing Orders only
->	687	LegQty	R	Amount	The amount to be traded on this leg	Fixing Orders only
->	556	LegCurrency	R	String	The Dealt Currency for this leg	Fixing Orders only
->	31027	SpotDate		LocalMktDate	The Spot date for this leg	Fixing Orders only
->	31026	SplitValueDate		LocalMktDate	Stores the Split Value date for this leg if used	Fixing Orders only When settlement is split across two dates, one is stored in valueDate, and the other in splitValueDate
->	624	LegSide	R	Char	The side for this leg	Fixing Orders only 1 = Client Buy 2 = Client Sell
->	611	LegMaturity Date	R	LocalMktDate	Leg Maturity Date	Formatted as YYYYMMDD
->	587	LegSettlType		String	Leg Settlement Type for Value Date Type field	Fixing Orders only
->	588	LegSettlDate		LocalMktDate	Leg Settlement Date	Formatted as YYYYMMDD
->	675	LegSettlCurrency		String	Settlement Currency	Fixing Orders only
->	6215	LegTenor		String	Leg Tenor	Fixing Orders only
->	9253	UPIValue		String	Contains the unique product identifier for the trade being executed	FRVAOP, FRNDOP, FRNDFO
->	30255	LegAcctUSI		String	Transaction Identification Code for the Leg	
->	30263	LegForward Points		Price	Field added to support Forward First Workflow: Buy-side trader will use this field to send the agreed forward points for this individual leg to the sell-side provider as established during the RFQ process	

->	670	NoLegAllocs		Integer	The number of allocations for this leg	Fixing Orders only Integer > 0
->->	672	LegIndividualAllocID		String	Unique identifier for this leg allocation	Fixing Orders only
->->	671	LegAllocAccount		String	The customer account mnemonic	Fixing Orders only
->->	9252	LegAllocUSIValue		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall value with their own value when they return an Execution Report	Fixing Orders only
->->	9255	LegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed	Fixing Orders only
->->	30462	LegAllocNpft		Boolean	Non-Forming Price Tag at allocation level	Fixing Orders only
->->	30423	LegAllocSecFin		Boolean	Security Financing flag of the allocation. Only used on forwards allocations	Fixing Orders only 'N' = Default 'Y'
->->	673	LegAllocQty		Amount	The amount to be allocated for this leg allocation	Fixing Orders only
->->	9212	LegAllocSide		Char	The side for this allocation	Fixing Orders only 1 = Client Buy 2 = Client Sell
->->	1367	LegAllocCurrency		String	The allocation dealt currency	Fixing Orders only
->->	9211	LegAllocContraAmount	C	String	Contra amount calculated using Provider Quoted rate	Fixing Orders only
60		TransactTime	R	UTCTimestamp	Time this order request was initiated/released by the trader, trading system, or intermediary	Formatted as yyyyMMdd-HH:mm:ss.SSS

The following TargetStrategy definitions apply to Orders

TWAP - TargetStrategy (847) = 1001

Since Algo Orders are filled over time they will always be submitted with an **OrdType(40)=E (Previously Indicated)**. Note, Algos only applies to single leg orders.

Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
957	NoStrategyParameters		Integer	Indicates the number of strategy parameters	2
->	958	StrategyParameterName	String	The name of the strategy parameter. Start of repeating group	StartTime
->	959	StrategyParameterType	Integer	The type of the strategy parameter	19 = UTCTimestamp
->	960	StrategyParameterValue	String	The value for the strategy parameter	YYYYMMDD-HH:MM:SS
->	958	StrategyParameterName	String	The name of the strategy parameter. Start of repeating group	EndTime
->	959	StrategyParameterType	Integer	The type of the strategy parameter	19 = UTCTimestamp
->	960	StrategyParameterValue	String	The value for the strategy parameter	YYYYMMDD-HH:MM:SS

VWAP - TargetStrategy (847) = 1002

Since Algo Orders are filled over time they will always be submitted with an **OrdType(40)=E (Previously Indicated)**. Note, Algos only applies to single leg orders.

Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
957	NoStrategyParameters		Integer	Indicates the number of strategy parameters	2
->	958	StrategyParameterName	String	The name of the strategy parameter. Start of repeating group.	StartTime
->	959	StrategyParameterType	Integer	The type of the strategy parameter	19 = UTCTimestamp
->	960	StrategyParameterValue	String	The value for the strategy parameter	YYYYMMDD-HH:MM:SS
->	958	StrategyParameterName	String	The name of the strategy parameter. Start of repeating group.	EndTime
->	959	StrategyParameterType	Integer	The type of the strategy parameter	19 = UTCTimestamp
->	960	StrategyParameterValue	String	The value for the strategy parameter	YYYYMMDD-HH:MM:SS

FIXING - TargetStrategy (847) = 1003

A **Fixing Order** is a specific type of Resting Order that is scheduled to execute at a specific time and place based on the Fixing Series, Fixing Time, and Fixing Time Zone provided in the Strategy Parameters of the Order.

Since Fixing Orders are filled at the specific price determined by the market at that time, they will always be submitted with an **OrdType(40)= 1 (Market)**.

Note, Fixing Orders can apply to single leg or multi-leg orders.

Tag		Tag Name	Required, Conditional	Data Type	Description	Supported Values
	957	NoStrategyParameters		Integer	Indicates the number of strategy parameters	4
->	958	StrategyParameterName	R	String	The name of the strategy parameter. Start of repeating group.	FixingSeries
->	959	StrategyParameterType	R	Integer	The type of the strategy parameter	14 = String
->	960	StrategyParameterValue	R	String	The value for the strategy parameter	The name or mnemonic associated with the fixing series
->	958	StrategyParameterName	R	String	The name of the strategy parameter. Start of repeating group.	FixingDate
->	959	StrategyParameterType	R	Integer	The type of the strategy parameter	14 = String
->	960	StrategyParameterValue	R	String	The value for the strategy parameter	YYYYMMDD
->	958	StrategyParameterName	R	String	The name of the strategy parameter. Start of repeating group.	FixingTime
->	959	StrategyParameterType	R	Integer	The type of the strategy parameter	14 = String
->	960	StrategyParameterValue	R	String	The value for the strategy parameter	HH:MM
->	958	StrategyParameterName	R	String	The name of the strategy parameter. Start of repeating group.	FixingTimeZone
->	959	StrategyParameterType	R	Integer	The type of the strategy parameter	14 = String
->	960	StrategyParameterValue	R	String	The value for the strategy parameter	HK = Asia/Hong_Kong SNG = Asia/Singapore TOK = Asia/Tokyo SYD = Australia/Sydney EURO CET LDN = Europe/London NY = America/New_York GMT

OrderCancelReplaceRequest (G)

The **OrderCancelReplaceRequest** message is sent by the initiator to the acceptor and represents a new order to replace an existing order from the client/user.

OrderCancelReplaceRequest (G)					
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
11	ClOrdId	R	String	Unique identifier for Order as assigned by the buy-side (institution, broker, intermediary etc). Identified by SenderCompID (49) or OnBehalfOfCompID (115) as appropriate. Uniqueness must be guaranteed within a single trading day. Firms, particularly those which electronically submit multi-day orders, trade globally or throughout market close periods, should ensure uniqueness across days; for example, by embedding a date within the ClOrdID field.	May be any alphanumeric combination.
41	OrigClOrdId	R	String	ClOrdID of the previous non-rejected order (NOT the initial order of the day) when canceling or replacing an order.	May be any alphanumeric combination.
40	OrdType	R	Char	The order type	1 = Market 2 = Limit 3 = Stop 4 = Stop Limit E = Previously Indicated
59	TimeInForce		Char	Specifies how long the order remains in effect	0 = Day (default) 1 = GTC 6 = GTD
126	ExpireTime	C	UTCTim estamp	Required when TimeInForce(59) = GTD(6). Time/Date of order expiration (always expressed in UTC (Universal Time Coordinated, also known as "GMT"))	Formatted as yyyyMMdd-HH:mm:ss.SSS
18	ExecInst	C	Char	Only if OrdType(40) <> 1 Execution instructions, applied for pause/resume	S = Suspend / pause 2 = Work / resume
21	HandlingInstruction	R	Char	Required for FIX 4.2 and 4.3 implementations. Instructions for order handling.	1 = Automated - No Intervention 2 = Automated - Intervention Ok 3 = Manual
55	Symbol	R	String	Currency pair to be traded.	Formatted as RRR/YYY
54	Side	R	Char	Side of order.	1 = Client Buy 2 = Client Sell
15	Currency	R	String	Dealt currency of the order.	Either RRR or YYY

44	Price	R, C	Price	Required when OrdType(40) = Limit(2) or Stop Limit(4). The limit rate at which to execute the order.	
38	OrdQty	R	Amount	The amount of the dealt currency to be traded.	
60	TransactTime	R	UTCTimestamp	Time this order request was initiated/released by the trader, trading system, or intermediary.	Formatted as yyyyMMdd-HH:mm:ss.SSS
99	StopPrice	R, C	Price	Required when OrdType(40) = Stop(3) or StopLimit(4). The Stop price at which to trigger the order.	
1	Account	R	String	The account mnemonic.	The customer's account.
31345	Npft	R	Boolean	Required for MTF Trade. Non-Price Forming Trade Flag	Y/N
31344	TradeCapacity	R	String	Required for MTF Trade. Customer Trade Capacity	DEAL
30210	SecFin	R	Boolean	Required for MTF Trade. Security Financial Instrument Flag	Y/N
25044	CustomerLEI	R	String	Required for MTF Trade. Customer LEI	
30413	AcctLEI	R	String	Required for MTF Trade. Account LEI	
30211	LegUSI	R	String	Leg USI	Will be populated for all venues
30391	AllocUSI	R	String	Allocation USI	Will be populated for all venues
453	NoPartyIDs	R	Integer	Required for MTF Orders Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole	Integer > 0
448	PartyID	R, C	String	Required if NoPartyIDs(453) > 0	Party identifier code. See PartyIDSource (447) and PartyRole (452). See Appendix for supported values.
447	PartyIDSource	R, C	Char	Required if NoPartyIDs(453) > 0	Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Applicable values depend upon PartyRole (452) specified. See Appendix for supported values.

452	PartyRole	R, C	Integer	Required if NoPartyIDs(453) > 0. Identifies the type of PartyID(448)	Identifies the type or role of the PartyID (448) specified. See Appendix for supported values.
847	TargetStrategy		Integer	Target strategy of the order	1001 = TWAP 1002 = VWAP 1003 = Fixing See TargetStrategy section for supported values and description
957	NoStrategyParameters		Integer	Indicates the number of strategy parameters	
->	958	StrategyParameterName	R, C	String	The name of the strategy parameter. Start of repeating group.
->	959	StrategyParameterType	R, C	Integer	The type of the strategy parameter 1 = Int 2 = Length 3 = NumInGroup 4 = SeqNum 5 = TagNum 6 = Float 7 = Qty 8 = Price 9 = PriceOffset 10 = Amt 11 = Percentage 12 = Char 13 = Boolean 14 = String 15 = MultipleCharValue 16 = Currency 17 = Exchange 18 = MonthYear 19 = UTCTimestamp 20 = UTCTimeOnly 21 = LocalMktDate 22 = UTCDateOnly 23 = Data
->	960	StrategyParameterValue	R, C	String	The value for the strategy parameter, as defined below Conditional to TargetStrategy (847)

TWAP - TargetStrategy (847) =1001

Since Algo Orders are filled over time they will always be submitted with an **OrdType(40)=E (Previously Indicated)**.

Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
957	NoStrategyParameters		Integer	Indicates the number of strategy parameters	2
->	958	StrategyParameterName	String	The name of the strategy parameter. Start of repeating group.	StartTime
->	959	StrategyParameterType	Integer	The type of the strategy parameter	19 = UTCTimestamp
->	960	StrategyParameterValue	String	The value for the strategy parameter	YYYYMMDD-HH:MM:SS
->	958	StrategyParameterName	String	The name of the strategy parameter. Start of repeating group.	EndTime
->	959	StrategyParameterType	Integer	The type of the strategy parameter	19 = UTCTimestamp
->	960	StrategyParameterValue	String	The value for the strategy parameter	YYYYMMDD-HH:MM:SS

VWAP - TargetStrategy (847) =1002

Since Algo Orders are filled over time they will always be submitted with an **OrdType(40)=E (Previously Indicated)**.

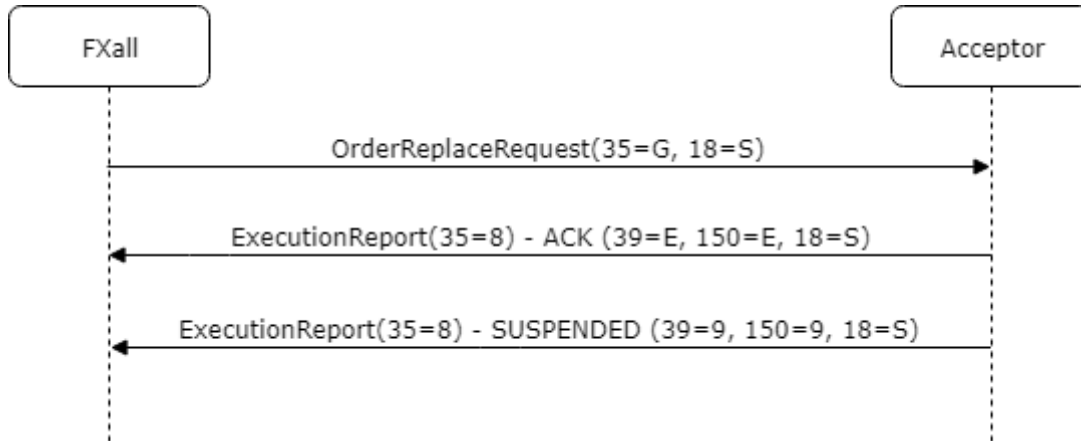
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
957	NoStrategyParameters		Integer	Indicates the number of strategy parameters	2
->	958	StrategyParameterName	String	The name of the strategy parameter. Start of repeating group.	StartTime
->	959	StrategyParameterType	Integer	The type of the strategy parameter	19 = UTCTimestamp
->	960	StrategyParameterValue	String	The value for the strategy parameter	YYYYMMDD-HH:MM:SS
->	958	StrategyParameterName	String	The name of the strategy parameter. Start of repeating group.	EndTime
->	959	StrategyParameterType	Integer	The type of the strategy parameter	19 = UTCTimestamp
->	960	StrategyParameterValue	String	The value for the strategy parameter	YYYYMMDD-HH:MM:SS

FIXING - TargetStrategy (847) =1003

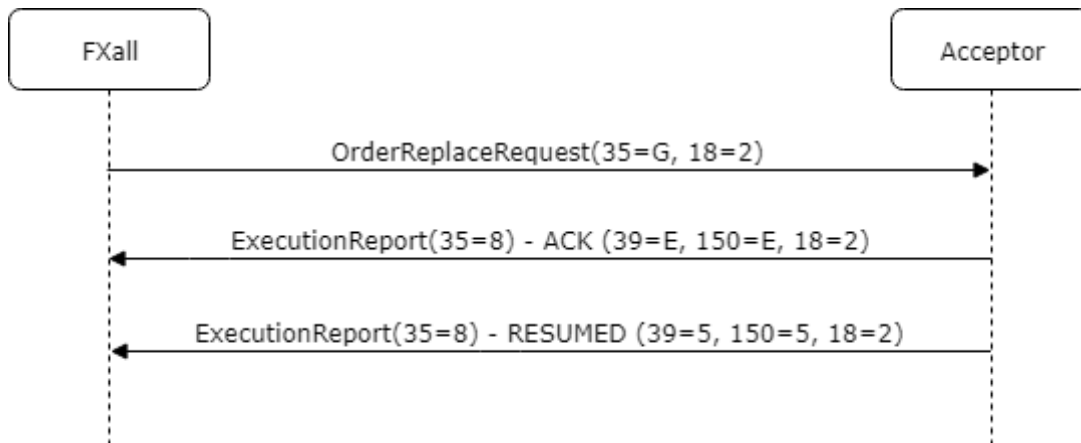
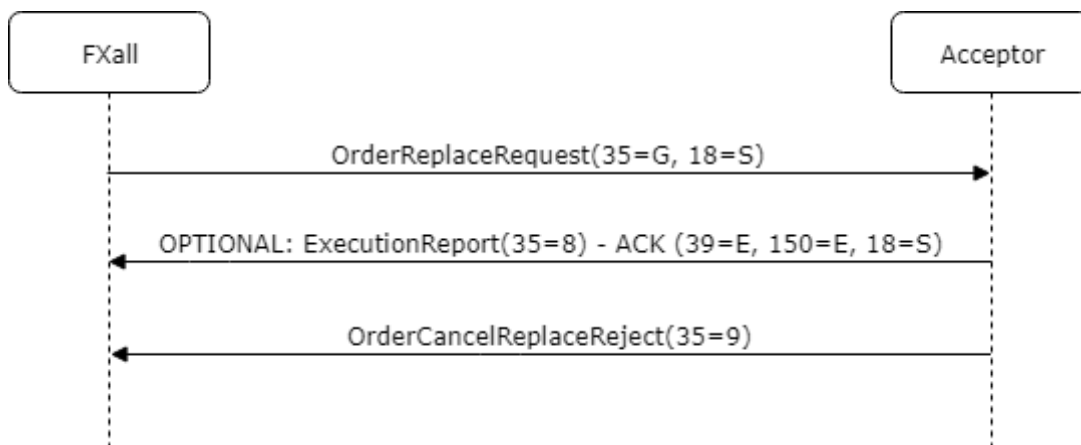
A **Fixing Order** is a specific type of Resting Order that is scheduled to execute at a specific time and place based on the Fixing Series, Fixing Time, and Fixing Time Zone provided in the Strategy Parameters of the Order.

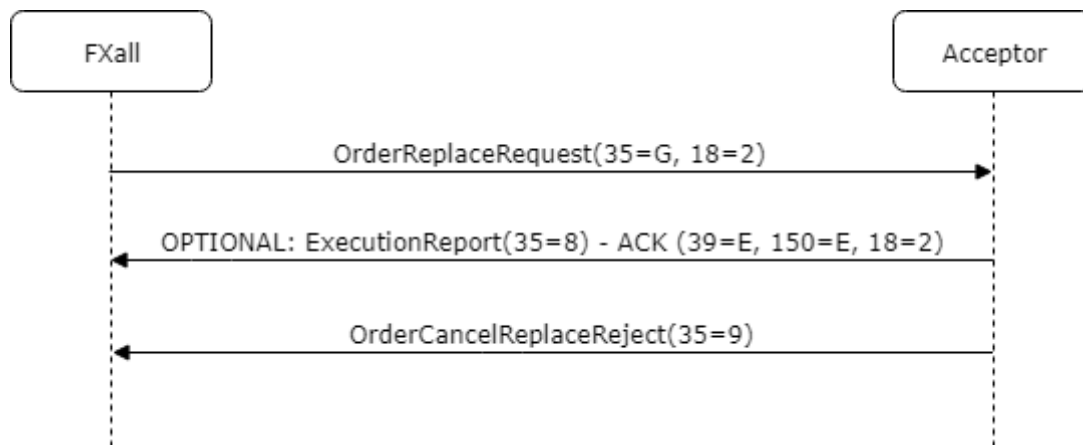
Since Fixing Orders are filled at the specific price determined by the market at that time, they will always be submitted with an **OrdType(40) = 1 (Market)**.

Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
957	NoStrategyParameters		Integer	Indicates the number of strategy parameters	4
->	958	R	String	The name of the strategy parameter. Start of repeating group.	FixingSeries
->	959	R	Integer	The type of the strategy parameter	14 = String
->	960	R	String	The value for the strategy parameter	The name or mnemonic associated with the fixing series
->	958	R	String	The name of the strategy parameter. Start of repeating group.	FixingDate
->	959	R	Integer	The type of the strategy parameter	14 = String
->	960	R	String	The value for the strategy parameter	YYYYMMDD
->	958	R	String	The name of the strategy parameter. Start of repeating group.	FixingTime
->	959	R	Integer	The type of the strategy parameter	14 = String
->	960	R	String	The value for the strategy parameter	HH:MM
->	958	R	String	The name of the strategy parameter. Start of repeating group.	FixingTimeZone
->	959	R	Integer	The type of the strategy parameter	14 = String
->	960	R	String	The value for the strategy parameter	HK = Asia/Hong_Kong SNG = Asia/Singapore TOK = Asia/Tokyo SYD = Australia/Sydney EURO CET LDN = Europe/London NY = America/New_York GMT

Pause & Resume Example (Limit and Algo orders only)**Order pause: success**

- Note: as a convenience we also accept 35=8 REPLACED (150=5) as a pause ack

Order resume: success**Order pause: reject**

Order resume: reject**OrderCancelRequest (F)**

The **OrderCancelRequest** is sent by the initiator to the acceptor and represents the cancellation of an existing order from the client/user.

OrderCancelRequest (F)					
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
11	ClOrdId	/R	String	Unique identifier for Cancel as assigned by the buy-side (institution, broker, intermediary etc.) (Identified by SenderCompID (49) or OnBehalfOfCompID (115) as appropriate). Uniqueness must be guaranteed within a single trading day. Firms, particularly those which electronically submit multi-day orders, trade globally or throughout market close periods, should ensure uniqueness across days, for example by embedding a date within the ClOrdID field	May be any alphanumeric combination
41	OrigClOrdId	R	String	ClOrdID of the order being cancelled	May be any alphanumeric combination
37	OrderId		String	OrderID is required to be unique for each chain of orders	May be any alphanumeric combination
55	Symbol	R	String	Currency pair to be traded	Formatted as RRR/YYY
54	Side	R	Char	Side of order	1 = Client Buy 2 = Client Sell
38	OrdQty	R	Amount	The amount of dealt currency on the order	

60	TransactTime	R	UTCTimestamp	Time this cancel request was initiated/released by the trader, trading system, or intermediary.	Formatted as yyyyMMdd-HH:mm:ss.SSS
----	--------------	---	--------------	---	------------------------------------

OrderStatusRequest (H)

The **OrderStatusRequest** message is sent by the initiator to the acceptor and represents the request for status of an existing order from the client/user.

OrderStatusRequest (H)					
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values
11	ClOrdId	R	String	Unique ID of order for which to request the status of	May be any alphanumeric combination
131	QuoteReqID		String	Unique identifier for the RFQ negotiation. Can be used in Forward First workflow to reference back to the previous Quote	May be any alphanumeric combination
31984	BatchId		Integer	Unique identifier of Multibank Batch	Used if Order is part of Multibank Batch
31985	BatchCount		Integer	Number of Orders in Multibank Batch	Used if Order is part of Multibank Batch
117	QuoteId		String	Unique identifier for the quote	Used for Forward First workflow to reference back to the RFQ message
30472	StatusRequest Reason		Char	Enumerated value defining the reason for the OrderStatusRequest message	1 = RFQ Flow (chase ER reply to a QuoteResponse) 2 = Order flow (chase ER reply to a NewOrder message) =2 to be used here for Order workflow
55	Symbol	R	String	Currency pair to be traded	Formatted as RRR/YYY
54	Side		Char	Side of order	1 = Client Buy 2 = Client Sell
30389	SpotRate		Price	Indicative Spot rate	
30390	ForwardPoints		Price	Agreed forward points, used for Forward First	
555	NumberOfLegs		Integer	Number of legs	For multileg Orders only
->	654	LegRefId	String	The unique identifier for this leg	Integer > 0
->	624	LegSide	String	The side for this leg	
->	30263	LegForwardPoints	Price	Agreed Forward Points for this leg	Used for Forward First

DontKnowTrade (Q)

The ***DontKnowTrade*** message is sent by the initiator to the acceptor and indicates the rejection of an **Execution Report** by the initiator.

DontKnowTrade (Q)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
37	OrderID	R	String	Broker Order ID as identified on problem execution	May be any alphanumeric combination
17	ExecID	R	String	Execution ID of problem Execution	May be any alphanumeric combination
55	Symbol	R	String	Currency pair of the problem execution	Formatted as RRR/YYYY
54	Side	R	Char	Side of order	1 = Client Buy 2 = Client Sell
127	DKReason	R	String	Reason for reject	A = Unknown Symbol B = Wrong Side C = Quantity exceeds order D = No matching order E = Price exceeds limit F = Calculation difference Z = Other
31	LastPrice		Price	Required if specified on the ExecutionRpt	
32	LastQty		Amount	Required if specified on the ExecutionRpt	
58	Text		String	Reason for reject	May be any alphanumeric combination

Execution Acknowledgement (BN)

The optional **Execution Acknowledgement** message is sent by the initiator to the acceptor and indicates the successful processing of an **Execution Report** by the initiator. Must be requested for it to be enabled.

Execution Acknowledgement (BN)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
37	OrderID	R	String	Broker Order ID as identified on problem execution	May be any alphanumeric combination
1036	ExecAckStatus	R	Char	Indicates the status of the execution acknowledgement.	1 = Accepted
17	ExecID	R	String	Execution ID of the execution	May be any alphanumeric combination
55	Symbol	R	String	Currency pair of the problem execution	Formatted as RRR/YYYY
54	Side	R	Char	Side of order	1 = Client Buy 2 = Client Sell
38	OrdQty	R	Amount	The amount of the dealt currency to be traded	
31	LastPrice	R	Price	Required if specified on the ExecutionRpt	
32	LastQty	R	Amount	Required if specified on the ExecutionRpt	
151	LeavesQty	R	Amount	Quantity open for further execution. Conditionally required if specified on the Execution Report	
14	CumQty	R	Amount	The cumulative executed quantity for the chain of orders. Conditionally required if specified on the Execution Report	
6	AvgPrice	R	Price	The average Spot Rate for the chain of orders. Conditionally required if specified on the Execution Report	

ORDERS - ACCEPTOR MESSAGES

OrderCancelReplaceReject (9)

The **OrderCancelReplaceReject** is sent by the acceptor to the initiator and indicates the rejection of an **OrderCancelRequest** or **OrderCancelReplaceRequest** by the acceptor.

OrderCancelReplaceReject (9)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
11	ClOrdId	R	String	Unique ID of cancel request, as assigned by the institution	May be any alphanumeric combination
41	OrigClOrdId	R	String	ClOrdID of the previous non-rejected order (NOT the initial order of the day) when canceling or replacing an order	May be any alphanumeric combination
37	OrderId	R	String	Unique identifier for Order as assigned by sell-side (broker, exchange, ECN). Uniqueness must be guaranteed within a single trading day. Firms which accept multi-day orders should consider embedding a date within the OrderID field to assure uniqueness across days	If CxlRejReason="Unknown order", specify "NONE"
39	OrderStatus	R	String	The order status after this cancel reject is applied	0 = New 1 = Partially filled 2 = Filled 3 = Done for day 4 = Canceled 5 = Replaced 6 = Pending Cancel 7 = Stopped 8 = Rejected 9 = Suspended A = Pending New B = Calculated C = Expired E = Pending Replace
434	CxlRejectResponseTo	R	Char	Identifies the type of request for which a Cancel Reject responds to	1 = Order Cancel Request 2 = Order Cancel Replace Request
102	CxlRejectReason		Char	Code to identify reason for cancel rejection	0 = Too late to cancel 1 = Unknown order 2 = Broker/Exchange Option 3 = Order already in pending cancel or pending replace status 4 = Unable to process Order Mass Cancel request

					5 = OrigOrdModTime (586) did not match last TransactTime(60) of order 6 = Duplicate ClOrdID(11) received 99 = Other
58	Text	R	String	Reason for reject.	May be any alphanumeric combination

ExecutionReport (8)

The **ExecutionReport** is sent by the acceptor to the initiator in response to **NewOrderSingle**, **NewOrderMultiLeg**, **CancelRequest**, **CancelReplaceRequest**, and **OrderStatusRequest** as a notification of a state change to the order.

Please note, any multi-leg and multi-allocation fields are currently limited to Fixing Order workflow only

ExecutionReport (8)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
37	OrderID	R	String	OrderID is required to be unique for each chain of orders	May be any alphanumeric combination
11	ClOrdID	R	String	Required for executions against electronically submitted orders which were assigned an ID by the institution or intermediary	May be any alphanumeric combination
41	OrigClOrdID	R	String	Conditionally required for response to an electronic Cancel or Cancel/Replace request (ExecType=Pending Cancel, Replace or Canceled). ClOrdID of the previous accepted order (NOT the initial order of the day) when canceling or replacing an order	May be any alphanumeric combination
131	QuoteReqID		String	Unique identifier for the RFQ negotiation	May be any alphanumeric combination
117	QuoteID		String	Unique identifier for this traded quote as sent by the Provider	Used in Forward First when sending the Fixing Order to the winning bank, to reference back to the RFQ message
30472	StatusRequestReason		Char	Enumerated value defining the reason for the Execution Report response. Forward First Order flow should use "2"	1 = RFQ Flow (ER reply to a QuoteResponse) 2 = Order flow (ER reply to a NewOrder message)
38	OrdQty		Amount	The amount of the dealt currency traded	
17	ExecID	R	String	Unique identifier of execution message as assigned by sell-side (broker, exchange, ECN). Will be zero for ExecType=I (Order Status)	May be any alphanumeric combination

18	ExecInst		Char	Execution instructions, applied for pause/resume	S = Suspend / pause 2 = Work / resume
19	ExecRefID		String	Required for Trade Cancel and Trade Correct ExecType messages	May be any alphanumeric combination
150	ExecType	R	Char	Describes the purpose of the execution report. 0 = New 1 = Partial Fill (4.2 Only) 2 = Fill (4.2 Only) 3 = Done for day 4 = Canceled 5 = Replaced 6 = Pending Cancel 7 = Stopped 8 = Rejected 9 = Suspended A = Pending New B = Calculated C = Expired E = Pending Replace F = Trade (partial fill or fill - 4.3 and above) G = Trade Correct (4.3 and above) H = Trade Cancel (4.3 and above) I = Order Status (4.3 and above)	1 & 2 was replaced by F in version 4.3 and above. G, H, I were implemented using tag 20 in version 4.2
20	ExecTransType		Char	Identifies transaction type. Only Supported for FIX 4.2. A combination of tags 20 and 39 are used to describe purpose of the Execution Report	0 = New 1 = Cancel 2 = Correct 3 = Status
39	OrderStatus	R	Char	Describes the current state of a CHAIN of orders, same scope as OrderQty, CumQty, LeavesQty, and AvgPR	0 = New 1 = Partially filled 2 = Filled 3 = Done for day 4 = Canceled 5 = Replaced 6 = Pending Cancel 7 = Stopped 8 = Rejected 9 = Suspended A = Pending New B = Calculated C = Expired E = Pending Replace
40	OrdType	R	Char	The order type	1 = Market 2 = Limit 3 = Stop 4 = Stop Limit
55	Symbol	R	String	Currency pair to be traded	Formatted as RRR/YYY
54	Side	R	Char	Side of order	1 = Client Buy 2 = Client Sell
31	LastPrice		Price	Price of this fill.	

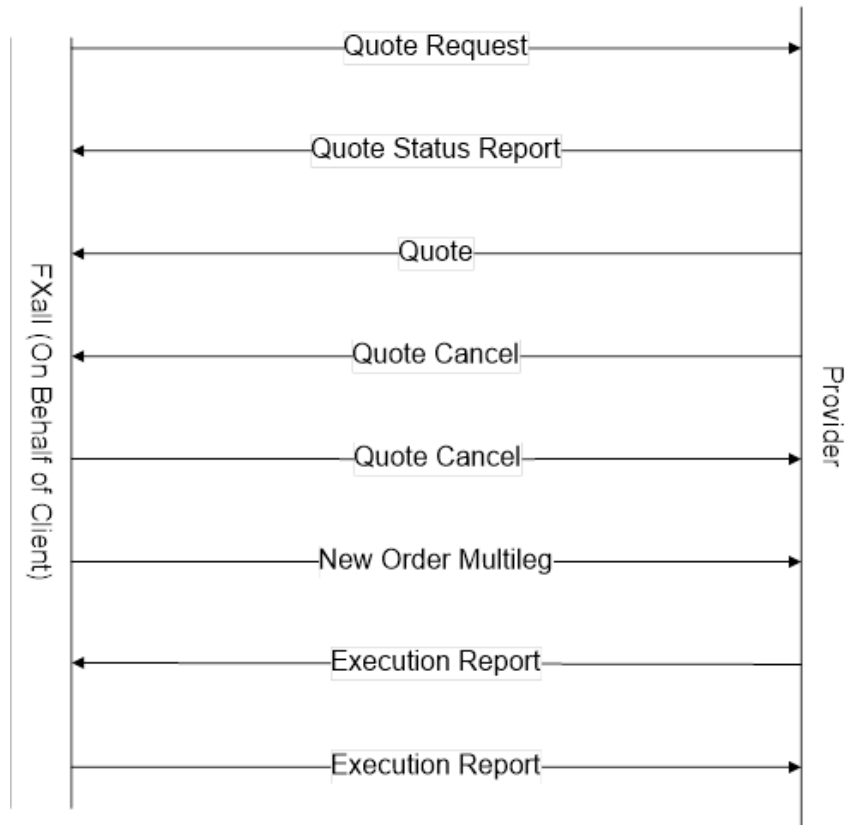
				Required if ExecType = Trade or Trade Correct. Should represent the "all-in" (LastSpotRate + LastForwardPoints + fees) rate for F/R orders	
32	LastQty		Amount	Quantity bought/sold on this (last) fill. Required if ExecType = Trade or Trade Correct	
151	LeavesQty		Amount	Quantity open for further Execution. If the OrdStatus is Canceled, DoneForTheDay, Expired, Calculated or Rejected (in which case the order is no longer active) then LeavesQty could be 0, otherwise LeavesQty = OrderQty - CumQty	
14	CumQty		Amount	The cumulative executed quantity for the order. Required if ExecType = Trade or Trade Correct	
437	ContraTradeQty		Amount	The contra trade quantity of this fill	
119	SettlCurrAmt		Amount	The cumulative contra trade quantity of the order	
120	SettlCurrency		String	The currency that SettleCurrAmt is expressed in.	
4783	SettlLocation	C	String	Precious Metals Settlement Location - Informational field	Used for Precious Metal trades only, applies to spot or offvenue
6	AvgPrice		Price	The average Spot Rate for the order. Required if ExecType = Trade or Trade Correct	
155	SettlCurrFXRate		Price	The Average "All In" rate of the total trade so far, inclusive of commission and points Required if ExecType = Trade or Trade Correct	
194	LastSpotRate		Price	The raw spot rate for this execution. Required if ExecType = Trade or Trade Correct	
195	LastForwardPoints		Price	The forward points for this execution. Required on Final Execution for Forward Orders	
13	CommType		Char	The commission type	1 = per share 2 = percentage 3 = absolute

12	Commission		Price	The commission markup for this execution. If the CommType(13)=2 then a commission of 5% should be represented as 0.05	
64	SettlDate		LocalMktDate	Specific date of trade settlement ("Value Date") in YYYYMMDD format. Expressed in local time at place of settlement. If omitted FXall will assign a value date based on the ValueDate that was sent on the order or the Spot Date if no ValueDate was specified. If present, must match the ValueDate for the Order. Mixed Value Dates are not Supported	Formatted as YYYYMMDD
75	TradeDate		LocalMktDate	Indicates date of trade referenced in this message in YYYYMMDD format. Absence of this field indicates current day. Expressed in local time at place of execution	Formatted as YYYYMMDD
30413	AcctLEI		String	Required for MTF Trade. Account LEI	
30211	LegUSI		String	Leg USI	Will be populated for all venues
30391	AllocUSI		String	Allocation USI	Will be populated for all venues
555	NumberOfLegs	R	Integer	Number of legs in structure	Fixing Orders only
->	654	LegRefID	R	String	The unique identifier of this leg Fixing Orders only Integer > 0
->	600	LegSymbol	R	String	Security Symbol for the individual leg. See Symbol (55) field for description Fixing Orders only
->	609	LegSecurityType		String	Indicates type of security Fixing Orders only FXSPOT = FX Spot FXFWD = FX Forward /required/
->	30250	LegAccount	C	String	Specify the Leg's Account Fixing Orders only
->	30252	LegAcctLEI	C	String	LEI for the account within individual leg Fixing Orders only
->	687	LegQty	R	Amount	The amount to be traded on this leg Fixing Orders only
->	556	LegCurrency	R	String	The Dealt Currency for this leg All legs of a strategy have the same dealt currency
->	31027	SpotDate		LocalMktDate	The Spot date Fixing Orders only
->	31026	SplitValueDate		LocalMktDate	Stores the Split Value date for this leg if used Fixing Orders only When settlement is split across two dates,

						one is stored in valueDate, and the other in splitValueDate
->	624	LegSide	R	Char	The side for this leg	Fixing Orders only 1 = Client Buy 2 = Client Sell
->	621 5	LegTenor		String	Leg Tenor	Fixing Orders only
->	566	LegPrice		Price	The price (ALL-IN-Price) for the individual leg	Fixing Orders only
->	302 62	LegSpotRate		Price	Spot Rate associated with this individual leg	Fixing Orders only Must be quoted to the requested DPS
->	302 63	LegForwardPoints		Price	Forward Points associated with this individual leg. Required for Non-Spot orders	Fixing Orders only Must be quoted to the requested DPS
->	670	NoLegAllocs		Integer	The number of allocations for this leg	Fixing Orders only Integer > 0
->->	672	LegIndividualAllocID		String	Unique identifier for this leg allocation	Fixing Orders only
->->	671	LegAllocAccount		String	The account mnemonic	Fixing Orders only The customer's allocation account
->->	925 2	LegAllocUSValue		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall provided value with their value when they return an Execution Report. SEF trades will always use the FXall supplied value	Fixing Orders only
->->	925 5	LegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed	Fixing Orders only
->->	304 62	LegAllocNpft		Boolean	Non-Forming Price Tag at allocation level	Fixing Orders only Y,N
->->	304 23	LegAllocSecFin		Boolean	Security Financing flag of the allocation. Only used on forwards allocations	Fixing Orders only 'N' = Default 'Y'
->->	673	LegAllocQty		Amount	The amount to be allocated for this leg allocation	Fixing Orders only
->->	921 2	LegAllocSide		Char	The side for this allocation	Fixing Orders only 1 = Client Buy 2 = Client Sell
->->	136 7	LegAllocCurrency		String	The allocation dealt currency	Fixing Orders only Required for MTF
->->	921 1	LegAllocContraAmount	C	String	Contra amount calculated using Provider Quoted rate	Fixing Orders only
60		TransactTime		UTCTimestamp	The time the execution occurred	Fixing Orders only Formatted as yyyyMMdd-HH:mm:ss.SSS

Chapter 3 Options - RFQ

RFQ FLOW DIAGRAM



OPTIONS – INITIATOR MESSAGES

Standard FIX Header – Additional Fields

Tag	Tag Name	Required	Data Type	Description	Supported Values
50	SenderSubID	R	String	Assigned value used to identify the message originator (trader, desk, etc). This represents the FXall user mnemonic that is assigned by the acceptor via FXall's Admin Interface	The default format is "<user>@<customer>", unless the value is overridden by an assigned mnemonic in FXall's Admin Tool.

Quote Request (R)

The **Quote Request** is sent by the initiator to the acceptor to start the RFQ process and request quotes for one or more symbols.

Quote Request (R)						
Tag	Tag Name	Required	Data Type	Description	Supported Values	Notes
131	QuoteReqID	R	String	Unique identifier for the RFQ negotiation	May be any alphanumeric combination	
146	NoRelatedSym	R	Integer	Number of related symbols in Request	Integer > 0	Only 1 supported in initial release
55	Symbol	R	String	Currency pair to be quoted	Formatted as RRR/YYY	
54	Side		Char	The side of the request	1 = client buy 2 = client sell	The absence of a side implies that a two-sided quote is being requested
167	SecurityType	R	String	The type of security being quoted	FOR = Forex FORWARD = Forward OPT = Option	
303	QuoteRequestType		Char	Indicates the type of Quote Request	1 = Request for Premium Quote 2 = Request for Vol Quote	Default is Request for Premium Quote
537	QuoteType		Char	Indicates the type of Quote Request	0 = Indicative 1 = Tradable	Default is Tradable
9241	CalculationAgent		Char	Indicates which party will calculate trade details (such as premium)	1 = Provider 2 = Platform	Default is Provider
9242	QuoteRequestSource		Char		1 = Request For Quote 2 = Callout	Default is Request For Quote
9234	InCompetitionFlag		Char	Indicates if request is being sent to multiple providers	Y,N	
5830	PremiumCurrency	R	String	Currency premium will be paid in		
5020	PremiumDate	R	String		Formatted as YYYYMMDD	
9201	TotalSolveForPremium		Amount	Total premium specified to be paid by taker		Positive value is taker receives. Negative value is taker pays
63	SettlType		Char	Settlement type for order	P = Physical C = Cash	Default is P

120	SettlCurrency	C	String	Cash settlement currency for NDO		Required if SettlementType = C
4783	SettlLocation	C	String	Precious Metals Settlement Location - Informational field and can be ignored	Set to LON only	Used for Precious Metal trades only, applies to spot or offvenue
31997	Onshore	C	Char	Onshore/offshore flag for NDOs	Y,N	Required if SettlementType = C
1	Account	R	String	Account name for all legs on trade		If not supplied, allocations must be supplied for all trades
9240	SEF		Char	If trade is to be executed on FXall SEF	Y,N	
9239	BlockTrade		Char	If trade is above block trade size limit	Y,N	
9254	FinancialEntityDefinition		String	If a SEF trade, contains the Dodd-Frank financial entity definition code for the party requesting the trade.		
9256	SDRName		String	If a SEF trade, contains the name of the swap data repository that trades will be reported to		
25044	CustomerLEI		String	Customer LEI		Required for SEF or MTF Trade
31345	Npft		Boolean	Non-Price Forming Trade Flag	Y,N	Required for MTF Trade
31344	TradeCapacity		String	Customer Trade Capacity	DEAL, MTCH, AOTC	Required for MTF Trade
453	NoPartyIDs		Integer	Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole	Integer > 0	Required for MTF Trade
->	448	PartyID	String	Required if NoPartyIDs(453) > 0		Party identifier code. See PartyIDSource (447) and PartyRole (452). See Appendix for supported values.
->	447	PartyIDSource	Char	Required if NoPartyIDs(453) > 0		Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Note: applicable values

							depend upon PartyRole (452) specified. See Appendix for supported values.
->	452	PartyRole		Integer	Required if NoPartyIDs(453) > 0		Identifies the type or role of the PartyID (448) specified. See Appendix for supported values.
	9207	PriceQuotationType		Integer	Terms that price will be received in	Supported values: 1 = Ccy1/Ccy1 (Ccy1 %) 2 = Ccy2/Ccy2 (Ccy2 %) 3 = Ccy1/Ccy2 (Ccy1 Pips) 4 = Ccy2/Ccy1 (Ccy2 Pips) 5 = Prem Ccy/Ccy1 (Pips) 6 = Prem Ccy/Ccy2 (Pips)	Default is Ccy1 % 5 & 6 not currently supported
	9202	NumberOfStructures	R	Integer	Number of structures in this order	Integer > 0	
->	7940	StructureNumber	R	Integer	User supplied ID. Unique within order only		
->	9203	StructureName	R	Char	Name of structure	Supported Values: 1 Vanilla 2 Straddle 3 Strangle, 4 Risk Reversal 5 Spread 6 Seagull 7 Butterfly 8 Generic	
->	555	NumberOfLegs	R	Integer	Number of legs in structure		
->->	654	LegRefID	R	String	The unique identifier of this leg	Integer > 0	
->->	687	LegQty	R	Amount	The amount to be traded on this leg		
->->	556	LegCurrency	R	String	The Dealt Currency for this leg		All legs of a strategy have the same dealt currency

->->	3102 7	SpotDate		LocalM ktDate	The Spot date for this leg		
->->	624	LegSide		Char	The side for this leg	1 = client buy 2 = client sell	The absence of this field implies that a two-sided quote is being requested
->->	611	LegMaturit yDate	R	LocalM ktDate	Option Expiry Date	Formatted as YYYYMMDD	Either 611 and 588 or 6215 must be specified
->->	587	LegSettlTy pe		String	Indicates order settlement period		
->->	588	LegSettlDa te	R	LocalM ktDate	Option delivery date.	Formatted as YYYYMMDD	
->->	612	LegStrikeP rice	R	Price	Strike Price for an Option		
->->	9204	LegStrikeT ype		Char	Specified type of strike	V = Value D = Delta R = Solve For A = ATMDNS S = ATMS F = ATMF	Default is value
->->	6215	LegTenor		String	Leg Tenor		
->->	201	Put/Call	R	Char	Put Call flag for this leg	0 = Put, 1 = Call	
->->	9205	ExpiryCut	R	String	Expiry cut string. See appendix for allowed values		
->->	9206	OptionTyp e	R	Char	Type of option trade	1 = Vanilla 2 = Knockout 3= Double Knockout 4 = Digital 5 = One Touch 6 = Double One Touch 7 = Knockin 8 = Double Knockin 9 = KIKO 10 = No Touch 11 = Double No Touch	
->->	9253	UPIValue		String	Contains the unique product identifier for the trade being executed	FRVAOP, FRNDOP	Required for SEF trade
->->	9257	ReportingP arty		String	If a SEF trade, contains whether the client or provider has the responsibility to report this leg	T = Taker (Client), M=Maker (Provider)	
->->	3025 5	LegAcctUS I		String	Transaction Identification Code for the leg		

->->	9235	Trigger1		Price			Required for all exotics
->->	9243	Trigger1Type		Char	Type of Trigger	I = IN, O = OUT	Required for all exotics
->->	9244	Trigger1Direction		Char	Direction of Trigger	U = Triggers as spot crosses moving up, D = Triggers as spot crosses moving down	Required for all exotics
->->	9245	Trigger1Start		LocalMktDate	Start date of trigger window		
->->	9246	Trigger1End		LocalMktDate	End date of trigger window		
->->	9236	Trigger2		String			Required for double knockout and double one touch
->->	9247	Trigger2Type		Char	Type of Trigger	I = IN, O = OUT	Required for double knockout and double one touch
->->	9248	Trigger2Direction		Char	Direction of Trigger	U = Triggers as spot crosses moving up, D = Triggers as spot crosses moving down	Required for double knockout and double one touch
->->	9249	Trigger2Start		LocalMktDate	Start date of trigger window		
->->	9250	Trigger2End		LocalMktDate	End date of trigger window		
->->	670	NoLegAllocs		Integer	The number of allocations for this leg	Integer > 0	
->->	672	LegIndividualAllocID		String	Unique identifier for this leg allocation		
->->	671	LegAllocAccount		String	The account mnemonic	The customer's allocation account or the break account.	
->->	9252	LegAllocUSValue		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall provided value with their value when they return an Execution Report		

>-> >	9255	LegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed		
-> >->	3046 2	LegAllocNpft		Boolean	Non-Price Forming Trade Flag at allocation level	Y,N	Required for MTF Trade
-> >->	673	LegAllocQty		Amount	The amount to be allocated for this leg allocation		
-> >->	9212	LegAllocSide		Char	The side for this allocation	1 = client buy 2 = client sell	
	9220	HedgeType	R	Char		S = Spot F = Forward D = One Forward Hedge per Date N = No Hedge	
	9251	HedgeSEF		Char	If hedge trade is to be Executed on FXall SEF	Y,N	
	9237	HedgeDCO		String	If hedge is to be cleared, the derivatives clearing organization it will be sent to		
	3020 0	HedgeVenue		String	Execution venue of the hedge leg	SEF, MTF	Required if the Execution venue of the hedge leg is SEF or MTF
	9221	NumberOfHedgeLegs		Integer	Number of hedge legs in structure		
->	9222	HedgeLegID		Integer		Integer > 0	
->	3025 5	LegAcctUSI		String	Transaction Identification Code for the hedge leg		
->	9270	HedgeLegFields		Integer	Specifies which fields on hedge are set by client vs provider	0 = Amount and Rate specified by client 1 = Rate specified by client 2 = Amount specified by client 3 = Neither amount nor rate specified by client	Fields not specified will be supplied by provider
->	9223	HedgeLegAmount		Amount	The amount to be traded on this leg		
->	9224	HedgeLegCcy		String	The Dealt Currency for this leg		

->	9225	HedgeLeg Side		Char	The side for this leg	1 = client buy 2 = client sell	
->	9226	HedgeLeg Rate		Price	Rate of hedge		
->	9227	HedgeLeg SettleDate		LocalMktDate	Date of Maturity		
->	9260	HedgeLeg Settlement Type		Char	Is leg cash or physical settled. Default is physical	0 = Physical 1 = Cash	Default is P
->	9262	HedgeLeg SettleCcy		String	For NDF leg, the settlement currency		Required if hedge leg settlement type = C
->	9261	HedgeLeg FixingDate		LocalMktDate	For NDF hedge, the Fixing date		Required if hedge leg settlement type = C
->	9228	NoHedgeLegAllocs		Integer	The number of allocations for this leg		
->->	9229	HedgeLeg AllocAccount		String	The account mnemonic		
->->	9258	HedgeUSI Value		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall value with their own value when they return an Execution Report		
->->	9259	HedgeLeg AllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed		
->->	30463	LegAllocSecFinTxInd		Boolean	Security Finance indicator	Y,N	Required if the Execution venue of the hedge leg is MTF
->->	30462	LegAllocNpft		Boolean	Non-Price Forming Trade flag	Y,N	Required if the Execution venue of the hedge leg is MTF
->->	9230	HedgeLegIndividualAllocationID		String	Unique identifier for this leg allocation		
->->	9231	HedgeLeg AllocQty		Amount	The amount to be allocated for this leg allocation		
->->	9264	HedgeLeg AllocSide		Char	The side for this allocation	1 = Buy 2 = Sell	
	60	TransactTime	R	UTCTimestamp		Formatted as yyyyMMdd-HH:mm:ss.SSS	

Quote Cancel (Z)

The **Quote Cancel** is sent by the initiator to the acceptor to cancel a prior quote request for one or more symbols.

Quote Cancel (Z)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
131	QuoteReqID	R	String	Unique identifier for the Quote Request to be cancelled	May be any alphanumeric combination.
298	QuoteCancelType	R	Char	Identifies the type of Quote Cancel Request	1 = Cancel for Symbol(s) 2 = Cancel for Security Type(s) 3 = Cancel for Underlying Symbol 4 = Cancel for All Quotes
146	NoRelatedSym	R	Integer	Number of related symbols in Request	Integer > 0
55	Symbol	R	String	Currency pair for which to cancel the quote	Formatted as RRR/YYYY
60	TransactTime		UTCTimestamp		Formatted as yyyyMMdd-HH:mm:ss.SSS

Quote Acknowledgement (b)

The **Quote Acknowledgement** is sent by the initiator to the acceptor in response to **Quote** messages that are received.

Quote Acknowledgement (b)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
131	QuoteReqID	R	String	Required when acknowledgement is in response to a Quote request	May be any alphanumeric combination
117	QuoteID	R	String	Required when acknowledgment is in response to a Quote message	May be any alphanumeric combination
297	QuoteAckStatus	R	String	Status of the quote acknowledgement	0 = Accepted 1 = Canceled for Symbol(s) 2 = Canceled for Security Type(s) 3 = Canceled for Underlying 4 = Canceled All 5 = Rejected
300	QuoteRejectReason	R	Char	The reason the quote was rejected.	1 = Unknown symbol (Security)

				Required if QuoteAckStatus(297)= 5 (Rejected)	2 = Exchange (Security) closed 3 = Quote Request exceeds limit 4 = Too late to enter 5 = Unknown Quote 6 = Duplicate Quote 7 = Invalid bid/ask spread 8 = Invalid price 9 = Not authorized to quote security
58	Text		String	Free form description of acknowledgement	May be any alphanumeric combination

MarketDataRequest (V)

The **MarketDataRequest** is sent by the initiator to the acceptor in order to request transmission of real-time quote, order, trade, and/or other price information on a subscription basis.

MarketDataRequest (V)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
262	MDReqID	R	String	The unique identifier of the market data request. If SubscriptionRequestType=2 (Disable), then this is the identifier of the previous market data request	May be any alphanumeric combination
263	SubscriptionRequestType	R	String	SubscriptionRequestType indicates to the other party what type of response is expected. A snapshot request only asks for current information. A subscribe request asks for updates as the status changes. Unsubscribe will cancel any future update messages from the counter party	0 = Snapshot 1 = Snapshot + Updates (Subscribe) 2 = Disable previous Snapshot + Update Request (Unsubscribe)
264	MarketDepth	R	Integer	Depth of market for Book Snapshot	Integer > 0
267	NoMDEntryTypes	R	Integer	Number of MDEntryType fields requested	Integer > 0
-> 269	MDEntryType	R	Char	Must be the first field in this repeating group. This is a list of all the types of Market Data Entries that the firm	0 = Bid 1 = Offer 2 = Trade 3 = Index Value

					requesting the Market Data is interested in receiving.	4 = Opening Price 5 = Closing Price 6 = Settlement Price 7 = Trading Session High Price 8 = Trading Session Low Price 9 = Trading Session VWAP Price
146	NoRelatedSym	R	Integer		The number of related symbols	Integer > 0
55	Symbol	R	String		Currency pair to be quoted	Formatted as RRR/YYY

NewOrderMultiLeg (AB)

The **NewOrderMultiLeg** message is sent by the initiator to the acceptor and represents a new multi-leg order from the client/user.

Typical workflow would result in an Execution Report message being sent back from the acceptor. If the initiator does not receive any Execution (8) message after a NewOrderMultiLeg(AB) within an acceptable time window, the NewOrderMultiLeg message is resent with the **PossDup (43)** flag set to “Y” within the Header. The acceptor must answer with an Execution(8) message to this message.

NewOrderMultiLeg (AB)						
Tag	Tag Name	Required	Data Type	Description	Supported Values	Notes
131	QuoteReqID	R	String	Unique identifier for the RFQ negotiation	May be any alphanumeric combination	
11	ClOrdID	R	String	Required for executions against electronically submitted orders assigned an ID by the institution or intermediary		
117	QuoteID		String	Unique identifier for this quote as sent by the Provider	May be any alphanumeric combination	
55	Symbol	R	String			
54	Side	R	Char	Client side for the strategy		1 = client buy 2 = client sell
5830	PremiumCurrency	R	String	Currency premium will be paid in		
5020	PremiumDate	R	String		Formatted as YYYYMMDD	
9301	TotalPremium	R	Amount	Total premium specified to be paid or received by taker		If Side(54) = 1, Total Premium is negative. If Side(54) = 2, Total Premium is positive

63	SettlType	R	String	Settlement type for order	0 = Physical 1 = Cash	
120	SettlCurrency		String	Cash settlement currency for NDO		
4783	SettlLocation	C	String	Precious Metals Settlement Location - Informational field	Set to LON only	Used for Precious Metal trades only, applies to spot or offvenue
31997	Onshore	C	Char	Onshore/offshore flag for NDOs	Y,N	Required if SettlType = 1
1	Account	R	String	Account name for all legs on trade		
9240	SEF		Char	If trade is to be Executed on FXall SEF	Y,N	
9239	BlockTrade		Char	If trade is above block trade size limit	Y,N	
9254	FinancialEntityDefinition		String	If a SEF trade, contains the Dodd-Frank financial entity definition code for party requesting the trade		
9256	SDRName		String	If a SEF trade, contains the name of the swap data repository that trades will be reported to		
25044	CustomerLEI		String	Submitting Party LEI		Required for SEF or MTF Trade
31345	Npft		Boolean	Non-Price Forming Trade Flag	Y,N	Required for MTF Trade
31344	TradeCapacity		String	Customer Trade Capacity	DEAL, MTCH, AOTC	Required for MTF Trade
453	NoPartyIDs		Integer	Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole	Integer > 0	Required for MTF Trade
->	448	PartyID	String	Required if NoPartyIDs(453) > 0		Party identifier code. See PartyIDSource (447) and PartyRole(452) See Appendix for supported values
->	447	PartyIDSource	Char	Required if NoPartyIDs(453) > 0		Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Note: applicable values depend

							upon PartyRole (452) specified. See Appendix for supported values
->	452	PartyRole		Integer	Required if NoPartyIDs(453) > 0		Identifies the type or role of the PartyID (448) specified. See Appendix for supported values
	9207	PriceQuotationType		Integer	Terms that price will be received in	Supported values: 1 = Ccy1/Ccy1 (Ccy1 %) 2 = Ccy2/Ccy2 (Ccy2 %) 3 = Ccy1/Ccy2 (Ccy1 Pips) 4 = Ccy2/Ccy1 (Ccy2 Pips) 5 = Prem Ccy/Ccy1 (Pips) 6 = Prem Ccy/Ccy2 (Pips)	Default is Ccy1 % 5 & 6 not currently supported
	9202	NumberOfStructures	R	Integer	Number of structures in this order		
->	7940	StructureNumber	R	Integer	User supplied ID. Unique within order only		
->	9203	StructureName	R	Char	Name of structure	Supported Values: 1 Vanilla 2 Straddle 3 Strangle 4 Risk Reversal 5 Spread 6 Seagull 7 Butterfly 8 Generic	
->	555	NumberOfLegs	R	Integer	Number of legs in structure		
->->	654	LegRefID	R	String	The unique identifier of this leg	Integer > 0	
->->	687	LegQty	R	Amount	The amount to be traded on this leg		
->->	556	LegCurrency	R	String	The Dealt Currency for this leg		All legs of a strategy have the same dealt currency
->->	31027	SpotDate		LocalMkt Date	The Spot date for this leg		

->->	624	LegSide	R	Char	The side for this leg	1 = client buy 2 = client sell	
->->	611	LegMaturityDate	R	LocalMkt Date	Option Expiry Date	Formatted as YYYYMMDD	
->->	587	LegSettleType		String	Indicates order settlement period		
->->	588	LegSettleDate	R	LocalMkt Date	Option delivery date	Formatted as YYYYMMDD	
->->	612	LegStrikePrice	R	Price	Strike Price for an Option		
->->	201	Put/Call	R	Char	Put Call flag for this leg	0 = Put 1 = Call	
->->	9205	ExpiryCut	R	String	Expiry cut string. See Appendix for allowed values		
->->	9206	OptionType	R	Char	Type of option trade	1 = Vanilla 2 = Knockout 3 = Double Knockout 4 = Digital 5 = One Touch 6 = Double One Touch 7 = Knockin 8 = Double Knockin 9 = KIKO 10 = No Touch 11 = Double No Touch	
->->	9253	UPIValue		String	Contains the unique product identifier for the trade being executed	FRVAOP FRNDOP	Required for SEF trade
->->	9257	ReportingParty		String	If a SEF trade, contains whether the client or provider has the responsibility to report this leg	T = Taker (Client) M = Maker (Provider)	
->->	30255	LegAcctUSI		String	Transaction Identification Code for the leg		
->->	9235	Trigger1		Price			Required for all exotics
->->	9243	Trigger1Type		Char	Type of Trigger	I = IN O = OUT	Required for all exotics
->->	9244	Trigger1Direction		Char	Direction of Trigger	U = Triggers as spot crosses moving up, D = Triggers as spot crosses moving down	Required for all exotics
->->	9245	Trigger1Start		LocalMkt Date	Start date of trigger window		

->->	9246	Trigger1End		LocalMktDate	End date of trigger window		
->->	9236	Trigger2		String			Required for double knockout and double one touch
->->	9247	Trigger2Type		Char	Type of Trigger	I = IN O = OUT	Required for double knockout and double one touch
->->	9248	Trigger2Direction		Char	Direction of Trigger	U = Triggers as spot crosses moving up D = Triggers as spot crosses moving down	Required for double knockout and double one touch
->->	9249	Trigger2Start		LocalMktDate	Start date of trigger window		
->->	9250	Trigger2End		LocalMktDate	End date of trigger window		
->->	9302	LegPremium	R	Amount	Premium for this leg		If LegSide(624) =1, LegPremium is negative If LegSide(624) =2, LegPremium is positive
->->	9303	LegVol		Price			On request for vol quotes market rates will always be included in new order message
->->	566	LegPrice		Price	Price (in terms requested in tag 9207 in request)		
->->	188	SpotRate		Price			On request for vol quotes market rates will always be included in new order message
->->	189	FowardPoints		Price			On request for vol quotes market rates will always be included in new order message
->->	9310	LegAllInRate		Price			On request for vol quotes market rates will always be included in new order message
->->	9304	Depo		Price			On request for vol quotes market rates will always be included in new order message

->->	9305	CounterDepo		Price			On request for vol quotes market rates will always be included in new order message
->->	670	NoLegAllocs	R	Integer	The number of allocs for this leg	Integer > 0	
->->->	672	LegIndividualAllocID		String	Unique identifier for this leg allocation		
->->->	671	LegAllocAccount		String	The account mnemonic	Customer's allocation account or the break account	
->->->	9252	LegAllocUSIValue		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall value with their own value when they return an Execution Report		
->->->	9255	LegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed		Non-Price Forming Trade Flag at allocation level
->->->	30462	LegAllocNpft		Boolean	Non-Price Forming Trade Flag at allocation level	Y,N	Required for MTF trades
->->->	673	LegAllocQty	R	Amount	The amount to be allocated for this leg allocation		
->->->	9212	LegAllocSide	R	Char	The side for this allocation	1 = client buy 2 = client sell	
->->->	9210	AllocPremium	R	Amount	Premium for this allocation		If LegAllocSide(9219)=1, AllocPremium is negative If LegAllocSide(9219)=2, AllocPremium is positive
9220		HedgeType	R	String		S = Spot F = Forward D = One Forward Hedge per Date N = No Hedge	
9251		HedgeSEF		Char	If hedge trade is to be executed on FXall SEF	Y,N	
9237		HedgeDCO		String	If hedge is to be cleared, the derivatives clearing organization it will be sent to		

30200		HedgeVenue		String	Execution venue of the hedge leg	SEF,MTF	Required if the Execution Venue of the hedge leg is SEF or MTF
9221		NumberOfHedgeLegs	R	Integer	Number of hedge legs in structure		
->	9222	HedgeLegID	R	Integer		Integer > 0	
->	30255	LegAcctUSL		String	Transaction Identification Code for the hedge Leg		
->	9223	HedgeLegAmount	R	Amount	The amount to be traded on this leg		
->	9224	HedgeLegCcy	R	String	The Dealt Currency for this leg	S, F, N	
->	9225	HedgeLegSide	R	Char	The side for this leg	1 = client buy 2 = client sell	
->	9226	HedgeLegRate	R	Amount	Rate of hedge		
->	9227	HedgeLegSettleDate	R	LocalMktDate	Date of Maturity.		
->	9260	HedgeLegSettlementType		Char	Is leg cash or physical settled. Default is physical	0 = Physical 1 = Cash	Default is P
->	9262	HedgeLegSettleCcy		String	For NDF leg, the settlement currency		Required if hedge leg settlement type = C
->	9261	HedgeLegFixingDate		LocalMktDate	For NDF hedge, the Fixing date		Required if hedge leg settlement type = C
->	9228	NoHedgeLegAllocs	R	Integer	The number of allocations for this leg		
->->	9229	HedgeLegAllocAccount	R	String	The account mnemonic		
->->	9258	HedgeUSLValue		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall value with their own value when they return an Execution Report		
->->	9259	HedgeLegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed		
->->	30463	LegAllocSecFinTRInd		Boolean	Security Finance indicator	Y,N	Required if the Execution venue of the hedge leg is MTF
->->	30462	LegAllocNpft		Boolean	Non-Price Forming Trade flag	Y,N	Required if the Execution venue of the hedge leg is MTF

->->	9230	HedgeLegIndividualAllocID	R	String	Unique identifier for this leg allocation		
->->	9231	HedgeLegAllocQty	R	Amount	The amount to be allocated for this leg allocation		
->->	9264	HedgeLegAllocSide	R	Char	The side for this allocation	1 = client buy 2 = client sell	
	60	TransactTime	R	UTCTime stamp		Formatted as yyyyMMdd-HH:mm:ss.SS	

ExecutionReport (8)

Once FXall receives the Execution report from the provider, FXall will send a mirror Execution report back to the provider with the TransactTime (tag 60) field containing the MTF execution timestamp. There is no added MIFID related field in this message for providers.

ExecutionReport (8)						
Tag	Tag Name	Required	Data Type	Description	Supported Values	
131	QuoteReqID	R	String	Unique identifier for the RFQ negotiation	May be any alphanumeric combination	
37	OrderID	R	String	OrderID is required to be unique for each chain of orders		
11	ClOrdID	R	String	Required for Executions against electronically submitted orders which were assigned an ID by the institution or intermediary.		
117	QuoteID		String	Unique identifier for this traded quote as sent by the Provider		
453	NoPartyIDs		Integer	Number of parties specified on trade		
->	448	PartyID	String	Name of entity on trade		
->	452	PartyRole	String	Refers to role of party defines in this block	3 = Client ID 7 = Counterparty 64 = MTF 73 = Execution Venue	
->	447	PartyIDSource	Char			
150	ExecType	R	Char	Describes the purpose of the Execution report. 2 = Fill (4.2 Only)		
20	ExecTranstype	R	Char	Identifies transaction type. Only Supported for FIX 4.2 implementations. A combination of tags 20 and 39 are used to describe the purpose of the Execution Report	0 = New 1 = Cancel 2 = Correct 3 = Status	
39	OrderStatus	R	Char		2 = Filled	
40	OrdType	R	Char			
55	Symbol	R	String			
54	Side	R	Char			
5830	PremiumCurrency	R	String	Currency premium will be paid in		
5020	PremiumDate	R	String		Formatted as YYYYMMDD	
9301	TotalPremium	R	Amount	Total premium specified to be paid by taker		
63	SettlType	R	String	Indicates order settlement period	0 = Physical 1 = Cash	
120	SettlCurrency		String	Cash settlement currency for NDO		

4783	SettlLocation	C	String	Precious Metals Settlement Location - Informational field	Set to LON only. Used for Precious Metal trades only, applies to spot or offvenue
31997	Onshore	C	Char	Onshore/offshore flag for NDOs	Y,N
1	Account	R	String	Account name for all legs on trade	
9240	SEF		Char	If trade was Executed on FXall SEF	Y,N
9239	BlockTrade		Char	If trade is above block trade size limit	Y,N
9254	FinancialEntityDefinition		String	If a SEF trade, contains the Dodd-Frank financial entity definition code for the party requesting the trade	
9256	SDRName		String	If a SEF trade, contains the name of the swap data repository that trades will be reported to	
25044	CustomerLEI		String	Submitting Party LEI	
31345	Npft		Boolean	Non-Price Forming Trade Flag	Y,N
31344	TradeCapacity		String	Customer Trade Capacity	DEAL, MTCH, AOTC
9207	PriceQuotationType	R	Integer	Terms that price will be received in	Supported values: 1 = Ccy1/Ccy1 (Ccy1 %) 2 = Ccy2/Ccy2 (Ccy2 %) 3 = Ccy1/Ccy2 (Ccy1 Pips) 4 = Ccy2/Ccy1 (Ccy2 Pips) 5 = Prem Ccy/Ccy1 (Pips) 6 = Prem Ccy/Ccy2 (Pips) 5 & 6 not currently supported
9202	NumberOfStructures	R	Integer	Number of structures in this order	Integer > 0
->	7940	StructureNumber	R	Integer	User supplied ID. Unique within order only
->	9203	StructureName	R	String	Name of structure
->	555	NumberOfLegs	R	Integer	Number of legs in structure
->->	654	LegRefID	R	String	The unique identifier of this leg
->->	687	LegQty	R	Amount	The amount to be traded on this leg
->->	437	LegContraQty	R	Amount	The contra amount for this leg

->->	556	LegCurrency	R	String	The Dealt Currency for this leg	All legs of a strategy have the same dealt currency
->->	624	LegSide	R	Char	The side for this leg	1 - Buy 2 - Sell
->->	611	LegMaturityDate	R	LocalMktDate	Option Expiry Date	Formatted as YYYYMMDD
->->	587	LegSettlType		String	Indicates order settlement period	
->->	588	LegSettlDate	R	LocalMktDate	Option delivery date	Formatted as YYYYMMDD
->->	612	LegStrikePrice	R	Amount	Strike Price for an Option	
->->	201	Put/Call	R	Char	Put Call flag for this leg	0 = Put, 1 = Call
->->	9205	ExpiryCut	R	String	Expiry cut string. See Appendix for allowed values	
->->	9206	OptionType	R	Char	Type of option trade	1 = Vanilla 2 = Knockout 3 = Double Knockout 4 = Digital 5 = One Touch 6 = Double One Touch 7 = Knockin 8 = Double Knockin 9 = KIKO 10 = No Touch 11 = Double No Touch
->->	9235	Trigger1		Price		
->->	9253	UPIValue		String	Contains the unique product identifier for the trade being executed	FRVAOP, FRNDOP
->->	9257	ReportingParty		String	If a SEF trade, contains whether the client or provider has the responsibility to report this leg	T = Taker (Client), M=Maker (Provider)
->->	30255	LegAcctUSI		String	Transaction Identification Code for the leg	
->->	9243	Trigger1Type		Char	Type of Trigger	I = IN, O = OUT
->->	9244	Trigger1Direction		Char	Direction of Trigger	U = Triggers as spot crosses moving up, D = Triggers as spot crosses moving down
->->	9245	Trigger1Start		LocalMktDate	Start date of trigger window	
->->	9246	Trigger1End		LocalMktDate	End date of trigger window	
->->	9236	Trigger2		String		
->->	9247	Trigger2Type		Char	Type of Trigger	I = IN, O = OUT
->->	9248	Trigger 2 Direction		Char	Direction of Trigger	U = Triggers as spot crosses moving up, D = Triggers as spot crosses moving down
->->	9249	Trigger2Start		LocalMktDate	Start date of trigger window	

->->	9250	Trigger2End		LocalMktDate	End date of trigger window	
->->	9302	LegPremium	R	Amount	Premium for this leg	
->->	9303	LegVol	R	Price		
->->	566	LegPrice	R	Price	Price (in terms requested in tag 9207 in request)	
->->	188	SpotRate	R	Price		
->->	189	ForwardPoints	R	Price		
->->	9310	LegAllInRate	R	Price		
->->	9304	Depo		Price		
->->	9305	CounterDepo		Price		
->->	670	NoLegAllocs	R	Integer	The number of allocations for this leg	Integer > 0
->->->	672	LegIndividualAllocID	R	String	Unique identifier for this leg allocation	
->->->	671	LegAllocAccount	R	String	The account mnemonic.	The customer's allocation account or the break account
->->->	9252	LegAllocUSValue		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall provided value with their value when they return an Execution Report	
->->->	9255	LegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed	
->->->	30462	LegAllocNpft		Boolean	Non-price forming Trade Flag at allocation level	Y,N
->->->	673	LegAllocQty	R	Amount	The amount to be allocated for this leg allocation	
->->->	9211	LegAllocCounterQty		Amount	The counter amount to be allocated for this leg allocation	
->->->	9212	LegAllocSide	R	Char	The side for this allocation	1 - Buy 2 - Sell
->->->	9210	AllocPremium	R	Amount	Premium for this allocation	
	9220	HedgeType	R	String		S = Spot F = Forward D = One Forward Hedge per Date N = No Hedge
	9251	HedgeSEF		Char	If hedge trade is to be executed on FXall SEF	Y,N
	9237	HedgeDCO		String	If hedge is to be cleared, the derivatives clearing organization it will be sent to	
	30200	HedgeVenue		String	Execution venue of the hedge leg	SEF,MTF
	9221	NumberOfHedgeLegs	R	Integer	Number of hedge legs in structure	
->	9222	HedgeLegID	R	Integer		Integer > 0
->	30255	LegAcctUSI		String	Transaction Identification Code for the hedge leg	
->	9223	HedgeLegAmount	R	Amount	The amount to be traded on this leg	
->	9224	HedgeLegCcy	R	String	The Dealt Currency for this leg	
->	9225	HedgeLegSide	R	Char	The side for this leg	1 = Buy 2 = Sell

->	9226	HedgeLegRate	R	Amount	Rate of hedge	
->	9227	HedgeLegSettleDate	R	LocalMktDate	Date of Maturity	
->	9233	HedgeLegCounterAmount		Amount	The counter amount to be traded on this leg	
->	9263	HedgeLegUSI		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall provided value with their value when they return an Execution Report	
->	9260	HedgeLegSettlementType		Char	Is leg cash or physical settled. Default is physical	0 = Physical 1 = Cash
->	9262	HedgeLegSettleCcy		String	For NDF leg, the settlement currency	
->	9261	HedgeLegFixingDate		LocalMktDate	For NDF, the Fixing date	
->	9228	NoHedgeLegAllocs	R	Integer	The number of allocations for this leg	
->->	9229	HedgeLegAllocAccount	R	String	The account mnemonic	
->->	9258	HedgeAllocUSIValue		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall provided value with their value when they return an Execution Report	
->->	9259	HedgeLegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being Executed	
->->	30463	LegAllocSecFinTRInd		Boolean	Security Finance indicator	Y,N
->->	30462	LegAllocNpft		Boolean	Non-Price Forming Trade flag	Y,N
->->	9230	HedgeLegIndividualAllocID	R	String	Unique identifier for this leg allocation	
->->	9231	HedgeLegAllocQty	R	Amount	The amount to be allocated for this leg allocation	
->->	9264	HedgeLegAllocSide	R	Char	The side for this allocation	1 = Buy 2 = Sell
->->	9232	HedgeLegAllocContraQty		Amount	The counter amount to be allocated for this leg allocation	
60		TransactTime	R	UTCTimestamp		

DontKnowTrade (Q)

The ***DontKnowTrade*** message is sent by the initiator to the acceptor and indicates the rejection of an **Execution Report (8)** by the initiator.

DontKnowTrade (Q)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
37	OrderID	R	String	Broker Order ID as identified on problem execution	May be any alphanumeric combination
17	ExecID	R	String	Execution ID of problem Execution	May be any alphanumeric combination
127	DKReason	R	String	Reason for reject	A = Unknown Symbol B = Wrong Side C = Quantity exceeds order D = No matching order E = Price exceeds limit F = Calculation difference Z = Other
58	Text		String	Reason for reject	

OPTIONS - ACCEPTOR MESSAGES

Quote (S)

The **Quote** message is sent by the acceptor to the initiator in response to a **QuoteRequest**. It may also be used to publish unsolicited **Quotes**.

Quote (S)						
Tag	Tag Name	Required	Data Type	Description	Supported Values	Notes
131	QuoteReqID	R	String	Required when in response to a Quote Request	May be any alphanumeric combination	
117	QuoteID	R	String	Unique identifier for this quote	May be any alphanumeric combination	
55	Symbol	R	String	The symbol that is being quoted	Formatted as RRR/YYYY	
54	Side		Char	Side of quote for entire order	From customer perspective	The absence of a side implies a two-sided quote
537	QuoteType		Char	Indicates the type of Quote Request	0 = Indicative 1 = Tradable	Default is Tradable. Indicative is not supported yet
9301	TotalPremium	R	Amount	Total premium for this order		Positive = taker receives, negative = taker pays. Mandatory if Quote Request Type is Request for Premium Quote
631	MidPx		Price	Indicates the optional Provider Mid Price		
9202	NumberOfStructures	R	Integer	Number of structures in this order		
->	7940	StructureNumber	R	Integer	ID submitted in quote request	Integer > 0
->	555	NumberOfLegs	R	Integer	Number of legs in structure	Integer > 0
->->	654	LegRefID	R	Integer	ID submitted in quote request	
->->	687	LegQty	R	Amount	Amount quote is for	
->->	31027	SpotDate		LocalMkt Date	The Spot date for this leg	
->->	624	LegSide	R	Char	Side of leg for this quote	The absence of a side implies a two-sided quote
->->	587	LegSettlType		String	Indicates order settlement period	
->->	588	LegSettlDate	R	LocalMkt Date		
->->	9302	LegPremium	R	Amount	Premium for this leg	Mandatory if Quote Request Type is Request for Premium Quote

->->	9303	LegVol	R	Price			
->->	566	LegPrice	C	Price	Price (in terms requested in tag 9207 in request)		Mandatory if Quote Request Type is Request for Premium Quote
->->	811	PriceDelta	R	Price	Delta in percentage of leg		Leg delta notional/leg notional with no conversion if not in the same ccy
->->	188	SpotRate	C	Price			Mandatory if Quote Request Type is Request for Premium Quote
->->	189	ForwardPoints	C	Price			
->->	9310	LegAllInRate	C	Price			
->->	9304	Depo		Price			
->->	9305	CounterDepo		Price			
->->	9306	CalculatedLegStrike		Price			
	9221	NumberOfHedgeLegs	R	Integer	Number of hedge legs in structure		
->	9222	HedgeLegID	R	Integer	ID submitted in quote request	Integer > 0	
->	9223	HedgeLegAmount	R	Amount	The amount to be traded on this leg	Integer > 0	Calculated by maker if not specified by user input
->	9225	HedgeLegSide	R	Char	The side for this leg	1 = Buy 2 = Sell	Calculated by maker if not specified by user input
->	9226	HedgeLegRate	R	Amount	Rate of hedge		Calculated by maker if not specified by user input
->	9227	HedgeLegSettleDate	R	LocalMkt Date	Date of Maturity		Calculated by maker if not specified by user input
->	9261	HedgeLegFixingDate		LocalMkt Date	For NDF, the Fixing date		Calculated by maker if not specified by user input for NDF hedge
60		TransactTime		UTCTime stamp		Formatted as yyyyMMdd-HH:mm:ss.SSS	

Quote Cancel (Z)

The **Quote Cancel** is sent by the acceptor to the initiator to cancel a prior quote. The QuoteCancelType tag will be used to specify a rescind (current quote is withdrawn but future quotes can be sent) vs. a cancel (no further quotes will be sent in response to this quote request.)

Quote Cancel (Z)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
131	QuoteReqID	R	String	Unique identifier for the Quote Request to be cancelled	May be any alphanumeric combination
117	QuoteID		String	Unique identifier for the Quote	May be any alphanumeric combination. Conditionally required if QuoteCancelType = 1
298	QuoteCancelType		Char	Identifies the type of Quote Cancel Request	1 = Rescind this specific quote. 4 = Cancel
146	NoRelatedSym	R	Integer	Number of related symbols in Request	Integer > 0
55	Symbol	R	String	Currency pair for which to cancel the quote	Formatted as RRR/YYY
60	TransactTime		UTCTimestamp		Formatted as yyyyMMdd-HH:mm:ss.SSS

Quote Status Report (AI)

The **Quote Status Report** is sent by the acceptor to the initiator in response to **QuoteRequest**. It is an optional message to specify receipt of a quote request or forwarding the quote to dealer intervention.

Quote Status Report (AI)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
131	QuoteReqID	R	String	Required when acknowledgement is in response to a Quote request	May be any alphanumeric combination
117	QuoteID	R	String	Required when acknowledgment is in response to a Quote message	May be any alphanumeric combination
146	NoRelatedSym	R	Integer	Number of related symbols in Request	Integer > 0
55	Symbol	R	String	Currency pair for which to cancel the quote.	Formatted as RRR/YYY
297	QuoteAckStatus	R	String	Status of the quote acknowledgement	0 = Accepted 1 = Canceled for Symbol(s) 2 = Canceled for Security Type(s) 3 = Canceled for Underlying

					4 = Canceled All 5 = Rejected 10 = Sent to Dealer Intervention
58	Text		String	Free form string for description of acknowledgement	May be any alphanumeric combination

Quote Request Reject (AG)

The **Quote Request Reject** is sent by the acceptor to the initiator to indicate that a quote request will not be responded to.

Quote Request Reject (AG)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
131	QuoteReqID	R	String	Unique identifier for the Quote Request to be cancelled	May be any alphanumeric combination
658	QuoteRequestRejectReason	R	Char	The reason the quote request was rejected	1 = Unknown symbol (Security) 2 = Exchange(Security) closed 3 = Quote Request exceeds limit 4 = Too late to enter 5 = Invalid price 6 = Not authorized to request quote 7 = No match for inquiry 8 = No market for instrument 9 = No inventory 10 = Pass 99 = Other
146	NoRelatedSym	R	Integer	Number of related symbols in Request	Integer > 0
55	Symbol	R	String	Currency pair for which to cancel the quote	Formatted as RRR/YYY
58	Text		String	The reason the quote request was rejected	
60	TransactTime		UTCTimestamp		Formatted as yyyyMMdd-HH:mm:ss.SSS

Market Data Snapshot Full Refresh (W)

MarketDataSnapshot messages sent by the acceptor to the initiator in response to a **MarketDataRequest**

Market Data Snapshot Full Refresh (W)						
Tag	Tag Name	Required	Data Type	Description	Supported Values	Notes
262	MDReqID	R	String			
146	NoRelatedSym	R	Integer			
-> 55	Symbol	R	String			
-> 9350	DeltaType		Char	Type of deltas quoted	S = Spot, F = Fwd	Default = S
-> 9351	ATMType		Char	Definition of ATM point	A = ATM Delta Neutral Straddle F = ATMF	Default = A
-> 9352	DeltaCurrency		String	Delta currency for vol surface		
-> 9353	PremiumCurrency		String	Premium currency for vol surface		
-> 267	NoMDEntryTypes	R	Integer	Number of points requested		
-> 269	MDEntryType	R	Char		0 = Bid 1 = Offer	
-> 270	MDEntryPrice	R	Price	The volatility of the instrument		
-> 611	MaturityDate	R	Date	Market date for volatility		
-> 6215	TenorValue		String	Corresponding tenor value		
-> 167	SecurityType	R	Integer	Type of instrument quoted	0 = ATM 1 = Call 2 = Put 3 = RR 4 = Strangle	
-> 811	PriceDelta		Price	Delta value of quote		Leg delta notional/ leg notional with no conversion if not in the same ccy

Market Data Snapshot Increment Refresh (R)

The **MarketDataSnapshotIncrement** message is sent by the acceptor to the initiator in response to a change in the previously transmitted **MarketDataSnapshot** message.

Market Data Snapshot Increment Refresh (R)						
Tag	Tag Name	Required	Data Type	Description	Supported Values	Notes
262	MDReqID	R	String			
146	NoRelatedSym	R	Integer			
->	55	Symbol	R	String		
->	9350	DeltaType		Char	Type of deltas quoted	S = Spot, F = Fwd Default = S
->	9351	ATMType		Char	Definition of ATM point	A = ATM Delta Neutral Straddle F = ATMF Default = A
->	9352	DeltaCurrency		String	Delta currency for vol surface	
->	9353	PremiumCurrency		String	Premium currency for vol surface	
->	267	NoMDEntryTypes	R	Integer		
->->	269	MDEntryType	R	Char		0 = Bid, 1 = Offer
->->	270	MDEntryPrice	R	Price	The volatility of the instrument	
->->	611	MaturityDate	R	Date	Market date for volatility	
->->	6215	TenorValue		String	Corresponding tenor value	
->->	167	SecurityType	R	Integer	Type of instrument quoted	0 = ATM 1 = Call 2 = Put 3 = RR 4 = Strangle
->->	811	PriceDelta		Price	Delta value of quote	Leg delta notional/leg notional with no conversion if not in the same ccy

Market Data Request Reject (Y)

The **MarketDataRequestReject** is sent by the acceptor to the initiator in response to a **MarketDataRequest** that cannot be honored by the acceptor.

Market Data Request Reject (Y)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
262	MDReqID	R	String	The unique identifier of a prior Market Data Request	May be any alphanumeric combination
281	MDReqRejReason		Char	The MsgType of the FIX message that is referenced	0 = Unknown symbol 1 = Duplicate MDReqID 2 = Insufficient Bandwidth 3 = Insufficient Permissions 4 = Unsupported SubscriptionRequestType 5 = Unsupported MarketDepth 6 = Unsupported MDUpdateType 7 = Unsupported AggregatedBook 8 = Unsupported MDEntryType
58	Text		String	Description of why Market Data Request was rejected	

ExecutionReport (8)

The **ExecutionReport** is sent by the acceptor to the initiator in response to **NewOrderSingle**, **NewOrderMultiLeg**, **CancelRequest**, **CancelReplaceRequest**, and **OrderStatusRequest** as a notification of a state change to the order.

ExecutionReport (8)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
131	QuoteReqID	R	String	Unique identifier for the RFQ negotiation	May be any alphanumeric combination
37	OrderID	R	String	OrderID is required to be unique for each chain of orders	
11	ClOrdID	R	String	Required for Executions against electronically submitted orders which were assigned an ID by the institution or intermediary	
117	QuoteID		String	Unique identifier for this traded quote as sent by the Provider	
453	NoPartyIDs		Integer	Number of parties specified on trade	
->	448	PartyID	String	Name of entity on trade	
->	452	PartyRole	String	Refers to role of party defined in this block	3 = Client ID 7 = Counterparty 64 = MTF 73 = Execution Venue
->	447	PartyIDSource	Char		

				Describes the purpose of the Execution report. 0 = New 1 = Partial Fill (4.2 Only) 2 = Fill (4.2 Only) 3 = Done for day 4 = Canceled 5 = Replaced 6 = Pending Cancel 7 = Stopped 8 = Rejected 9 = Suspended A = Pending New B = Calculated C = Expired E = Pending Replace F = Trade (partial fill or fill - 4.3 and above) G = Trade Correct (4.3 and above) H = Trade Cancel (4.3 and above) I = Order Status (4.3 and above)	
150	ExecType	R	Char		
103	OrdRejReason		Char	4 = Too late to enter 5 = Unknown Quote 11 = Insufficient credit 99 = Other (including Reject for Credit reasons)	
20	ExecTranstype	R	Char	Identifies transaction type. Only Supported for FIX 4.2 implementations. A combination of tags 20 and 39 are used to describe the purpose of the Execution Report	0 = New 1 = Cancel 2 = Correct 3 = Status
39	OrderStatus	R	Char		0 = New 1 = Partially filled 2 = Filled 3 = Done for day 4 = Canceled 5 = Replaced 6 = Pending Cancel 7 = Stopped 8 = Rejected 9 = Suspended A = Pending New B = Calculated C = Expired E = Pending Replace
40	OrdType	R	Char		
55	Symbol	R	String		
54	Side	R	Char		
5830	PremiumCurrency	R	String	Currency premium will be paid in	
5020	PremiumDate	R	String		Formatted as YYYYMMDD
9301	TotalPremium	R	Amount	Total premium specified to be paid by taker	
63	SettlType		String	Settlement type for order	0 = Physical 1 = Cash
120	SettlCurrency		String	Cash settlement currency for NDO	

31997	Onshore	C	Char	Onshore/offshore flag for NDOs	Y,N
1	Account	R	String	Account name for all legs on trade	
9240	SEF		Char	If trade was Executed on FXall SEF	Y,N
9239	BlockTrade		Char	If trade is above block trade size limit	Y,N
9254	FinancialEntityDefinition		String	If a SEF trade, contains the Dodd-Frank financial entity definition code for the party requesting the trade	
9256	SDRName		String	If a SEF trade, contains the name of the swap data repository that trades will be reported to	
25044	CustomerLEI		String	Submitting Party LEI	
31345	Npft		Boolean	Non-Price Forming Trade Flag	Y,N
31344	TradeCapacity		String	Customer Trade Capacity	DEAL,MTCH,AOTC
9207	PriceQuotationType	R	Integer	Terms that price will be received in	Supported values: 1 = Ccy1/Ccy1 (Ccy1 %) 2 = Ccy2/Ccy2 (Ccy2 %) 3 = Ccy1/Ccy2 (Ccy1 Pips) 4 = Ccy2/Ccy1 (Ccy2 Pips) 5 = Prem Ccy/Ccy1 (Pips) 6 = Prem Ccy/Ccy2 (Pips) 5 & 6 not currently supported
9202	NumberOfStructures	R	Integer	Number of structures in this order	Integer > 0
->	7940	StructureNumber	R	Integer	User supplied ID. Unique within order only
->	9203	StructureName	R	String	Name of structure
->	555	NumberOfLegs	R	Integer	Number of legs in structure
->->	654	LegRefID	R	String	The unique identifier of this leg
->->	687	LegQty	R	Amount	The amount to be traded on this leg
->->	437	LegContraQty	R	Amount	The contra amount for this leg
->->	556	LegCurrency	R	String	The Dealt Currency for this leg
->->	31027	SpotDate		LocalMktDate	The Spot date for this leg
->->	624	LegSide	R	Char	The side for this leg
->->	611	LegMaturityDate	R	LocalMktDate	Option Expiry Date

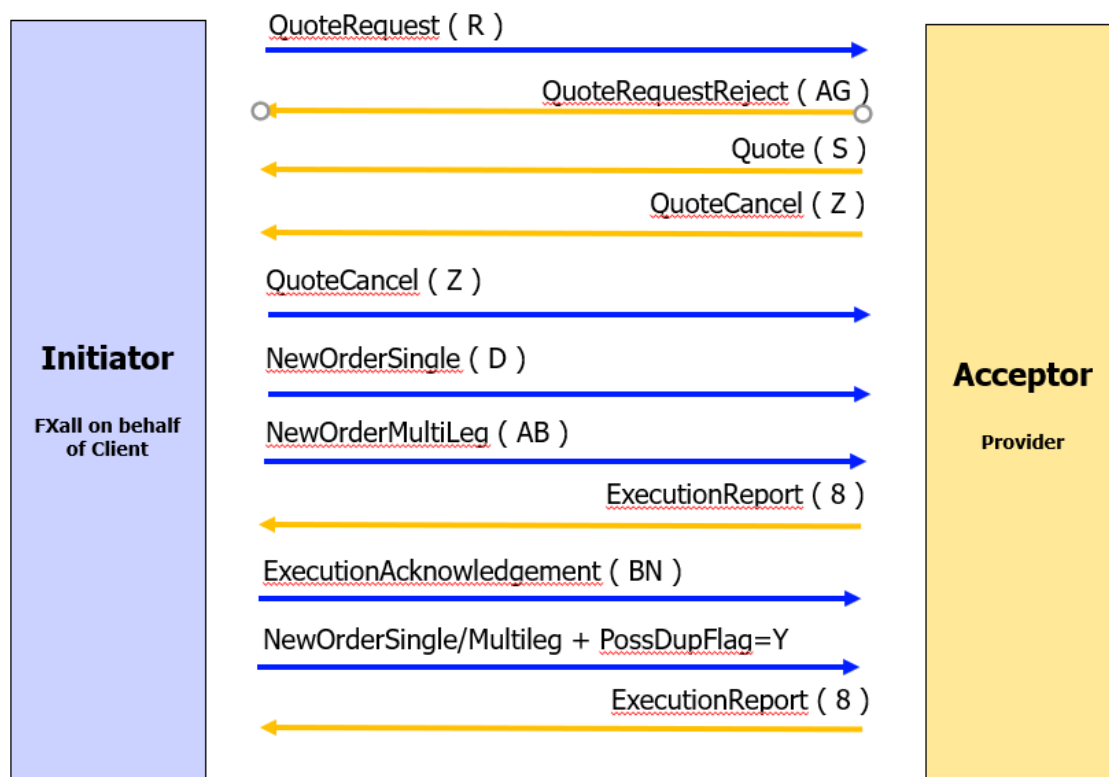
->->	588	LegSettlDate	R	LocalMktDate	Option delivery date	Formatted as YYYYMMDD
->->	612	LegStrikePrice	R	Amount	Strike Price for an Option	
->->	201	Put/Call	R	Char	Put Call flag for this leg	0 = Put, 1 = Call
->->	9205	ExpiryCut	R	String	Expiry cut string. See Appendix for allowed values	
						1 = Vanilla 2 = Knockout 3 = Double Knockout 4 = Digital 5 = One Touch 6 = Double One Touch 7 = Knockin 8 = Double Knockin 9 = KIKO 10 = No Touch 11 = Double No Touch
->->	9206	OptionType	R	Char	Type of option trade	
->->	9235	Trigger1		Price		
->->	9253	UPIValue		String	Contains the unique product identifier for the trade being executed	FRVAOP, FRNDOP
->->	9257	ReportingParty		String	If a SEF trade, contains whether the client or provider has the responsibility to report this leg	T = Taker (Client), M=Maker (Provider)
->->	3025 5	LegAcctUSI		String	Transaction Identification Code for the leg	
->->	9243	Trigger1Type		Char	Type of Trigger	I = IN, O = OUT
->->	9244	Trigger1Direction		Char	Direction of Trigger	U = Triggers as spot crosses moving up D = Triggers as spot crosses moving down
->->	9245	Trigger1Start		LocalMktDate	Start date of trigger window	
->->	9246	Trigger1End		LocalMktDate	End date of trigger window	
->->	9236	Trigger2		String		
->->	9247	Trigger2Type		Char	Type of Trigger	I = IN, O = OUT
->->	9248	Trigger2Direction		Char	Direction of Trigger	U = Triggers as spot crosses moving up, D = Triggers as spot crosses moving down
->->	9249	Trigger2Start		LocalMktDate	Start date of trigger window	
->->	9250	Trigger2End		LocalMktDate	End date of trigger window	
->->	9302	LegPremium	R	Amount	Premium for this leg	
->->	9303	LegVol	R	Price		
->->	566	LegPrice	R	Price	Price (in terms requested in tag 9207 in request)	
->->	188	SpotRate	R	Price		
->->	189	ForwardPoints	R	Price		
->->	9310	LegAllInRate	R	Price		

->->	9304	Depo		Price		
->->	9305	CounterDepo		Price		
->->	670	NoLegAllocs	R	Integer	The number of allocations for this leg	Integer > 0
->->->	672	LegIndividualAllocID	R	String	Unique identifier for this leg allocation	
->->->	671	LegAllocAccount	R	String	The account mnemonic	The customer's allocation account or the break account
->->->	9252	LegAllocUSValue		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall provided value with their value when they return an Execution Report	
->->->	9255	LegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed	
->->->	30462	LegAllocNpft		Boolean	Non-price forming Trade Flag at allocation level	
->->->	673	LegAllocQty	R	Amount	The amount to be allocated for this leg allocation	
->->->	9211	LegAllocCounterQty		Amount	The counter amount to be allocated for this leg allocation	
->->->	9212	LegAllocSide	R	Char	The side for this allocation	1 = Buy 2 = Sell
->->->	9210	AllocPremium	R	Amount	Premium for this allocation	
	9220	HedgeType	R	String		S = Spot F = Forward D = One Forward Hedge per Date N = No Hedge
	9251	HedgeSEF		Char	If hedge trade is to be Executed on FXall SEF	Y,N
	9237	HedgeDCO		String	If hedge is to be cleared, the derivatives clearing organization it will be sent to	
	30200	HedgeVenue		String	Execution venue of the hedge leg	SEF,MTF
	9221	NumberOfHedgeLegs	R	Integer	Number of hedge legs in structure	
->	9222	HedgeLegID	R	Integer		Integer > 0
->	30255	LegAcctUSI		String	Transaction Identification Code for the hedge leg	
->	9223	HedgeLegAmount	R	Amount	The amount to be traded on this leg	
->	9224	HedgeLegCcy	R	String	The Dealt Currency for this leg	
->	9225	HedgeLegSide	R	Char	The side for this leg	1 = Buy 2 = Sell
->	9226	HedgeLegRate	R	Amount	Rate of hedge	
->	9227	HedgeLegSettleDate	R	LocalMktDate	Date of maturity	
->	9233	HedgeLegCounterAmount		Amount	The counter amount to be traded on this leg	
->	9263	HedgeLegUSI		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall provided value with their value when they return an Execution Report	

->	9260	HedgeLegSettlementType		Char	Is leg cash or physical settled. Default is physical	0 = Physical 1 = Cash
->	9262	HedgeLegSettlementCcy		String	For NDF leg, the settlement currency	
->	9261	HedgeLegFixingDate		LocalMktDate	For NDF, the Fixing date	
->	9228	NoHedgeLegAllocs	R	Integer	The number of allocations for this leg	
->->	9229	HedgeLegAllocAccount	R	String	The account mnemonic	
->->	9258	HedgeAllocUSIValue		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall provided value with their value when they return an Execution Report	
->->	9259	HedgeLegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed	
->->	30463	LegAllocSecFinTRInd		Boolean	Security Finance indicator	Y,N
->->	30462	LegAllocNpft		Boolean	Non-Price Forming Trade flag	Y,N
->->	9230	HedgeLegIndividualAllocID	R	String	Unique identifier for this leg allocation	
->->	9231	HedgeLegAllocQty	R	Amount	The amount to be allocated for this leg allocation	
->->	9264	HedgeLegAllocSide	R	Char	The side for this allocation	1 = Buy 2 = Sell
->->	9232	HedgeLegAllocContraQty		Amount	The counter amount to be allocated for this leg allocation	
60		TransactTime	R	UTCTimestamp		
58		Text		String	Description of the rejection reason if ExecType(150) = 8	

Chapter 4 QuickTrade – Cash RFQ

RFQ FLOW DIAGRAM



QUICKTRADE – INITIATOR MESSAGES

Quote Request (R)

The **Quote Request** is sent by the initiator to the acceptor to start the RFQ process and request quotes for one or more symbols. Note, a blank value in the Required/Conditional column denotes an optional value.

Quote Request (R)						
Tag	Tag Name	Required, Conditional	Data Type	Description	Supported Values	Notes
131	QuoteReqID	R	String	Unique identifier for the RFQ negotiation	May be any alphanumeric combination	FXall ID
31984	BatchID		Integer	Unique identifier for the MultiBank Batch		Used if RFQ is part of MultiBank Batch
31985	BatchCount		Integer	Number of RFQs in the MultiBank Batch		Used if RFQ is part of MultiBank Batch
146	NoRelatedSym	R	Integer	Number of related symbols in Request	1	Only 1 supported in initial release
55	Symbol	R	String	Currency pair to be quoted	Formatted as RRR/YYY	

31022	Deliverable		Char	Indicates if the currency pair is deliverable	Y,N	Used for NDF and ND swaps
31997	Onshore		Char	Onshore/offshore flag for NDF transactions and deliverable forwards on ND currencies	Y,N	
31986	MixedDealtFlag		Char	Indicates the RFQ has mixed dealt allocations	Y,N	
30471	ForwardFirstFlag		Char	Indicates the RFQ is for Forward First, used to initiate Forward First workflow	Y,N	
31991	RoundDownCurrency		String	List of currencies that will have amounts truncated		
31994	SpotDPS		Integer	FXall Spot precision		
31995	PointDPS		Integer	FXall Forward points precision		
30379	AllInDPS		Integer	FXall all-in precision		
30381	BaseAmountDPS		Integer	FXall Base Amount Precision		
30382	ContraAmountDPS		Integer	FXall Contra Amount Precision		
54	Side		Char	The side of the request	1 = client buy 2 = client sell	The absence of a side implies that a two-sided quote is being requested
167	SecurityType		String	The type of security being quoted	FXSPOT = FX Spot FXFWD = FX Forward FXSWAP = FX SWAP FXBLOCK = FX Single Spot Portfolio	Indicates type of security. If not present the Security Type is FXSPOT
31992	QuoteRequestType		String	Indicates the type of Quote Request	1 = Send Details 2 = AutoQuoted 3 = SnapinRate 4 = 2-stage forward pricing 5 = Forward First	3 is used in FXall Treasury Center to get quotes from Providers 4 means FXall will send a spot rate to quote from 5 means FXall will send a forward points quote to quote from
537	QuoteType		Char	Indicates the type of Quote	0 = Indicative 1 = Tradable	Default is Tradable
9234	InCompetitionFlag		String	Indicates if request is being sent to multiple providers	Y,N,M	

1913	NumCompetitors		Integer	Returns 1 if the order is not in comp, 2 if there are 2 to 5 banks in competition and 6 if there are 6 or more providers		Enables the provider to identify the number of banks in competition
4783	SettlLocation	C	String	Precious Metals Settlement Location - Informational field	Set to LON only	Used for Precious Metal trades only, applies to spot or offvenue
1	Account	C	String	Account name for all legs on trade		Required if trade not pre-allocated
30413	AcctLEI	C	String	LEI for the account specified in the Tag 1 (Account)		Required if trade not pre-allocated
38	OrderQty	C	Qty	Quantity of the Quote	For FX Swaps the only supported valued is 0	FX Swap only Refer to the LegQty(687) for quantities of the near/far legs
31987	EstimatedOrderQty	C	Char	Indicates whether the order amount is firm or estimated	Y,N	Used for mixed dealt orders when using calculated contra amounts
15	Currency	C	String	Specify the required currency of the quoted price		Required for FX Spot and FX Forward
9239	BlockTrade		Char	If trade is above block trade size limit	Y,N	
9254	FinancialEntityDefinition		String	If a SEF trade, contains the Dodd-Frank financial entity definition code for the party requesting the trade		
9256	SDRName		String	If a SEF trade, contains the name of the swap data repository that trades will be reported to		
25044	CustomerLEI		String	Customer LEI		Required for SEF or MTF Trade
31345	NPFT		Char	Non-Price Forming Trade Flag	Y,N	Required for MTF Trade
9240	SEF		Char	Denotes whether the trade was executed on FXall SEF	Y,N	Required for SEF Trade
31344	TradeCapacity		String	Customer Trade Capacity	MTF: DEAL, MTCH, AOTC SEF: PRINCIPAL AGENCY:	Required for MTF and SEF Trades

	453	NoPartyIDs		Integer	Repeating group below should contain unique combinations of PartyID, PartyIDSource and Party Role	Integer > 0	Required for MTF
->	448	PartyID		String	Party identifier code	Required if NoPartyIDs(453) > 0	See Appendix for supported values
->	447	PartyIDSource		Char	Identifies class or source of the PartyID (448) value	Required if NoPartyIDs(453) > 0	Note: applicable values depend upon PartyRole (452) specified. See Appendix for supported values
->	452	PartyRole		Integer	Identifies the type or role of the PartyID (448) specified	Required if NoPartyIDs(453) > 0	See Appendix for supported values
	555	NumberOfLegs	R	Integer	Number of legs		
->	654	LegRefID	R	String	The unique identifier of this leg	Integer > 0	
->	687	LegQty	R	Amount	The amount to be traded on this leg		For Mixed Dealt, it is the sum of: 1. allocation amount if the allocation currency is the same as the leg currency, 2. allocation contra amount if the allocation currency is not the same as the leg currency
->	31988	EstimatedLegQty	C	Char	Indicates whether the leg amount is firm or estimated	E = estimated Blank otherwise	
->	556	LegCurrency	R	String	The Dealt Currency for this leg		All legs of a strategy have the same dealt currency
->	31027	SpotDate		LocalMktDate	The Spot date for this leg		
->	31026	SplitValueDate		LocalMktDate	Stores the Split Value date for this leg if used		When settlement is split across two dates, one is stored in valueDate, and the other in splitValueDate

->	624	LegSide		Char	The side for this leg	1 = Buy 2 = Sell	The absence of a side implies that a two-sided quote is being requested
->	31989	LegRelativeSide	C	Char	Relative side of the leg compared to leg one	Empty on Leg One On other legs: 1 = Same as leg one 2 = Opposite to leg one	Used for 2-way SingleSpotPortfolios only (FXBLOCK)
->	611	LegMaturityDate	R	LocalMktDate	Leg Maturity Date	Formatted as YYYYMMDD	For NDFs this tag represents the fixing date of the contract therefore there is no need for an additional LegFixingDate (31018) tag
->	587	LegSettlType		String	Leg Settlement Type for Value Date Type field		
->	588	LegSettlDate		LocalMktDate	Leg Settlement Date	Formatted as YYYYMMDD	Either 611 and 588 or 6215 must be specified
->	675	LegSettlCurrency		String	Leg Settlement Currency		
->	6215	LegTenor		String	Leg Tenor		
->	31020	LegFixingSource	C	String	Fixing source of the Non-Deliverable currency pair		Indicates the FXall default NDF fixing setup for the currency pair Only set if 'Deliverable' tag (31022) is set to Y
->	31017	LegFixingTime	C	String	Fixing Time of the Non-Deliverable currency pair		Indicates the FXall default NDF fixing setup for the currency pair Only set if 'Deliverable' tag (31022) is set to Y
->	9253	UPIValue		String	Contains the unique product identifier for the trade being executed	FRNDFO	Used in SEF trades only (NDF)
->	9257	ReportingParty		String	If a SEF trade, contains whether the client or provider has the responsibility to report this leg	T = Taker (Client) M = Maker (Provider)	Required for SEF trade
->	30255	LegAcctUSI		String	Transaction Identification Code for the leg		

->	600	LegSymbol	R	String	Security Symbol for the individual leg. See Symbol (55) field for description		Required for SEF and MTF
->	609	LegSecurityType		String	Indicates type of security	FXSPOT = FX Spot FXFWD = FX Forward	If not present the Security Type is FXSPOT
->	30442	LegNpft		Boolean	Non-Forming Price Tag at allocation level	Y,N	
->	30463	LegSecFin		Boolean	Security Financing flag of the allocation. Only used on forwards allocations	'N' = Default 'Y'	Required for MTF. Client may specify whether the trade was a securities financing trade, e.g. set the flag to 'true' = 'Y'
->	670	NoLegAllocs		Integer	The number of allocations in this leg	Integer > 0	
->->	672	LegIndividualAllocID		String	Unique identifier for this leg allocation		
->->	671	LegAllocAccount		String	The account mnemonic	Customer's allocation account or the break account	
->->	9252	LegAllocUSIValue		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall value with their own value when they return an Execution Report		
>->	9255	LegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed		
->->	30462	LegAllocNpft		Boolean	Non-Price Forming Trade Tag at allocation level	Y,N	
->->	30423	LegAllocSecFin		Boolean	Security Financing flag of the allocation. Only used on forwards allocations	'N' = Default 'Y'	Required for MTF. Client may specify whether the trade was a securities financing trade, e.g. set the flag to 'true' = 'Y'
->->	673	LegAllocQty		Amount	The amount to be allocated for this leg allocation		
->->	9212	LegAllocSide		Char	The side for this allocation	1 = Buy 2 = Sell	

->	30473	LegAllocRelativeSide		Char	Relative side of the allocation compared to the side of the leg of the allocation	1 = Same as allocation leg value 2 = Opposite to allocation leg value	Used for 2-way SingleSpotPortfolios only (FXBLOCK)
->->	1367	LegAllocCurrency		String	The allocation currency		
->->	9211	LegAllocContraAmount	C	Amount	Allocation Contra amount calculated using FXall IQ		Used for Mixed Dealt if LegAllocCurrency (1367) <> Currency (15)
->->	31990	EstimatedLegAllocContraAmount	C	Char	Indicates whether the allocation contra amount is firm or estimated	Y,N	Used for Mixed Dealt if LegAllocCurrency (1367) <> Currency (15)
->->	30426	AllocSecurityConversion		String	Security Conversion Flag of the underlying account	BLANK 'N' 'Y' If the value is Blank / not defined by the client, then tag should not be displayed	Used for SEF trades only
60	TransactTime	R	UTCTimestamp	Formatted as yyyyMMdd-HH:mm:ss.SSS			Used for SEF trades only
58	Text		String	Free format text string			

Quote Response (AJ)

The **Quote Response** message is sent by the initiator to the acceptor to respond to a Quote (S) message and end a negotiation dialogue.

This message currently only applies to Forward First workflow as described in the [Appendix](#). The winning bank will get a QuoteResponse message with a QuoteRequestType of 7 (End Trade) along with QuoteID and agreed forward points. The other banks will receive a QuoteCancel (Z) message.

Quote Response (AJ)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
131	QuoteReqID	R	String	Unique identifier for the Quote Request	May be any alphanumeric combination
31984	BatchId		Integer	Unique Identifier for the MultiBankBatch	Used if the RFQ is part of a MultiBank Batch
31985	BatchCount		Integer	Number of RFQs in the MultiBank Batch	Used if the RFQ is part of a MultiBank Batch
694	QuoteRespType	R	Integer	Identifies the type of Quote Response Field always set to 7 for the winning bank	1 = Hit/Lift 2 = Counter 3 = Expired 4 = Cover 5 = Done Away 6 = Pass 7 = End Trade (indicates an end to trade negotiation) 8 = Timed Out
117	QuoteID	C	String	Unique identifier for the Quote sent by the Provider	May be any alphanumeric combination
54	Side		String	The side of the request 1 = client buy 2 = client sell	This is required when responding to a two-way RFQ. The value will denote the side of the quote accepted
30389	SpotRate		Price	The IQ spot points to the sell-side provider	
30390	ForwardPoints		Price	The agreed Forward Points	
146	NoRelatedSym	R	Integer	Number of related symbols in the request	Integer > 0
55	Symbol	R	String	Currency pair for which to cancel the quote	Formatted as RRR/YYY
555	NumberOfLegs	R	Integer	Number of legs	
-> 654	LegRefID		String	The unique identifier of this leg	Integer > 0
-> 624	LegSide		String	The side for this leg	Required for two-way multi-leg RFQ's. The value will denote the side of the quote accepted
-> 30263	LegForwardPoints		Price	Agreed Forward Points associated with this individual leg	
60	TransactTime		UTCTimestamp		Formatted as yyyyMMdd-HH:mm:ss.SSS
58	Text		String	Free format text string	

Quote Cancel (Z)

The **Quote Cancel** message is sent by the initiator to the acceptor to cancel quotes. In the RFQ FIX implementation the QuoteCancel [type 'Z'] <QuotCRI> is used bi-directionally:

- Clients can also use this message to terminate an entire quote request.

Quote Cancel (Z)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
131	QuoteReqID	R	String	Unique identifier for the Quote Request to be cancelled	May be any alphanumeric combination
31984	BatchId		Integer	Unique Identifier for the MultiBankBatch	Used if the RFQ is part of a MultiBank Batch
31985	BatchCount		Integer	Number of RFQs in the MultiBank Batch	Used if the RFQ is part of a MultiBank Batch
117	QuoteID	C	String	Unique identifier for the Quote	May be any alphanumeric combination. Conditionally required if QuoteCancelType = 5
298	QuoteCancelType		Char	Identifies the type of Quote Cancel Request	4 = Cancel for All Quotes 5 = Cancel quote specified in QuoteID /supported only on a provider quote cancel/ 99 = Other /supported only on reply to the customer request/ To be used in the cases where provider rejects request or provider is not reachable
146	NoRelatedSymbol	R	Integer	Number of related symbols in the request	Integer > 0
55	Symbol	R	String	Currency pair for which to cancel the quote	Formatted as RRR/YYY
60	TransactTime		UTCTimestamp		Formatted as yyyyMMdd-HH:mm:ss.SSS
58	Text		String	Free format text string	

NewOrderSingle (D) – Single Leg Order

The **NewOrderSingle** is sent by the initiator to the acceptor to “Accept” a quote from the provider for a single leg RFQ.

Typical workflow would result in an Execution Report message being sent back from the acceptor. If the initiator does not receive any Execution (8) message after a NewOrderSingle(D) within an acceptable time window, the NewOrderSingle message is resent with the **PossDup (43)** flag set to “Y” within the Header. The acceptor must answer with an Execution(8) message to this message.

NewOrderSingle (D)						
Tag	Tag Name	Required	Data Type	Description	Supported Values	Notes
11	ClOrdID	R	String	Unique identifier for Order as assigned by the client		Identified by SenderCompID (49) or OnBehalfOfCompID (5). Uniqueness must be guaranteed within a single trading day
7108	ReferenceID		String	This is an alphanumeric value that should not exceed 16 characters		If present this will be copied from tag 282 MD Originator on the market incremental data refresh
131	QuoteReqID	R	String	Unique identifier for the RFQ negotiation	May be any alphanumeric combination	
31984	BatchID		Integer	Unique Identifier for the MultiBankBatch		Used for MultiBank Batch RFQ
31985	BatchCount		Integer	Number of RFQs in the MultiBank Batch		Used for MultiBank Batch RFQ
117	QuoteID		String	Unique identifier for this traded quote as sent by the Provider		
146	NoRelatedSym	R	Integer	Number of related symbols in Request	1	Only 1 supported in initial release
55	Symbol	R	String	Currency pair to be quoted	Formatted as RRR/YYY	
31022	Deliverable		Char	Indicates if the currency pair is deliverable	Y,N	Used for NDF and ND swaps
4783	SettlLocation	C	String	Precious Metals Settlement Location - Informational field	Set to LON only	Used for Precious Metal trades only, applies to spot or offvenue
31997	Onshore	C	Char	Onshore/offshore flag for NDF transactions and deliverable forwards on ND currencies	Y,N	

31986	MixedDealtFlag		Char	Indicates the RFQ has mixed dealt allocations	Y,N	
31991	RoundDownCcy		Char		Y,N	
31994	SpotDPS		Integer	FXall Spot precision		
31995	PointDPS		Integer	FXall Forward points precision		
30379	AllInDPS		Integer	FXall all-in precision		
30381	BaseAmountDPS		Integer	FXall Base Amount Precision		
30382	ContraAmountDPS		Integer	FXall Contra Amount Precision		
54	Side	R	Char	The side of the trade	1 = client buy 2 = client sell	
167	SecurityType	R	String	The type of security being quoted	FXSPOT = FX Spot FXFWD = FX Forward FXSWAP = FX SWAP FXBLOCK = FX Single Spot Portfolio	Indicates type of security. If not present the Security Type is FXSPOT
1	Account	C	String	Account name for all legs on trade		Required if trade not pre-allocated
30413	AcctLEI	C	String	LEI for the account specified in the Tag 1		Required if trade not pre-allocated
38	OrderQty	C	Qty	Amount of currency (Tag 15) that is requested for the order		For Mixed Dealt, it is the sum of: 1. the allocation amount if the allocation currency is the same as the leg currency, 2. the allocation contra amount if the allocation currency is not the same as the leg currency
15	Currency	C	String	Specify the required currency of the quoted price		Required for FXSPOT and FXFWD
31027	SpotDate		LocalMktDate	The Spot date		
31026	SplitValueDate		LocalMktDate	Stores the Split Value date if used		When settlement is split across

						two dates, one is stored in valueDate, and the other in splitValueDate
40	OrdType		Char	Type of the order	Supported values: RFQ: D = Quoted [Previously Quoted] Order	
611	LegMaturityDate	R	LocalMktDate	Leg maturity date	Formatted as YYYYMMDD	Either 611 and 588 or 6215 must be specified
587	LegSettlType		String	Leg Settlement Type for Value Date Type field		
588	LegSettlDate		LocalMktDate	Leg Settlement date	Formatted as YYYYMMDD	
675	LegSettlCurrency		String	Leg Settlement currency		
6215	LegTenor		String	Leg Tenor		
31020	LegFixingSource	C	String	Fixing source of the Non-Deliverable currency pair		Indicates the FXall default NDF fixing setup for the currency pair Only set if 'Deliverable' tag (31022) is set to Y
31017	LegFixingTime	C	String	Fixing Time of the Non-Deliverable currency pair		Indicates the FXall default NDF fixing setup for the currency pair Only set if 'Deliverable' tag (31022) is set to Y
9253	UPIValue		String	Contains the unique product identifier for the trade being executed	FRNDFO	Used in SEF trades only (NDF)
9257	ReportingParty		String	If a SEF trade, contains whether the client or provider has the responsibility to report this trade	T = Taker (Client), M=Maker (Provider)	
566	LegPrice		Price	The price (ALL-IN-Price) for the individual leg		
30262	LegSpotRate		Price	Spot Rate associated with this individual leg		

30263	LegForwardPoints		Price	Forward Points associated with this individual leg		Required for NON-SPOT orders
2346	LegMidPR		Price	Leg Mid price/rate		
9310	LegAllInRate		Price	Leg All In Rate		
9239	BlockTrade		Char	If trade is above block trade size limit	Y,N	
9254	FinancialEntityDefinition		String	Contains the Dodd-Frank financial entity definition code for the party requesting the trade		Required for SEF Trade
9256	SDRName		String	Contains the name of the swap data repository that trades will be reported to		Required for SEF Trade
25044	CustomerLEI		String	Customer LEI		Required for SEF or MTF Trade
31345	NPFT		Boolean	Non-Price Forming Trade Flag	Y,N	Required for MTF Trade
9240	SEF		Char	Denotes whether the trade was executed on FXall SEF	Y,N	Required for SEF Trade
31344	TradeCapacity		String	Customer Trade Capacity	MTF: DEAL, MTCH, AOTC SEF: PRINCIPAL AGENCY:	Required for MTF and SEF Trades
453	NoPartyIDs		Integer	Repeating group below should contain unique combinations of PartyID, PartyIDSource and Party Role	Integer > 0	Required for MTF
->	448	PartyID	String	Party identifier code	Required if NoPartyIDs(453) > 0	See Appendix for supported values
->	447	PartyIDSource	Char	Identifies class or source of the PartyID (448) value	Required if NoPartyIDs(453) > 0	Note: applicable values depend upon PartyRole (452) specified See Appendix for supported values.
->	452	PartyRole	Integer	Identifies the type or role of the PartyID (448) specified	Required if NoPartyIDs(453) > 0	See Appendix for supported values
670	NoLegAllocs		Integer	The number of allocations for this leg	Integer > 0	
->	672	LegIndividualAllocID	String	Unique identifier for this leg allocation		
->	671	LegAllocAccount	String	The customer account mnemonic		
->	9252	LegAllocUSIValue	String	Contains the unique swap identifier that will be reported on	.	

					the trade. The provider can override the FXall value with their own value when they return an Execution Report		
->	9255	LegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed		
->	30462	LegAllocNpft		Boolean	Non-Forming Price Tag at allocation level	Y,N	
->	30423	LegAllocSecFin		Boolean	Security Financing flag of the allocation. Only used on forwards allocations	'N' = Default 'Y'	
->	673	LegAllocQty		Amount	The amount to be allocated for this leg allocation		
->	9212	LegAllocSide		Char	The side for this allocation	1 = Client Buy 2 = Client Sell	
->	1367	LegAllocCurrency		String	The allocation dealt currency		
->	9211	LegAllocContraAmount	C	String	Contra amount calculated using Provider Quoted rate		
->	30426	AllocSecurityConversion		String	Security Conversion Flag of the underlying account	BLANK 'N' 'Y' If the value is Blank / not defined by the client, then tag should not be displayed	Required for SEF trades
60		TransactTime	R	UTCTimestamp		Formatted as yyyyMMdd-HH:mm:ss.SS	
58		Text		String	Free format text string		

NewOrderMultileg (AB) – Multi-Leg Order

The **NewOrderMultileg** is sent by the initiator to the acceptor to “Accept” a quote from the provider for a multi leg RFQ.

Typical workflow would result in an Execution Report message being sent back from the acceptor. If the initiator does not receive any Execution (8) message after a NewOrderMultileg(AB) within an acceptable time window, the NewOrderMultileg message is resent with the **PossDup (43)** flag set to “Y” within the Header. The acceptor must answer with an Execution(8) message to this message.

NewOrderMultileg (AB)						
Tag	Tag Name	Required	Data Type	Description	Supported Values	Notes
11	ClOrdID	R	String	Unique identifier for Order as assigned by the client		Identified by SenderCompID (49) or OnBehalfOfCompID (5) as appropriate). Uniqueness must be guaranteed within a single trading day.
131	QuoteReqID	R	String	Unique identifier for the RFQ negotiation	May be any alphanumeric combination	
31984	BatchId		Integer	Unique Identifier for the MultiBankBatch		Used if the RFQ is part of a MultiBank Batch
31985	BatchCount		Integer	Number of RFQs in the MultiBank Batch		Used if the RFQ is part of a MultiBank Batch
117	QuoteID		String	Unique identifier for this traded quote as sent by the Provider		
146	NoRelatedSym	R	Integer	Number of related symbols in Request.	1	Only 1 supported in initial release
55	Symbol	R	String	Currency pair to be quoted	Formatted as RRR/YYY	
31022	Deliverable		Char	Indicates if the currency pair is deliverable	Y,N	Used for NDF and ND swaps
4783	SettlLocation	C	String	Precious Metals Settlement Location - Informational field	Set to LON only	Used for Precious Metal trades only, applies to spot or offvenue
31997	Onshore		Char	Onshore/offshore flag for NDF transactions and deliverable forwards on ND currencies	Y,N	
31986	MixedDealtFlag		Char	Indicates the RFQ has mixed dealt allocations	Y,N	
31991	RoundDownCcy		Char		Y,N	

31994	SpotDPS		Integer	FXall Spot precision		
31995	PointDPS		Integer	FXall Forward points precision		
30379	AllInDPS		Integer	FXall all-in precision		
30381	BaseAmount DPS		Integer	FXall Base Amount Precision		
30382	ContraAmountDPS		Integer	FXall Contra Amount Precision		
54	Side		Char	The side of the Trade	1 = client buy 2 = client sell	
167	SecurityType	R	String	The type of security being quoted	FXSPOT = FX Spot FXFWD = FX Forward /required/ FXSWAP = FX SWAP /required/ FXBLOCK = FX Single Spot Portfolio	Indicates type of security. If not present the Security Type is FXSPOT
1	Account		String	Account name for all legs on trade		If not supplied, allocations must be supplied for all trades
30413	AcctLEI	C	String	LEI for the account specified in Tag 1		
38	OrderQty	C	Qty	Amount of currency (Tag 15) that is requested for the order		
15	Currency	C	String	Specify the required currency of the quoted price		Required for FXSPOT and FXFWD
40	OrdType		Char	Type of the order	Supported values: RFQ: D = Quoted [Previously Quoted] Order	
9239	BlockTrade		Char	If trade is above block trade size limit	Y,N	
9254	FinancialEntityDefinition		String	If a SEF trade, contains the Dodd-Frank financial entity definition code for the party requesting the trade		
9256	SDRName		String	If a SEF trade, contains the name of the swap data repository that trades will be reported to		

30424	SecurityConversion		Char	Security Conversion Flag. Related to specific Foreign Exchange transaction defined by CFTC	Supported values: BLANK 'N' 'Y' If the value is Blank /not defined by the client, then tag should not be displayed	Required for SEF Trade
25044	CustomerLEI		String	Customer LEI		Required for SEF or MTF Trade
31345	NPFT		Boolean	Non-Price Forming Trade Flag	Y,N	Required for MTF Trade
9240	SEF		Char	Denotes whether the trade was executed on FXall SEF	Y,N	Required for SEF Trade
31344	TradeCapacity		String	Customer Trade Capacity	MTF: DEAL, MTCH, AOTC SEF: PRINCIPAL AGENCY:	Required for MTF and SEF Trades
453	NoPartyIDs		Integer	Repeating group below should contain unique combinations of PartyID, PartyIDSource and Party Role	Integer > 0	Required for MTF
->	448	PartyID	String	Party identifier code	Required if NoPartyIDs(453) > 0.	See Appendix for supported values
->	447	PartyIDSource	Char	Identifies class or source of the PartyID (448) value	Required if NoPartyIDs(453) > 0	Note: applicable values depend upon PartyRole (452) specified. See Appendix for supported values
->	452	PartyRole	Integer	Identifies the type or role of the PartyID (448) specified	Required if NoPartyIDs(453) > 0	See Appendix for supported values
555	NumberOfLegs	R	Integer	Number of legs		
->	654	LegRefID	R	String	The unique identifier of this leg	Integer > 0
->	600	LegSymbol	R	String	Security Symbol for the individual leg. See Symbol (55) field for description	

->	609	LegSecurityType		String	Indicates type of security	FXSPOT = FX Spot FXFWD = FX Forward /required/	If not present the Security Type is FXSPOT
->	30250	LegAccount	C	String	Leg account		
->	30252	LegAcctLEI	C	String	LEI for the leg account		
->	687	LegQty	R	Amount	The amount to be traded on this leg		
->	556	LegCurrency	R	String	The Dealt Currency for this leg		
->	31027	SpotDate		LocalMktDate	The Spot date for this leg		
->	31026	SplitValueDate		LocalMktDate	Stores the Split Value date for this leg if used		When settlement is split across two dates, one is stored in valueDate, and the other in splitValueDate
->	624	LegSide	R	Char	The side for this leg	1 = Client Buy 2 = Client Sell	
->	611	LegMaturityDate	R	LocalMktDate	Leg Maturity Date	Formatted as YYYYMMDD	For NDFs this tag represents the fixing date of the contract therefore there is no need for an additional LegFixingDate (31018) tag
->	587	LegSettlType		String	Leg Settlement Type for Value Date Type field		
->	588	LegSettlDate		LocalMktDate	Leg Settlement Date	Formatted as YYYYMMDD	
->	675	LegSettlCurrency		String	Settlement Currency		
->	6215	LegTenor		String	Leg Tenor		
->	31020	LegFixingSource	C	String	Fixing source of the Non-Deliverable currency pair		Indicates the FXall default NDF fixing setup for the currency pair Only set if 'Deliverable' tag (31022) is set to Y
->	31017	LegFixingTime	C	String	Fixing Time of the Non-Deliverable currency pair		Indicates the FXall default NDF fixing setup for the currency pair Only set if 'Deliverable' tag (31022) is set to Y

->	566	LegPrice		Price	The price (ALL-IN-Price) for the individual leg		
->	30262	LegSpotRate		Price	Spot Rate associated with this individual leg		
->	30263	LegForwardPoints		Price	Forward Points associated with this individual leg. Required for NON-SPOT orders		
->	2346	LegMidPR		Price	Leg Mid price/rate		
->	9310	LegAllInRate		Price	Leg All In Rate		
->	9253	UPIValue		String	Contains the unique product identifier for the trade being executed	FRNDFO	Used in SEF trades only (NDF)
->	9257	ReportingParty		String	If a SEF trade, contains whether the client or provider has the responsibility to report this leg	T = Taker (Client), M=Maker (Provider)	
->	30255	LegAcctUSI		String	Transaction Identification Code for the leg		
->	30442	LegNpft		Boolean	Non-Forming Price Tag at allocation level	Y,N	Required for MTF trade
->	30463	LegSecFin		Boolean	Security Financing flag of the allocation. Only used on forwards allocations	'N' = Default 'Y'	Required for MTF trade
->	670	NoLegAllocs		Integer	The number of allocations for this leg	Integer > 0	
->->	672	LegIndividualAllocID		String	Unique identifier for this leg allocation		
->->	671	LegAllocAccount		String	The account mnemonic	The customer's allocation account or the break account	
->->	9252	LegAllocUSIValue		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall value with their own value when they return an Execution Report		
->->	9255	LegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed		Required for SEF and MTF
->->	30462	LegAllocNpft		Boolean	Non-Forming Price Tag at allocation level	Y,N	Required for MTF

->->	30423	LegAllocSecFin		Boolean	Security Financing flag of the allocation. Only used on forwards allocations	'N' - Default 'Y'	Client may specify whether the trade was a securities financing trade, e.g. set the flag to 'true' = 'Y'
->->	673	LegAllocQty		Amount	The amount to be allocated for this leg allocation		
->->	9212	LegAllocSide		Char	The side for this allocation	1 = Client Buy 2 = Client Sell	
->->	1367	LegAllocCurrency		String	The allocation dealt currency		
->->	9211	LegAllocContraAmount	C	String	Contra amount calculated using Provider Quoted rate		
->->	30426	AllocSecurityConversion		String	Security Conversion Flag of the underlying account. Related to specific Foreign Exchange transaction defined by CFTC Supported	BLANK 'N' 'Y' If the value is Blank /not defined by the client, then tag should not be displayed/	Required for SEF Trade
60		TransactTime	R	UTCTimestamp		Formatted as yyyyMMdd-HH:mm:ss.SS	
58		Text		String	Free format text string		

Execution Acknowledgement (BN)

The optional **Execution Acknowledgement** message is sent by the initiator to the acceptor and indicates the successful processing of an **Execution Report (8)** by the initiator.

Providers must request to be enabled for this optional message.

In particular this message sends the MTF execution timestamp back to the provider, so that providers can report a consistent MTF execution timestamp if doing their own MiFID2 transaction reporting.

Execution Acknowledgement (BN)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
37	OrderId	R	String	Broker Order ID as identified on problem execution	May be any alphanumeric combination.
1036	ExecAckStatus	R	Char	Indicates the status of the execution acknowledgement	1 = Accepted 2 = Don't Know/Rejected
127	DKReason	N	Char	Conditionally required when ExecAckStatus = 2 (Don't know / Rejected)	
60	TransactTime	R	UTCTimestamp		Formatted as yyyyMMdd-HH:mm:ss.SSS
58	Text		String	Free format text string	

OrderStatusRequest (H)

Forward First workflow only. The **OrderStatusRequest** message is sent by the initiator to the acceptor to chase up an ExecutionReport(8) message which the winning bank should send as a response to a QuoteResponse(AJ) message.

OrderStatusRequest (H)						
Tag		Tag Name	Required, Conditional	Data Type	Description	Supported Values
131		QuoteReqID	R	String	Unique identifier of the RFQ negotiation.	
31984		BatchId		Integer	Unique identifier of Multibank Batch	Used if Order is part of Multibank Batch
31985		BatchCount		Integer	Number of Orders in Multibank Batch	Used if Order is part of Multibank Batch
117		QuoteId		String	Unique identifier for the quote	Used for Forward First workflow to reference back to the RFQ message
30472		StatusRequest Reason		Char	Enumerated value defining the reason for the OrderStatusRequest message	1 = RFQ Flow (chase ER reply to a QuoteResponse) 2 = Order flow (chase ER reply to a NewOrder message) =1 to be used here for RFQ workflow
55		Symbol	R	String	Currency pair to be traded	Formatted as RRR/YYY.
54		Side		Char	Side of order	1 = Client Buy 2 = Client Sell
30389		SpotRate		Price	Indicative Spot rate	
30390		ForwardPoints		Price	Agreed forward points, used for Forward First	
555		NumberOfLegs		Integer	Number of legs	For multileg Orders only
->	654	LegRefId		String	The unique identifier for this leg	Integer > 0
->	624	LegSide		String	The side for this leg	
->	30263	LegForwardPoints		Price	Agreed Forward Points for this leg	Used for Forward First

DontKnowTrade (Q)

The **DontKnowTrade** message is sent by the initiator to the acceptor and indicates the rejection of an **Execution Report (8)** by the initiator.

DontKnowTrade (Q)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
37	OrderID	R	String	Broker Order ID as identified on problem execution	May be any alphanumeric combination.
17	ExecID	R	String	Execution ID of problem Execution	May be any alphanumeric combination.
127	DKReason	R	String	Reason for reject	A = Unknown Symbol B = Wrong Side C = Quantity exceeds order D = No matching order E = Price exceeds limit F = Calculation difference Z = Other
58	Text		String	Reason for reject	

QUICKTRADE – ACCEPTOR MESSAGES**QuoteRequestReject (AG)**

The **QuoteRequestReject** is sent by the acceptor to the initiator to quote request has been rejected by the provider.

QuoteRequestReject (AG)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
131	QuoteReqID	C	String	Quote Request ID	May be any alphanumeric combination. Not required if QuoteAckStatus = 4
297	QuoteAckStatus	R	String	Status of the quote acknowledgement	1 = Canceled for Symbol(s) 2 = Canceled for Security Type(s) 3 = Canceled for Underlying 4 = Canceled All 5 = Rejected
300	QuoteRejectReason	C	Char	The reason the quote request was rejected.	1 = Unknown symbol (Security)

				Required if QuoteAckStatus(2 97)=5 (Rejected)	2 = Exchange (Security) closed 3 = Quote Request exceeds limit 4 = Too late to enter 5 = Unknown Quote 6 = Duplicate Quote 7 = Invalid bid/ask spread 8 = Invalid price 9 = Not authorized to quote security 11 = Insufficient credit 27 = Exceeded maximum notional order amount 99 = Other (Reject for credit reasons will have a reject code of 99-Other)
58	Text		String	Free form description of acknowledgement Description of the rejection reason if QuoteAckStatus(2 97) = 5	

Quote Cancel (Z)

The **Quote Cancel** message is sent by the acceptor to the initiator to cancel quotes. In the RFQ FIX implementation the QuoteCancel (Z) is used bi-directionally:

- Providers use this message to cancel an individual quote or all quotes associated with a quote request.

Quote Cancel (Z)					
Tag	Tag Name	Required	Data Type	Description	Supported Values
131	QuoteReqID	R	String	Unique identifier for the Quote Request to be cancelled.	May be any alphanumeric combination
31984	BatchId		Integer	Unique Identifier for the MultiBankBatch	Used if the RFQ is part of a MultiBank Batch
31985	BatchCount		Integer	Number of RFQs in the MultiBank Batch	Used if the RFQ is part of a MultiBank Batch
117	QuoteID	C	String	Unique identifier for the Quote	May be any alphanumeric combination. Conditionally required if QuoteCancelType = 5
298	QuoteCancelType		Char	Identifies the type of Quote Cancel Request	4 = Cancel for All Quotes 5 = Cancel quote specified in QuoteID

					/supported only on a provider quote cancel/ 99 – Other /supported only on reply to the customer request/ To be used in the cases where provider rejects request or provider is not reachable
146	NoRelatedSym	R	Integer	Number of related symbols in Request	Integer > 0
55	Symbol	R	String	Currency pair for which to cancel the quote	Formatted as RRR/YYY
60	TransactTime		UTCTimestamp		Formatted as yyyyMMdd-HH:mm:ss.SSS
58	Text		String	Free format text string	

Quote (S)

The **Quote (S)** is sent by the acceptor to quote a Quote Request.

Please note Providers must send one single Quote message for two-way RFQ requests, with both the Bid and Ask quotes contained within. The message below describes how to achieve that.

Quote (S)						
Tag	Tag Name	Required	Data Type	Description	Supported Values	Notes
131	QuoteReqID	R	String	Unique identifier for the RFQ negotiation	May be any alphanumeric combination	
30470	ProviderQuoteReference		String	This is an alphanumeric value that should not exceed 16 characters		Banks can use this field to identify the quote as opposed to QuoteID
117	QuoteID		String	Unique Identifier of the traded quote provided by provider		
31984	BatchID		Integer	Unique Identifier for the MultiBankBatch / Mixed Dealt		Used if the RFQ is part of Mixed Dealt or MultiBank Batch
31985	BatchCount		Integer	Number of RFQs in the MultiBank Batch / Mixed Dealt		Used if the RFQ is part of Mixed Dealt or MultiBank Batch
146	NoRelatedSym	R	Integer	Number of related symbols in Request	1	Only 1 supported in initial release
55	Symbol	R	String	Currency pair to be quoted	Formatted as RRR/YYY	
31022	Deliverable		Char	Indicates if the currency pair is deliverable	Y,N	Used for NDF and ND swaps

31997	Onshore		Char	Onshore/offshore flag for NDF transactions and deliverable forwards on ND currencies	Y,N	
54	Side		Char	The side of the request	1 = client buy 2 = client sell	The absence of a side implies a two-sided quote
167	SecurityType	R	String	The type of security being quoted	FXSPOT = FX Spot FXFWD = FX Forward FXSWAP = FX SWAP FXBLOCK = FX Single Spot Portfolio	Indicates type of security. If not present the Security Type is FXSPOT
1	Account		String	Account name for all legs on trade		If not supplied, allocations must be supplied for all trades
30413	AcctLEI	C	String	LEI for the account specified in Tag 1		
38	OrderQty	C	Qty	Amount of currency (tag 15) requested for the order		
15	Currency	C	String	Specify the required currency of the quoted price		Required for FXSPOT and FXFWD
40	OrdType		Char	Type of the order	Supported RFQ values: D = Quoted [Previously Quoted] order	
188	BidSpotRate	C	Price	Bid Spot Rate		Always populated for two-way RFQ's. Dependent on the Side field for standard RFQ
190	OfferSpotRate	C	Price	Offer Spot Rate		Always populated for two-way RFQ's. Dependent on the Side field for standard RFQ
189	BidForwardPoints	C	Price	Bid forward points		Always populated for two-way RFQ's. Dependent on the Side field for standard RFQ

191		OfferForwardPoints	C	Price	Offer forward points		Always populated for two-way RFQ's. Dependent on the Side field for standard RFQ
631		MidPx		Price	Indicates the optional Provider Mid Price		
555		NumberOfLegs	R	Integer	Number of legs in structure		
->	654	LegRefID	R	String	The unique identifier of this leg	Integer > 0	
->	600	LegSymbol	R	String	Security Symbol for the individual leg. See Symbol (55) field for description		
->	609	LegSecurityType		String	Indicates type of security	FXSPOT = FX Spot FXFWD = FX Forward /required/	
->	30250	LegAccount	C	String	Specify the leg's Account		
->	30252	LegAcctLEI	C	String	LEI for the account within individual leg		
->	687	LegQty	R	Amount	The amount to be traded on this leg		For Mixed Dealt, it is the sum of: 1. the allocation amount if the allocation currency is the same as the leg currency, 2. the allocation contra amount if the allocation currency is not the same as the leg currency
->	556	LegCurrency	R	String	The Dealt Currency for this leg		All legs of a strategy have the same dealt currency
->	31027	SpotDate		LocalMktDate	The Spot date for this leg		
->	31026	SplitValueDate		LocalMktDate	Stores the Split Value date for this leg if used		When settlement is split across two dates, one is stored in valueDate, and the other in splitValueDate
->	624	LegSide		Char	The side for this leg	1 = Client Buy 2 = Client Sell	The absence of a side implies a two-sided quote

->	611	LegMaturityDate	R	LocalMktDate	Leg Maturity Date	Formatted as YYYYMMDD	For NDFs this tag represents the fixing date of the contract therefore there is no need for an additional LegFixingDate (31018) tag
->	587	LegSettlType		String	Leg Settlement Type for Value Date Type field		
->	588	LegSettlDate		LocalMktDate	Leg Settlement Date		
->	675	LegSettlCurrency		String	Settlement Currency		
->	30386	LegBidSpotRate		Price	Bid Spot Price for this leg		Always populated for two-way RFQ's. Dependent on the Side field for standard RFQ
->	30387	LegOfferSpotRate		Price	Offer Spot Price for this leg		Always populated for two-way RFQ's. Dependent on the Side field for standard RFQ
->	1067	LegBidForwardPoints		Price	Bid forward points to be added to the BidSpotRate for this leg. May be a negative value		Always populated for two-way RFQ's. Dependent on the Side field for standard RFQ
->	1068	LegOfferForwardPoints		Price	Offer forward points to be added to the OfferSpotRate for this leg. May be a negative value		Always populated for two-way RFQ's. Dependent on the Side field for standard RFQ
->	6215	LegTenor		String	Leg Tenor		
->	31020	LegFixingSource		String	Fixing source of the Non-Deliverable currency pair		Indicates the FXall default NDF fixing setup for the currency pair Only set if 'Deliverable' tag 31022 is set to Y
->	31017	LegFixingTime		String	Fixing Time of the Non-Deliverable currency pair		Indicates the FXall default NDF fixing setup for the currency pair. Only set if 'Deliverable' tag 31022 is set to Y

->	566	LegPrice		Price	The price (All-In-Price) for the individual leg		
->	30262	LegSpotRate		Price	Spot Rate associated with this individual leg		Must be quoted to the requested DPS
->	30263	LegForwardPoints		Price	Forward Points associated with this individual leg. Required for NON-SPOT orders		Must be quoted to the requested DPS
->	670	NoLegAllocs		Integer	The number of allocations for this leg	Integer > 0	
->->	672	LegIndividualAllocationID		String	Unique identifier for this leg allocation		
->->	671	LegAllocAccount		String	The account mnemonic	The customer's allocation account	
->->	9255	LegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed		
->->	673	LegAllocQty		Amount	The amount to be allocated for this leg allocation		
->->	9212	LegAllocSide		Char	The side for this allocation	1 = Client Buy 2 = Client Sell	
->->	1367	LegAllocCurrency		String	The allocation dealt currency		
->->	9211	LegAllocContraAmount	C	String	Contra amount calculated using Provider Quoted rate		
60		TransactTime	R	UTCTimestamp		Formatted as yyyyMMdd-HH:mm:ss.SS	
58		Text		String	Free format text string		

Execution Report (8)

The **ExecutionReport** is sent by the acceptor to the initiator as a confirmation that a “Offer to Deal” has been accepted by the provider.

Execution Report (8)						
Tag	Tag Name	Required	Data Type	Description	Supported Values	Notes
131	QuoteReqID	R	String	Unique identifier for the RFQ negotiation	May be any alphanumeric combination	
117	QuoteID		String	Unique Identifier of the traded quote provided by provider		
297	TradeAckStatus	R	String	Status of the quote acknowledgement	1 = Accepted 5 = Rejected	
30472	StatusRequestReason		Char	Enumerated value defining the reason for the Execution Report response	1 = RFQ Flow (ER reply to a QuoteResponse) 2 = Order flow (ER reply to a NewOrder message)	Forward First RFQ flow should use “1”
103	OrdRejReason	C	Char	The reason the Trade request was rejected. Required if TradeAckStatus(297)=5 (Rejected)	4 = Too late to enter 5 = Unknown Quote 11 = Insufficient credit 99 = Other /Reject for Credit reasons will have a reject code of 99-Other/	
1328	RejectText		String	Reject reason		
55	Symbol	R	String	Currency pair to be quoted.	Formatted as RRR/YYY	
31022	Deliverable		Char	Indicates if the currency pair is deliverable	Y,N	Used for NDF and ND swaps
31997	Onshore		Char	Onshore/offshore flag for NDF transactions and deliverable forwards on ND currencies	Y,N	
54	Side		Char	The side of the request	1 = client buy 2 = client sell	Side of the request
167	SecurityType	R	String	The type of security being quoted	FXSPOT = FX Spot FXFWD = FX Forward /required/	Indicates type of security. If not present the Security Type is FXSPOT

					FXSWAP = FX SWAP /required/ FXBLOCK = FX Single Spot Portfolio	
1	Account		String	Account name for all legs on trade		If not supplied, allocations must be supplied for all trades
30413	AcctLEI	C	String	LEI for the account specified in the Tag 1		
38	OrderQty	C	Qty	Amount of currency (Tag 15) that is requested for the order		
15	Currency	C	String	Specify the required currency of the quoted price		Required for FXSPOT and FXFWD
4783	SettlLocation	C	String	Precious Metals Settlement Location - Informational field	Set to LON only	Used for Precious Metal trades only, applies to spot or offvenue
9239	BlockTrade		Char	If trade is above block trade size limit	Y,N	
9254	FinancialEntityDefinition		String	If a SEF trade, the Dodd-Frank financial entity definition code for the party requesting the trade		
9256	SDRName		String	If a SEF trade, the name of the swap data repository that trades will be reported to		
30424	SecurityConversion		Char	Security Conversion Flag. Related to specific Foreign Exchange transaction defined by CFTC	Supported values: BLANK 'N' 'Y' If Blank /not defined by the client, then tag should not be displayed/	Required for SEF Trade
25044	CustomerLEI		String	Customer LEI		Required for SEF / MTF Trade
31345	NPFT		Boolean	Non-Price Forming Trade Flag	Y,N	Required for MTF Trade
9240	SEF		Char	Denotes whether the trade was executed on FXall SEF	Y,N	Required for SEF Trade
31344	TradeCapacity		String	Customer Trade Capacity	MTF: DEAL, MTCH, AOTC SEF: PRINCIPAL AGENCY:	Required for MTF and SEF Trades
453	NoPartyIDs		Integer	Repeating group below should contain unique combinations of	Integer > 0	Required for MTF Trade

					PartyID, PartyIDSource and Party Role.		
->	448	PartyID		String	Party identifier code	Required if NoPartyIDs(453) > 0	See Appendix for supported values.
->	447	PartyIDSource		Char	Identifies class or source of the PartyID (448) value.	Required if NoPartyIDs(453) > 0	See Appendix for supported values.
->	452	PartyRole		integer	Identifies the type or role of the PartyID (448) specified	Required if NoPartyIDs(453) > 0	See Appendix for supported values.
	555	NumberOfLegs	R	Integer	Number of legs in structure		
->	654	LegRefID	R	String	The unique identifier of this leg	Integer > 0	
->	600	LegSymbol	R	String	Security Symbol for the individual leg. See Symbol (55) field for description		
->	609	LegSecurityType		String	Indicates type of security	FXSPOT = FX Spot FXFWD = FX Forward /required/	
->	30250	LegAccount	C	String	Specify the Leg's Account		
->	30252	LegAcctLEI	C	String	LEI for the account within individual leg		
->	687	LegQty	R	Amount	The amount to be traded on this leg		
->	556	LegCurrency	R	String	The Dealt Currency for this leg		All legs of a strategy have the same dealt currency
->	31027	SpotDate		LocalMktDate	The Spot date for this leg		
->	31026	SplitValueDate		LocalMktDate	Stores the Split Value date for this leg if used		When settlement is split across two dates, one is stored in valueDate, and the other in splitValueDate
->	624	LegSide	R	Char	The side for this leg	1 = Client Buy 2 = Client Sell	
->	611	LegMaturityDate	R	LocalMktDate	Leg Maturity Date	Formatted as YYYYMMDD	For NDFs this tag represents the fixing date of the contract therefore there is no need for a LegFixingDate (31018) tag
->	587	LegSettleType		String	Leg Settlement Type for Value Date Type field		

->	588	LegSettlDate		LocalMktDate	Leg Settlement Date	Formatted as YYYYMMDD	
->	675	LegSettlCurrency		String	Settlement Currency		
->	6215	LegTenor		String	Leg Tenor		
->	31020	LegFixingSource		String	Fixing source of the Non-Deliverable currency pair		Indicates the FXall default NDF fixing setup for the currency pair Only set if 'Deliverable' tag (31022) is set to Y
->	31017	LegFixingTime		String	Fixing Time of the Non-Deliverable currency pair		Indicates the FXall default NDF fixing setup for the currency pair Only set if 'Deliverable' tag (31022) is set to Y
->	566	LegPrice		Price	The price (ALL-IN-Price) for the individual leg		
->	30262	LegSpotRate		Price	Spot Rate associated with this individual leg		Must be quoted to the requested DPS
->	30263	LegForwardPoints		Price	Forward Points associated with this individual leg. Required for Non-Spot orders		Must be quoted to the requested DPS
->	2346	LegMidPR		Price	Leg Mid price/rate		Provider Mid price on SEF trades
->	9253	UPIValue		String	Contains the unique product identifier for the trade being executed	FRNDFO	Used in SEF trades only (NDF)
->	9257	ReportingParty		String	If a SEF trade, contains whether the client or provider has the responsibility to report this leg	T = Taker (Client), M = Maker (Provider)	Required for SEF trade
->	30255	LegAcctUSI		String	Transaction Identification Code for the leg		
->	30442	LegNpft		Boolean	Non-Forming Price Tag at allocation level	Y,N	Required for MTF trade
->	30463	LegSecFin		Boolean	Security Financing flag of the allocation. Only used on forwards allocations	'N' = Default 'Y'	Required for MTF trade

->	670	NoLegAllocs		Integer	The number of allocations for this leg	Integer > 0	
->->	672	LegIndividualAllocID		String	Unique identifier for this leg allocation		
->->	671	LegAllocAccount		String	The account mnemonic	Customer's allocation account	
->->	9252	LegAllocUSIValue		String	Contains the unique swap identifier that will be reported on the trade. The provider can override the FXall provided value with their value when they return an Execution Report		
->->	9255	LegAllocAccountLEI		String	Contains the legal entity identifier for the account that the trade is being executed		
->->	30462	LegAllocNpft		Boolean	Non-Forming Price Tag at allocation level	Y,N	
->->	30423	LegAllocSecFin		Boolean	Security Financing flag of the allocation. Only used on forwards allocations	'N' = Default 'Y'	
->->	673	LegAllocQty		Amount	Amount to be allocated for this leg allocation		
->->	9212	LegAllocSide		Char	The side for this allocation	1 = Client Buy 2 = Client Sell	
->->	1367	LegAllocCurrency		String	The allocation dealt currency		Required for MTF
->->	9211	LegAllocContraAmount	C	String	Contra amount calculated using Provider Quoted rate		
->->	30426	AllocSecurityConversion		String	Security Conversion Flag of the underlying account. Related to specific Foreign Exchange transaction defined by CFTC Supported	BLANK 'N' 'Y' If the value is Blank /not defined by the client, then tag should not be displayed/	Client may specify whether the trade was a securities financing trade, e.g. set the flag to 'true' = 'Y'
60		TransactTime	R	UTCTimestamp		Formatted as yyyyMMdd-HH:mm:ss.SSS	
58		Text		String	Free Format Text String		

Appendix

QUICKTRADE

Security Types

The table below defines the Security Types that can be traded via the RFQ FIX API.

Security Type	Field 167	Supported
Spot	FXSPOT	Yes
Forwards	FXFWD	Yes
SWAP	FXSWAP	Yes
SSP	FXBLOCK	Yes

The RFQ FIX API supports single account and multi-allocation trading. (See Account and Allocations Usage and Examples section for full description.)

The “acct@entity” format is used in account fields. Multi-allocation orders use Tag1 = PREALLOCATED and the “acct@entity” format in allocation account fields.

Spot and Forwards use the PreAllocGrp component to identify allocation accounts.

Field	Field Name	Value	Description
78	NoAllocs	2	Number of pre-trade allocation accounts
79	AllocAccount	acct1@entity	The allocation account
79	AllocAccount	acct2@entity	The allocation account

SWAPS use the LegPreAllocGrp component to identify allocation accounts.

Field	Field Name	Value	Description
670	NoLegAllocs	2	Number of pre-trade allocation accounts
671	LegAllocAccount	acct1@entity	The allocation account
671	LegAllocAccount	acct2@entity	The allocation account

Request for Quote States

Once successfully initiated, a multi-provider RFQ will last for 120 seconds (two minutes). There is no timeout period for a single provider RFQ.

The table below describes the basic states that can occur during the RFQ process.

User Action	Subsequent Action	RFQ State
Client sends a New Order [D or AB]	Order is filled	RFQ is terminated
Client sends a New Order [D or AB]	Order is rejected/canceled	RFQ is terminated
Client sends a Quote Cancel [Z] for a QuoteReqID	Quote Cancel [Z] for that QuoteReqID is sent to all providers	RFQ is terminated with all providers for that QuoteReqID

User Action	Subsequent Action	RFQ State
Client subscribes to <u>multiple providers</u> and takes no action for 120 seconds (two minutes)	Quote Cancel [Z] is automatically sent to all providers for that QuoteReqID	RFQ is terminated with all providers for that QuoteReqID
Client subscribes to <u>only one provider</u> and takes no action for 120 seconds (two minutes)	None	RFQ continues
Provider sends a Quote Cancel [Z] for all quotes	RFQ is terminated with that provider for that QuoteReqID	RFQ continues if multi-provider
Provider sends a Quote Cancel [Z] for a specific QuoteID	The individual quote is canceled (withdrawn)	RFQ continues

Deprecated Features and Tags

The use of `SettlCurrency` (Tag = 120) has been deprecated in RFQ FIX API Version 2.0 and is being phased out. The currency associated with a price is reported in `Currency` (Tag = 15). `SettlCurrency` (Tag = 120) is a non-mandatory tag that will continue to be returned for backwards compatibility.

The use of `LegSettlCurrency` (Tag = 675) has been deprecated in RFQ FIX API Version 2.0 and is being phased out. The currency associated with a particular leg's quantity is reported in `LegCurrency` (Tag = 556) and clients should use this tag instead. `LegSettlCurrency` (Tag = 675) is a non-mandatory tag that will continue to be returned for backwards compatibility.

The use of `LegAllocSettlCurrency` (Tag = 1367) has been deprecated in RFQ FIX API Version 2.0 and is being phased out. The RFQ FIX API does not support mixed dealt swaps and the allocation currency must be the same as the leg currency. Therefore clients should use `LegCurrency` (Tag = 556) as a proxy for leg allocation currency. `LegAllocSettlCurrency` (Tag = 1367) is a non-mandatory tag that will continue to be returned for backwards compatibility.

OPTIONS

Supported values for Expiry Cut

(defined in the FXall Production environment at time of writing this document)

EXPIRY_CUT_ID	CUT_NAME	CUT_TIME	CUT_LOCATION	TIME_ZONE
1	NEW_YORK	1000	NY	America/New_York
2	TOKYO	1500	TK	Asia/Tokyo
3	MERICO	1230	NY	America/Mexico_City
4	EMTA	1330	EMTA	Europe/Berlin
5	EMTA_BRL	1315	EMTA	America/Sao_Paulo
6	EMTA_KRW	1530	EMTA	Asia/Seoul
7	EMTA_CNY	915	EMTA	Asia/Shanghai
8	WARSAW	1100	WA	Europe/Warsaw
9	11SING	1100	SG	Asia/Singapore
10	HUNGARY	1200	HU	Europe/Budapest
11	TURKEY	1200	LN	Asia/Istanbul
12	MUMBAI	1300	MU	Asia/Calcutta
13	MOSCOW	1630	RU	Europe/Moscow
14	ISRAEL	1500	TE	Asia/Tel_Aviv
15	18SAOPAULO	1800	EMTA	America/Sao_Paulo
16	WMR16	1600	LN	Europe/London
17	EMTA11	1100	EMTA	Europe/London
18	ECB37	1415	FR	Europe/Frankfurt
19	HONG_KONG	1115	HKD	Asia/HongKong
20	MOSCOW_1230	1230	RU	Europe/Moscow
22	EMTA_COP	1030	EMTA	America/Bogota
23	MANILA	1130	MN	Asia/Manila
24	MALAYSIA	1000	KL	Asia/Kuala_Lumpur
25	TAIWAN	1100	TP	Asia/Taipei
27	TOK_0900	900	TK	Asia/Tokyo
28	RAU_1030LN	1030	LN	Europe/London
29	RAU_0930NY	930	NY	Europe/London
30	RAG_1200LN	1200	LN	Europe/London
31	BFIX_1	1300	LN	Europe/London
32	MUMBAI1330	1330	MU	Asia/Calcutta
33	EMTA10	1000	EMTA	Europe/London
34	EMTA_PHP	1130	EMTA	APAC/Manilla
51	BOGOTA	1030	BOGOTA	America/Bogota
91	CLSA	1030	CLSA	America/Santiago
111	IDJA	1000	IDJA	Asia/Indonesia
131	WMR15	1500	LN	Europe/London
151	HONG_KONG_1130	1130	HKG	Asia/HongKong
152	HONG_KONG_2PM	1400	HKD	Asia/HongKong

REGULATORY

Supported values for NoPartyIDs

Venue is MTF

Party Group	Tag	Field	Values
Order Venue	448	PartyID	MTF
	452	PartyRole	73 – Execution Venue
	447	PartyIDSource	D – Proprietary/Custom Code
Order MIC	448	PartyID	FXRQ
	452	PartyRole	64 – Multilateral Trading Facility
	447	PartyIDSource	G – MIC (Market Identifier Code)
Client ID	448	PartyID	Liquidity Taker Name
	452	PartyRole	3 – Client ID
	447	PartyIDSource	D – Proprietary/Custom Code

Venue is SEF (changes currently scheduled to take effect in May 2022)

Party Group	Tag	Field	Values
Order Venue	448	PartyID	SEF
	452	PartyRole	73 – Execution Venue
	447	PartyIDSource	D – Proprietary/Custom Code
Order MIC	448	PartyID	THRE
	452	PartyRole	73 – Execution Venue
	447	PartyIDSource	G – MIC (Market Identifier Code)
Client ID	448	PartyID	Liquidity Taker Name
	452	PartyRole	3 – Client ID
	447	PartyIDSource	D – Proprietary/Custom Code

Venue is Offvenue, Spot or None (changes currently scheduled to take effect in May 2022)

Party Group	Tag	Field	Values
Order Venue	448	PartyID	<i>Offvenue / Spot / None</i>
	452	PartyRole	73 – Execution Venue
	447	PartyIDSource	D – Proprietary/Custom Code
Order MIC	448	PartyID	RTSP
	452	PartyRole	73 – Execution Venue
	447	PartyIDSource	G – MIC (Market Identifier Code)
Client ID	448	PartyID	Liquidity Taker Name
	452	PartyRole	3 – Client ID
	447	PartyIDSource	D – Proprietary/Custom Code

ADDITIONAL SUPPORTED WORKFLOWS

Forward First RFQ

The workflow is similar to the QuickTrade Cash RFQ [workflow](#), however for Forward First the winning bank will get a QuoteResponse(AJ) message with a QuoteRequestType of 7 (End Trade) along with QuoteID. The subsequent 'NewOrder' Fixing message will contain the agreed points. The other banks will receive a QuoteCancel(Z) message.

This functionality will require the client to connect to two API's - the RFQ API will be used for the initial RFQ process, before the actual Fixing order is sent through the Fixing API

