FIX 5.0 SP2 Rules of Engagement

V2.0.20

Historia del documento

Fecha	Versión	Descripción	Autor
09/12/2010	1.0	First version of the R.O.E in which you'll find the market data messages	Primary
11/02/11	1.1	R.O.E is updated to FIX 5.0 SP02	Primary
25/02/2011	1.2	Contains Order Management and Security Definition	Primary
30/02/2011	1.3	Updated messages: Logon, New Order – Single, Market Data – Snapshot / Full Refresh	Primary

31/01/2012	2.0	Added messages:
		• Resend Request
		• Test Request
		• TradingSessionStatus
		• Order Status Request
		• OrderMass Cancel Report
		Changed messages;
		• In Logon message username and password is no longer required
		• News message add URLLink field.
		• Business Message Reject removes the fields BusinessRejectRefID, RefSeqNum and adds the options 1 to 7 and 18 to the BusinessRejectReason field.
		• New order single message adds :
		  ○ The option Market and Market with left over as limit order types.
		  ○ The All or None option to the ExecInst field.
		  ○ The TIFs GTC GTD and FOK.
		  ○ The Security Exchange field.
		• Order Cancel Request message changes: the OrderQty was removed and transactime and CLOrdID fields changes to not required, the Security Exchange and Account fields were added, and OrdId and OrigClOrdID are now conditionally required.
		• Order Cancel/Replace Request message adds the Transactime required field and Time in Force as not required field. The Price field is not required now, and the ExecInst field was removed from message.
		• Order Cancel Reject message adds options New, Partially Filled, Filled and Canceled to the ordStatus field. TransactTime and text field is no longer required. The field CxlRejReason adds the options Other and Too Late To Cancel-.
		• Order Mass Cancel Request adds the fields ClOrdID, TransactTime, Symbol SecurityExchange, Side, CFICode, and Parties Block data.
		• Execution Report adds the Market and Market with Left Over as
			Primary

		Limit OrderType for some response messages. OrigClOrdID and price fields are now conditionally required. TIF adds the GTD, GTC and FOK options. • The ExecutionReport message in response to OrderStatus request adds the field OrdStatusReqID. • The Market Data Snapshot full refresh changes the field MDReID to required and adds the fields MDEntryDate, MDEntryTime, TradeCondition, TradeSide, TradingSessionID, LotType, MinLotSize not required or conditionally required. The OrderID, MDEntryBuyer, MDEntrySeller, MDEntryPositionNo, and TrdType fields were removed. The MDEntryType = C, changes the type of entry from MDEntryPx to MDEntrySize, in accordance with the FIX specification. • The Security List message adds the fields MarketID and MarketSegmentID, both are not required, and removes the field Product. The CFICode field adds Bond, Swap and Future Spread options. • The Security List Response message adds the fields MarketID and MarketSegmentID, SecurityExchange, MinTradeVol, MaxTradeVol, UnderlyingSymbol, ExecInstValue and TimeInForce, OrdID not required and the TotNoRelatedSym, and LastFragment fields required. The RofexProps field was removed. The field PriceLowLimit and PriceHighLimit were replaced for LowPriceLimit and HighPriceLimit respectively and changes your tag number id.	
20/04/2012	2.0.1	In message D = "New Order Single", field 18 = ExecInst, is not longer required	Primary
07/05/2012	2.0.2	In message F = "Order Cancel Request", field 1 = Account, now becomes required.	Primary
15/05/2012	2.0.3	In the message q = "Order Mass Cancel Request" in field 530= MassCancelRequestType the types: • 2 = Cancel orders for an underlying security, • 3 = Cancel orders for a Product, • 5 = Cancel orders for a SecurityType, • 6 = Cancel orders for a trading session, • 8 = Cancel orders for a market, • 9 = Cancel orders for a market segment, • A = Cancel orders for a security group, • B = Cancel for Security Issuer, • C = Cancel for Issuer of Underlying Security are not supported, while the types: • 1 = Cancel orders for a security, • 4 = Cancel orders for a CFICode, • 7 = Cancel all orders	Primary

		are supported.	
13/06/2012	2.0.4	In message AF = “OrderMassStatusRequest” the fields 1= Account, and 207= SecurityExchange, are added. In message q = “OrderMassCancelRequest” the field 1300=MarketSegmentID was added. In message W =”Market Data Snaphot Full Refresh” the field 264=MarketDepth was added. In message y = “SecurityList”, the fields 559=SecurityListRequestType, and 9996= ContractPositionNumber, were added. In message 8= “ExecutionReport”, the field 18=ExecInst, was added.	Primary
10/08/2012	2.0.5	In message ExecutionReport=8 • when ExecType= I (Order Status) : o Field 39=OrderStatus adds the option “8” Rejected and eliminates the option “5” Replaced (which is deprecated in the fix protocol since version 4.3), It is clarified that this message is used both as response to the messages "OrderMassStatusRequest" and "OrderStatusRequest". • when the ExecType= I (Order Status) and OrderStatus = 8 o Field 103=OrdRejectReason was added. o Fields 40= OrderType, 31=LastPx and 32=LastQty are no longer sent. • When the ExecType = 8 o Fields 40= OrderType, 31=LastPx and 32=LastQty are no longer sent. o If OrderStatus=8 the grounds for rejection could be sent in the field 58=Text. In the Execution Reports messages added the Parties Group and some values in the fields when they are sent by default.	Primary
09/11/2012	2.0.6	In message y=“Security List”, fields 863 and 9996 have been corrected in the description to reflect that fragmented messages are sent per segment. In message W=“Market Data Snapshot Full Refresh” fields 290=MDEntryPositionNo and 828= TrdType that were previously omitted in error were added.	Primary

20/12/2012	2.0.7	In message D = "New Order-Single", the description of the field =18 was corrected, and in field= 59 GTD was an option was removed since it is not currently supported. In message y=”Security List” description of the field =461 was corrected.	Primary
16/01/2013	2.0.8	In message "W = Market Data Snapshot Full Refresh" the precision of 273=MDEntryTime field was changed, to add millisecond-level precision.	Primary
21/02/2013	2.0.9	Connection information and architecture was added. The descriptions of some messages such as: news, business message reject, Trading Session Status, and Order Cancel Request were improved for clarity. Information about typography and syntax conventions was added. Customized fields are now identified in messages. Message y = “SecurityList” was re-ordered. Messages “3”= Reject and “j”= Business message reject, had the description of the field =58 corrected for clarity and accuracy. Message "W = Market Data Snapshot Full Refresh" trade information was improved. In message F = “Order Cancel Request”, the OrderQtyData block was added, and fields 11=ClientOrderID and 60=TransactTime were changed to required. In message G = “Order Cancel/Replace Request” the OrderQtyData block was added, field 1=Account, was changed to required, and some ID fields descriptions have been improved. In message q = “Order Mass Cancel Request” , field 461=CFICode and the Instrument block were changed to conditionally required. In messages Execution Report (MsgType = 8): Order Canceled Response, Execution Report (MsgType = 8): Order Status Response – With orders, and Execution Report (MsgType = 8): Order Replaced Response, fields LastPx and LastQty were changed to conditionally required for more accuracy, and some field descriptions were improved.	Primary
03/05/2013	2.0.10	In “message type x = SecurityListRequestMessage”, field 559=SecurityListRequestType was simplified, to meet a single requirement. Message “r=OrderMassCancelReport” is not supported any more.	Primary

		The market trading hours were added as well as information to allow routing to external markets.	
15/11/2013	2.0.11	“Parties Block” was changed to add some "Party Roles" fields needed for Argentine Markets interconnection. Additional information about FIX sessions for Argentine market interconnection were introduced in this version.	Primary
09/01/2014	2.0.12	In message New Order Single = “D”, field Account=1 was changed to required. This version adds information about post-trading messages supported by the Exchange to implement Block Trade, Allocation, and Give Up functionality.	Primary
21/02/2014	2.0.13	In message type OrderMassCancelRequest=”q”, the description of field MarketSegmentID (1300) was changed to reflect reality more accurately. In message type OrderCancelReplaceRequest=”G” the field TimeInForce (59) was changed to Conditionally required to make clear its usage.	Primary
26/06/2014	2.0.14	In the message type SecurityList=”y”, the field Factor (228) was added to specify the Contract Value Factor by which price must be adjusted to determine the true nominal value of one derivatives contract. In the message type MarketDataSnapshotFullRefresh =”W”, the field MDEntryType (269) adds two new data volumes: x (nominal volume), and w (cash volume). In the message type NewOrderSingle=”D”, the field OrderType (40) adds new values to allow sending stop orders.	Primary
11/08/2014	2.0.15	In the message type "TradingSessionStatus = h", some changes were made in the TradingSessionSubID (652) field to be able to report the suspension of a market segment.	Primary
08/10/2014	2.0.16	In the message type "TradingSessionStatus = h", was added the field Text (58) to indicate those segments not belonging to ROFEX. In this version were added two new uses for the “Trade Capture Report” messages, one of them for request trade capture reports by account and the other for request trade capture reports by symbol. Some visual improvements were made to highlight the sample tags belonging to blocks.	Primary

26/01/2014	2.0.17	The list of values availables of the field =461 (CFICode) was expanded and some values were changed. In the message type “OrderCancelReplaceRequest” = G changes were made to the way a successful order's replacement, due to that the replaced order will be reported with a new id, while the order replaced will be canceled. In the message type “OrderMassStatusRequest” = AF the field SecurityStatus was added to allow to consult the states of all orders. Security Status Request and Security Status messages description were added.	Primary
13/04/2015	2.0.18	In the message type "News = B", the field MarketSegmentID (1300) was added to indicate the market segment target for the news message. In the message "SecurityList = y" the field MaxLotSize (5515) was added to inform the max lot size allowed for Block Trade orders. The field CFICode (461) adds type Index.	Primary
04/05/2015	2.0.19	In the message “SecurityList = y” the field EndDate (917) was added, to indicate the liquidation date for instruments of type Repurchase.	Primary
02/10/2015	2.0.20	The message “OrderCancelRequest =F” was modified to require only the OrderID field to cancel an order, It is no longer necessary to include the ClOrdID or OrigClOrdID tags.	Primary

Índice

Historia del documento 2
Introduction 11
Purpose and Document Scope 11
References to other documents 11
Typographical and Syntax Conventions 12
Connection information 13
Market Trading Hours 13
FIX Version used 13
Identification of the FIX session 13
IP Addresses 14
TCP Port Number 14
FIX Session Assignment 14
Identification of Instruments 14
Customer Support 14
Architecture 15
Common Components Blocks of Application Messages 16
Instrument Identification 16
Counter Party Identification 16
OrderQtyData Identification 17
Underlying Instrument Group Identification 17
Position Amount Data Identification 18
Position Qty Identification 18
RootParties Identification 19
Alloc Group Identification 19
OrdAlloc Group Identification 20
Messages 21
Header and Trailer 21
Standard Message Header 21
Standard Message Trailer 22
Message summary 23
Summary of supported Messages 23

Session Layer Messages 25
Message Specification 25
Logon (MsgType = A) 25
Heartbeat (MsgType = 0) 25
Test Request (MsgType = 1) 26
Resend Request (MsgType = 2) 26
Reject – Session Level (MsgType = 3) 26
Sequence Reset (MsgType = 4) 28
Logout (MsgType = 5) 28
Common messages 29
Message Specification 29
News (MsgType = B) 29
Business Message Reject (MsgType = j) 30
Message Flow – Common messages 30
Application Messages – Market and Trading Session Status 31
Message Specification 31
Trading Session Status (MsgType = h) 31
Application Messages - Order Management 32
Message Specification 32
New Order – Single (MsgType = D) 32
Order Cancel Request (MsgType = F) 34
Order Cancel/Replace Request (MsgType = G) 35
Order Cancel Reject (MsgType = 9) 38
Order Status Request (MsgType = H) 39
Order Mass Status Request (MsgType = AF) 40
Order Mass Cancel Request (MsgType =q) 41
Execution Report (MsgType = 8): New 43
Execution Report (MsgType = 8): Order Canceled Response 45
Execution Report (MsgType = 8): Order Replaced Response 47
Execution Report (MsgType = 8): Order Filled/Partially Filled Response 49
Execution Report (MsgType = 8): Order Status Response – No orders 51
Execution Report (MsgType = 8): Order Status Response – With orders 53
Execution Report (MsgType = 8): Reject Message Response 55

Message Flow - Order Management 56
New Order – Single 56
Order Cancel Request 57
Order Cancel/Replace Request 58
Order Status Request 59
Application Messages – Market Data 60
Message specification 60
Market Data Request (MsgType = V) 60
Market Data – Snapshot / Full Refresh (MsgType = W) 61
Market Data Request Reject (MsgType = Y) 64
Message Flow - Market Data 66
Application Messages – Security Definition 69
Message specification 69
Security List Request (MsgType = x) 69
Security List (MsgType = y) 71
Security Status Request (MsgType = e) 76
Security Status (MsgType = f) 77
Message Flow - Security Definition 78
Application Messages – Post trade messages 79
Message specification 79
Trade Capture Report Request (MsgType = AD): for Regular Trades by Account 79
Trade Capture Report (MsgType = AE): for Regular Trades by Account 81
Trade Capture Report Request (MsgType = AD): for Regular Trades by Symbol 83
Trade Capture Report (MsgType = AE): for Regular Trades by String 84
Trade Capture Report (MsgType = AE): for Block Trades 86
TradeCaptureReportAck (MsgType = AR): for Block Trades 88
Trade Capture Report Request (MsgType = AD): for allocations and giveups 89
Trade Capture Report (MsgType = AE): for allocations and giveup 90
AllocationInstruction (MsgType = J) 93
AllocationInstructionAck (MsgType = P) 95
Confirmation (MsgType = AK): for allocations and giveups 96
ConfirmationAck (MsgType = AU) 99

Introduction

Purpose and Document Scope

ROFEX (*Rosario Futures EXchange*) released this document to provide an entry mechanism for market data subscription and order routing to its Electronic Trading Exchange using the FIX Protocol. The ROFEX Trading System provides the hardware and software needed to connect to its Electronic Derivatives Exchange. The goal of this document is to describe the message types and tags supported to successfully connect to the ROFEX FIX 5.0 SP2 interface.

References to other documents

For detailed information on each of these fields, please refer to the *FIX Protocol* specifications. ***www.fixprotocol.org.***

Typographical and Syntax Conventions

This document uses certain typographical conventions:

- ***Text in this style*** is used for identify Blocks of Data of any kind.
- → is used for indicates one level of depth in blocks of data, for example, ***Block Intrument***.
- →→ is used for indicates two levels of depth in blocks of data.
- →→→ is used for indicates three levels of depth in blocks of data.
- →→→→ is used for indicates four levels of depth in blocks of data.
- **<*125*>** a tag number enclosed between major and minor signs indicate that the field is a "Rofex' Custom field".

Conventions for tables:

Tag	FIX Name	Req	Format	Description
Tag Number	Field Name according to the Fix Protocol	Indicates if the field is required, possible values: Y: yes N: no C: conditionally required	field format used	Description of Use
→	Block Name	Idem		Idem
Tag number	NumInGroup	Idem	Field format used	Idem. Used as an example and shows how the fields must be completed
→→ ***Tag number***	*Field name*	*Idem*	*Field format used*	
→→ ***Tag number***	*Field name*	*Idem*	*Field format used*	

POWERED BY PRIMARY

Connection information

Market Trading Hours

FIX Session Hours: 9.00am-7pm

Trading Hours: 10:00am-6pm

FIX Version used

The version protocol used is Fix 5.0 SP2.

Identification of the FIX session

The exchange will provide every member with an Exchange Code, Member Code, Login Username and a Password. All messages sent by the member to the exchange should contain the provided Member Code in the *SenderCompID* and *OnBehalfOfCompID* fields. For test connections, the Member will be provided with a separate Exchange Code, Member Code, Login Username and Password. All messages sent by the Member to the exchange should also have the *TargetCompID* field set to the exchange's code provided, and will have to be set to "ROFX". No more than one FIX session can exist at the time with the same values for these fields. If a message is received with values that do not correspond with those of the session, it will be rejected and the connection closed. It should be noted that the values of these fields are inverted when the message is sent by the exchange, with respect to those sent by the client.

Interconnection agreements between markets

It is defined, according with practices used in other marketplaces, that to send orders to other markets or take orders from them, a different FIX session will be used for each direction.

To send orders from Primary to another market, Primary act as initiator of a session to that market, using the FIX version and dictionary provided by the other Exchange. Those wishing to submit market orders to Primary, must start a session, in which Primary act as "acceptor" using the FIX version, dictionary and specification provided by Primary.

This allows both markets to connect, regardless of differences between the dictionaries and specifications.

IP Addresses

All Members connecting to the system will be provided with a production DNS name and one or more test DNS names.

TCP Port Number

Primary's Router listens for Member connections on a TCP specific port number. This port number will be also provided by the exchange.

FIX Session Assignment

FIX comp IDs and IP addresses/DNS names for connection are assigned by Primary to connecting counterparties. The process is differentiated according to the counterparty category (banks, trading firms, vendors, other exchanges, etc). For more details, please contact Primary.

Identification of Instruments

The instruments are identified by the "symbol", which is unique in each market.

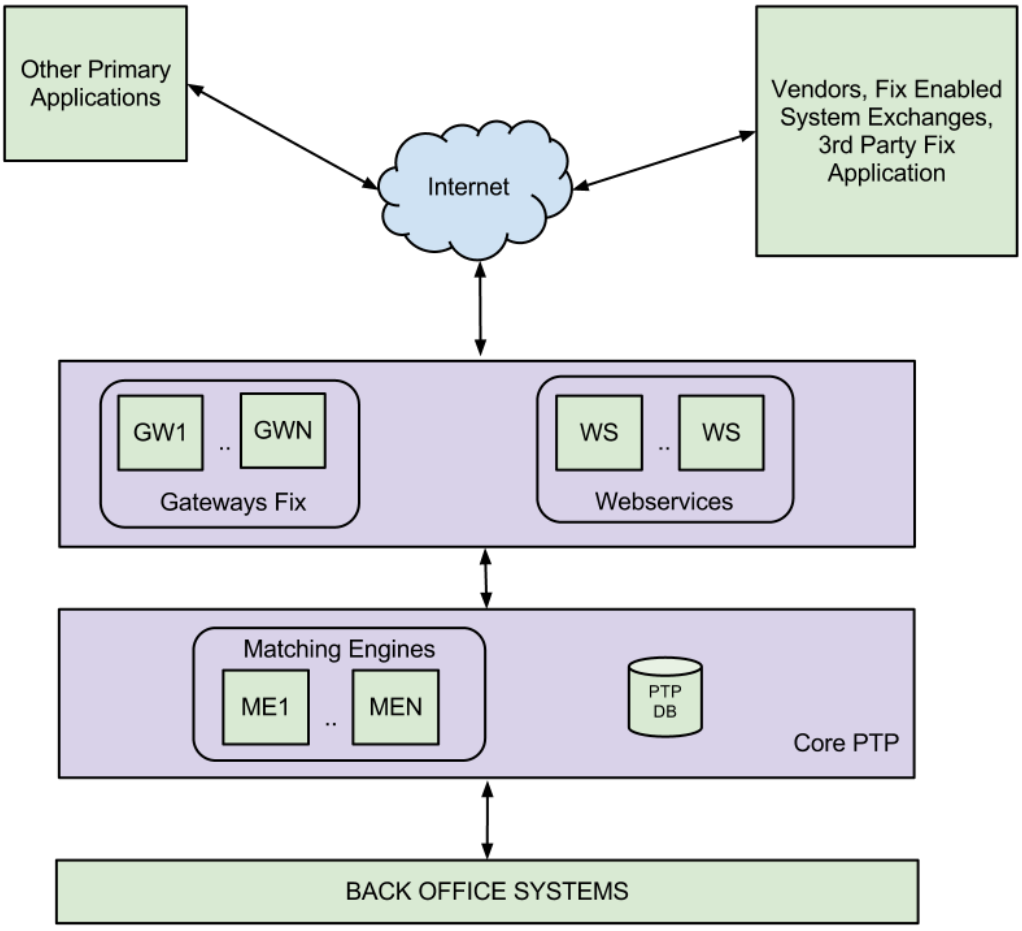
Customer Support

Technical Support

25 de Mayo 460 Piso 3
Buenos Aires, Argentina
+54 11 51999851/2
soporte@primary.com.ar

Architecture

External and internal clients will be connected to a “FIX Gateway” in a DMZ (by DNS –based load balancing) through a FIX session. The protocol version used is Fix 5.0 SP2. The messages will be routed to PTP Core for processing, and then the response will return from "FIX Gateway" to Clients.



Common Components Blocks of Application Messages

Instrument Identification

Instruments are uniquely identified using the block of fields presented below.

Tag	FIX Name	Req	Format	Description
→	*Block Instrument*			Set of "Instruments"
146	NoRelatedSym	Y	NumInGroup (Int)	Specifies the number of repeating symbols (instruments) specified
→→ ***55***	*Symbol*	*Y*	*String*	*The Symbol Name*
→→ ***207***	*SecurityExchange*	*N*	*String*	*Security exchange identifier. Value defined **ROFX***

Counter Party Identification

The Parties block is used in many application messages to specify the parties involved in the transaction. In the detailed definition of the messages that contains this block, the block is incorporated exactly as shown below. The list of possible values is restricted by the specific characteristics of the message.

Tag	FIX Name	Req	Valid Values	Format	Description
→	*Block Parties*				Set of "Parties"
453	NoPartyIDs	N	>0	NumInGroup (Int)	Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole.
→→ ***448***	*PartyID*	*N*		*String*	*Member code*
→→ ***447***	*PartyIDSource*	*C*	*D = Propietary custom code*	*Char*	*Required if NoPartyIDs has been specified*
→→ ***452***	*PartyRole*	*C*	*3 = Client ID* *24 = Customer Account* *11= Order Origination Trader* *(see description below for external markets)*	*Int*	*Indicates the role taken by the code specified in PartyID.* *Required if NoPartyIDs has been specified*
→→ ***<109>***	*ClientID*	*N*		*String*	*ClientID of order sender/modifier, related to ClOrdID field.*

POWERED BY PRIMARY

greements between markets

At least 4 values must be sent when submitting orders for routing to other Exchanges, using the repetitive group PartyIDs

Values added to the field PartyRole:

4 = Clearing Firm (clearing and settlement agent)

1 = Executing Firm (Negotiation agent)

12 = Executing Trader (Trader)

76 = DeskID (Terminal), currently not used by Primary.

Note that Clearing Firm and Executing Firm should use the values centrally-provided by CNV ("Comisión Nacional de Valores").

OrderQtyData Identification

Set of "OrderQtyData" fields.

Note: OrderQty = CumQty + LeavesQty (see exceptions above)

Tag	FIX Name	Req	Valid Values	Format	Description
→	*Block OrderQtyData*	Y		OrdQty	Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages"
→→ ***38***	*OrderQty*	*N*		*Qty*	*Quantity of order.*

Underlying Instrument Group Identification

Tag	FIX Name	Req	Valid Values	Format	Description
→	*Block UndInstrmtGrp*				Set of "Underlyings"
711	NoUnderlyings	N		NumInGroup (Int)	
→→	*Block UnderlyingInstrument*				
→→→ ***311***	*UnderlyingSymbol*	*N*		*String*	*Underlying security's Symbol.*

POWERED BY PRIMARY

Position Amount Data Identification

Set of "PositionAmountData" fields.

Tag	FIX Name	Req	Valid Values	Format	Description
→	*Block PositionAmountData*	N		OrdQty	Insert here the set of "OrderQtyData" fields defined in "Common Components of Application Messages"
753	NoPosAmt	N		NumInGroup (Int)	Number of Position Amount entries.
→→ ***707***	*PosAmtType*	*N*	*TVAR = Trade Variation Amount*	*String*	*Type of Position amount*
→→ ***708***	*PosAmt*	*N*		*Qty*	*Position amount*

Position Qty Identification

Set of "PositionQty" fields.

Tag	FIX Name	Req	Valid Values	Format	Description
→	*Block PositionQty*	N		Qty	Insert here the set of " Position Qty" fields defined in "Common Components of Application Messages"
702	NoPositions	N		NumInGroup (Int)	Number of position entries.
→→ ***703***	*PosType*	*N*	*ASF = As –of Trade Qty*	*String*	*Required if NoPositions > 1*
→→ ***704***	*LongQty*	*N*		*Qty*	*Long quantity*
→→ ***705***	*ShortQty*	*N*		*Qty*	*Short quantity*

RootParties Identification

Set of "RootParties" fields.

Tag	FIX Name	Req	Valid Values	Format	Description
→	*Block RootParties*	Y			Insert here the set of " RootParties" fields defined in "Common Components of Application Messages"
1116	NoRootPartyIDs	N		NumInGroup (Int)	Repeating group below should contain unique combinations of RootPartyID, RootPartyIDSource, and RootPartyRole
→→	***1117***	*RootPartyID*	*N*		*String*	*Used to identify source of RootPartyID. Required if RootPartyIDSource is specified. Required if NoRootPartyIDs > 0*
→→	***1118***	*RootPartyIDSource*	*N*		*char*	*Used to identify class source of RootPartyID value (e.g. BIC). Required if RootPartyID is specified. Required if NoRootPartyIDs > 0*
→→	***1119***	*RootPartyRole*	*N*		*int*	*Identifies the type of RootPartyID (e.g. Executing Broker). Required if NoRootPartyIDs > 0*

Alloc Group Identification

Set of "Alloc" fields.

Tag	FIX Name	Req	Valid Values	Format	Description
→	*Block Alloc*	Y			Conditionally required except when AllocTransType = Cancel, or when AllocType = "Ready-to-book" or "Warehouse instruction".
78	NoAllocs	N		NumInGroup (Int)	Number of repeating AllocAccount (79)/AllocPrice (366) entries.
→→	***79***	*AllocAccount*	*C*		*String*	*Required if NoAllocs > 0. Must be first field in repeating group.* *Conditionally required except when for AllocTransType="Cancel", or when AllocType= "Ready-To-Book" or "Warehouse instruction".*
→→	***366***	*AllocPrice*	*C*		*Price*	*AllocAccount plus AllocPrice form a unique Allocs entry.* *Executed price for an AllocAccount (79) entry.*
→→	***80***	*AllocQty*	*C*		*Qty*	*Conditionally required except when for AllocTransType="Cancel", or when AllocType= "Ready-To-Book" or "Warehouse instruction".* *Quantity to be allocated to specific sub-account.*

OrdAlloc Group Identification

Set of "OrderAlloc" fields.

Tag	FIX Name	Req	Valid Values	Format	Description
→	*Block OrderAlloc*	Y			Indicates number of orders to be combined for allocation. If order(s) were manually delivered set to 1 (one).Required when AllocNoOrdersType = 1.
73	NoOrders	N		NumInGroup (Int)	Indicates number of orders to be combined for average pricing and allocation.
→→ ***11***	*ClOrdID*	*C*		*String*	*Order identifier assigned by client if order(s) were electronically delivered over FIX (or otherwise assigned a ClOrdID) and executed. If order(s) were manually delivered (or otherwise not delivered over FIX) this field should contain string "MANUAL". Note where an order has undergone one or more cancel/replaces, this should be the ClOrdID of the most recent version of the order.* *Required when NoOrders(73) > 0 and must be the first repeating field in the group.*
→→ ***37***	*OrderID*	*C*		*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.*

Messages

Header and Trailer

Standard Message Header

Message Header sent by your company to the Exchange

Standard Message Header					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
8	BeginString	Y	FIXT.1.1	String	Identifies beginning of new message and protocol version. ALWAYS FIRST FIELD IN MESSAGE (Always unencrypted).
9	BodyLength	Y		Int	Message length, in bytes, up to the CheckSum field. *Always second field in message.* Always unencrypted. Maximun 500 Kbytes
34	MsgSeqNum	Y		Int	Message sequence number.
35	MsgType	Y	All Msg Types supported	String	Defines message type. Always third field in message.
1128	AppVerID	N	9 = FIX50SP2	String	Indicates application version using a service pack identifier.
43	PossDupFlag	N	Y = Possible Duplicate N = Original Transmission	Boolean	Indicates possible retransmission of message with this sequence number. The value for this tag must be set to "Y" when messages are resent as a result of a resend request.
49	SenderCompID	Y		String	Assigned value used to identify firm sending the message. All messages sent by your firm must have one SenderCompID that is agreed upon in advance with the Exchange.
52	SendingTime	Y		UTC Timestamp	Time message is sent by your company to the exchange. (always expressed in UTC (Universal Time Coordinated, also known as "GMT")
56	TargetCompID	Y	"ROFX"	String (32)	Identifies the router receiving the message. All messages sent by your firm to the exchange must have one TargetCompID.
97	PossResend	N	Y = Possible resend N = Original transmission	Boolean	Indicates that the message may contain information that has been sent under another sequence number.

115	OnBehalfOfCompID	N	String (32)	A unique identifier assigned by the exchange to your firm. This identifier must be present on all order related transactions as a means of identifying the originating source.
122	OrigSendingTime	N	UTC Timestamp	Required for messages resent as a result of a ResendRequest, including Gap Fill messages. If data is not available, set to same value as SendingTime.
128	DeliverToCompID	N	String (32)	Identifies the target executing system.
116	OnBehalfOfSubId	N	String (32)	Value sent by the client that indicates the screen or user from which it originated.
129	DeliverToSubId	N	String (32)	Value sent by the client indicating the specific destination to which the message is sent.

Interconnection agreements between markets

The following tags are reserved for future use in routing scenarios involving more than two markets (eg when a market sends orders to another via a third party):

-OnbehalfOfCompID
-DeliverToCompID
-HopGrp

We repeat the values for SenderCompID and TargetCompID in the tags OnbehalfOfCompID and DeliverToCompID respectively.

Standard Message Trailer

Message Trailer sent by your company to the exchange

Standard Message Trailer					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
10	CheckSum	Y		String (3)	Three byte, simple checksum. *Always last field in message*

POWERED BY PRIMARY

Message summary

Summary of supported Messages

The following table summarizes the session messages supported by the exchange.

Message	Message Type	Page
SESSION MESSAGES		
Logon	A	25
Heartbeat	0	25
Resend Request	2	26
Test Request	1	26
Reject – Session Level	3	26
Sequence Reset	4	28
Logout	5	28
COMMON MESSAGES		
News	B	29
Business Message Reject	j	30
APPLICATION MESSAGES		
TradingSessionStatus	h	32
New Order – Single	D	32
Order Cancel Request	F	34
Order Cancel/Replace Request	G	35
Order Cancel Reject	9	38
Order Status Request	H	39
OrderMass Status Request	AF	40
OrderMass Cancel Request	q	41
Execution Report: New, Response	8	43
Execution Report: Order Canceled Response	8	45

EXECUTION REPORT: ORDER REPLACED RESPONSE	8	45
EXECUTION REPORT: ORDER FILLED/PARTIALLY FILLED RESPONSE	8	49
EXECUTION REPORT: ORDER STATUS RESPONSE	8	51
EXECUTION REPORT: REJECT MESSAGE RESPONSE	8	55
MARKET DATA REQUEST	V	60
MARKET DATA – SNAPSHOT / FULL REFRESH	W	61
MARKET DATA REQUEST REJECT	Y	64
SECURITY LIST REQUEST	x	69
SECURITY LIST	y	71
SECURITY STATUS REQUEST	e	76
SECURITY STATUS	f	77
POST TRADE MESSAGES		
TRADE CAPTURE REPORT REQUEST	AD	89
TRADE CAPTURE REPORT	AE	86
TRADE CAPTURE REPORT ACK	AR	88
ALLOCATION INSTRUCTION	J	93
ALLOCATION INSTRUCTION ACK	P	95
CONFIRMATION	AK	96
CONFIRMATION ACK	AU	99

Session Layer Messages

Message Specification

This section details the session management messages used by the exchange.

Logon (MsgType = A)

The FIX Logon message (A) authenticates a user establishing a connection to a remote system. The Logon (A) message must be the first message sent by the application requesting to initiate a FIX session.

Possible Exchange's response messages: Logon (MsgType = A), Logout (MsgType = 5) or Reject – Session Level (MsgType = 3).

Logon (MsgType = A)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = A		
98	EncryptMethod	Y	0 = None	Int	Method of encryption.
108	HeartBtInt	Y	Integer >= 10	Int	Heartbeat Interval in seconds. HeartBtInt must be equal to or greater than "10".
553	Username	N		String	Username. Provided by the exchange.
554	Password	N		String	Password. Provided by the exchange.
1137	DefaultApplVerID	Y	9 =FIX50SP2	String	The default version of FIX being carried over this FIXT session
	Standard Trailer	Y			

Heartbeat (MsgType = 0)

The Heartbeat (0) monitors the status of the communication link and identifies when the last of a string of messages was not received.

Possible Exchange's response messages: None.

Heartbeat (MsgType = 0)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = 0		
112	TestReqID	C		String	Required if heartbeat message is generated in response to a Test Request message. In this case, this tag must contain the TestReqID that was sent in the Test Request message.
	Standard Trailer	Y			

Test Request (MsgType = 1)

The test request message forces a heartbeat from the opposing application. The test request message checks sequence numbers or verifies communication line status. The opposite application responds to the Test Request with a Heartbeat containing the TestReqID.

Test Request (MsgType = 1)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = 1		
112	TestReqID	Y		String	Identifier included in Test Request message to be returned in resulting Heartbeat
	Standard Trailer	Y			

Resend Request (MsgType = 2)

The resend request is sent by the receiving application to initiate the retransmission of messages. This function is utilized if a sequence number gap is detected, if the receiving application lost a message, or as a function of the initialization process.

Resend Request (MsgType = 2)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = 2		
7	BeginSeqNo	Y	Valid sequence number for session	Int	Message sequence number of first message in range to be resent.
16	EndSeqNo	Y	0 = Infinity	Int	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).
	Standard Trailer	Y			

Reject – Session Level (MsgType = 3)

The FIX Reject message should be issued when a message is received but cannot be properly processed due to a session-level rule violation.

This message will be sent by the Exchange when a session level error has occurred.

Reject (MsgType = 3)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = 3		
45	RefSeqNum	Y		Int	Reference message sequence number (MsgSeqNum) of rejected message.
371	RefTagID	N		Int	The tag number of the FIX field being referenced
372	RefMsgType	N		String	The MsgType of the FIX message being referenced
373	SessionRejectReason	N	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 = CompID problem 10 = SendingTime accuracy problem 11 = Invalid MsgType 12 = XML Validation error 13 = Tag appears more than once 14 = Tag specified out of required order 15 = Repeating group fields out of order 16 = Incorrect NumInGroup count for repeating group 17 = Non “data” value includes field delimiter (SOH character) 99 = Other	Int	Code to identify reason for a session-level Reject message. The server will report the reason for rejection in all messages
58	Text	N		String	Where possible, message to explain reason for rejection.
	Standard Trailer	Y			

Sequence Reset (MsgType = 4)

The Sequence Reset message has two modes: Gap Fill mode and Reset mode. Gap Fill mode is used in response to a FIX Resend Request when one or more messages must be skipped. Reset mode involves specifying an arbitrarily higher new sequence number to be expected by the receiver of the FIX Sequence Reset message, and is used to reestablish a FIX session after an unrecoverable application failure.

Possible Exchange's response messages: None.

Sequence Reset (MsgType = 4)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = 4		
36	NewSeqNo	Y		Int	New sequence number. This number cannot be lower than the expected incoming sequence number of either the client or the Exchange that originally sent the resend request.
123	GapFillFlag	N	Y = Gap Fill message, MsgSeqNum field is valid N = Sequence Reset, ignore MsgSeqNum	Boolean	Indicates that the Sequence Reset message is replacing administrative or application messages, which will not be resent.
	Standard Trailer	Y			

Logout (MsgType = 5)

The FIX Logout message initiates or confirms the termination of a FIX session. Disconnection without the exchange Logout messages should be interpreted as an abnormal condition.

Possible **Exchange's** response messages: Logout (MsgType = 5), Resend Request (MsgType = 2) or Reject – Session Level (MsgType = 3).

Logon (MsgType = 5)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = 5		
58	Text	N		String	Free format text string.
	Standard Trailer	Y			

Common messages

Message Specification

News (MsgType = B)

The news message is a general free format message between the broker and institution. The message is used by the exchange to notify to connected participants (brokers) of market news; contains flags to identify the news item's urgency

News (MsgType = B)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = B		
61	Urgency	N	0 = Normal 1 = Flash 2 = Background	Char	Default value is 0.
42	OrigTime	N		UTCTimestamp	Time of message origination
148	Headline	Y		String	Specifies the headline text
→	***Block LinesOfText***	Y			Set of "Lines of Text"
33	NoLinesOfText			NumInGroup	Specifies the number of repeating lines of text specified
→→ *58*	*Text*	*Y*		*String*	*Repeating field, number of instances defined in LinesOfText (33)*
149	URLLink	N		String	Optional. A URI (Uniform Resource Identifier) or URL (Uniform Resource Locator) link to additional information (i.e. http://www.primary.com/research.html)
1300	MarketSegmentID	Y	E.g.: "DDF", "DDA", "DUAL", "MERV", etc.	String	Market Segment for which this News message applies.
	Standard Trailer	Y			

POWERED BY PRIMARY

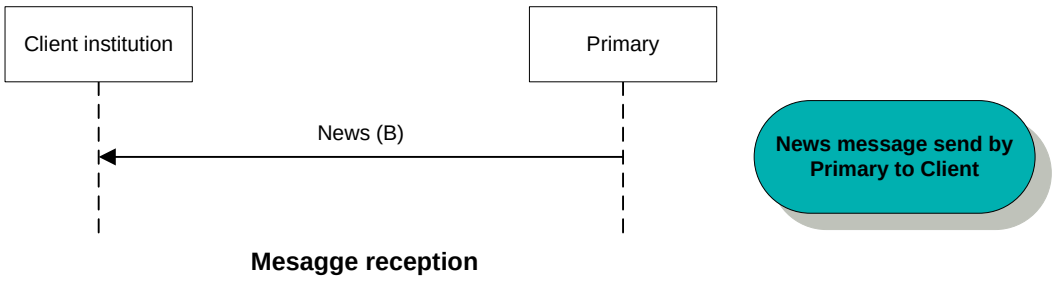
Business Message Reject (MsgType = j)

Message sent by the exchange when it receives a supported message that is syntactically correct in an unsupported situation, and there is no specific rejection message.

Business Message Reject (MsgType = j)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = j		
372	RefMsgType	Y		String	MsgType of the rejected message
380	BusinessRejectReason	Y	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 at this time 18 = Invalid price increment	Int	Reason for rejection
58	Text	N		String	Where possible, explanation of rejection.
	Standard Trailer	Y			

Message Flow – Common messages

News



Application Messages – Market and Trading Session Status

Message Specification

Trading Session Status (MsgType = h)

The Trading Session Status message provides information on the status of a market, and particularly, of the segments for the phase in which they are.

TradingSessionStatus (MsgType = h)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = h		
58	Text	C	"EXTERNAL"	String	Conditionally required when the market segment does not belong to ROFEX (commonly referred to as external segment).
335	TradSesReqID	N		String	Provided for a response to a specific Trading Session Status Request message.
336	TradingSessionID	Y	1= Day	String	Identifier for Trading Session.
625	TradingSessionSubID	Y	0 = Pre-Trading 1 = Trading 2 = Post-Trading 3 = After Hour 4 = Closed	String	Optional market assigned sub identifier for a trading phase within a trading session.
340	TradSesStatus	Y	0 = Unknown 1 = Halted 2 = Open 3 = Closed	Int	State of the trading session. 0= Unknown, if there isn't external fix market connection. 1 = Halted for suspended market or market segment. 2= Open for active market or market segment. 3= Closed for closed market or market segment.
1301	MarketID	Y	"ROFX"	String	Market for which Trading Session applies.
1300	MarketSegmentID	Y	E.g.: "DDF", "DDA", "DUAL", "MERV", etc.	String	Market Segment for which Trading Session applies.
325	UnsolicitedIndicator	N	Y = Message is being sent unsolicited.	Boolean	Set to 'Y' if message is sent unsolicited as a result of a previous subscription request.
	Standard Trailer	Y			

Application Messages - Order Management

This section describes messages exchanged that are relevant to order management, i.e. the sending of orders, cancellations, modifications and reporting of state changes.

Message Specification

New Order – Single (MsgType = D)

The New Order Single message is used by institutions to electronically submit orders to be executed by the exchange. Orders should have a unique identifier (tag ClOrdID <11>) assigned by the institution for a trading day. Orders with duplicate identifiers will be rejected by the exchange.

The acknowledgment of receipt of a New Order Single message is issued in the form of an Execution Report message.

Possible Exchange's response messages: Execution Report (MsgType = 8) or Reject – Session Level (MsgType = 3)

New Order - Single (MsgType = D)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = D		
1	Account	Y		String	Executing account mnemonic.
11	ClOrdID	Y		String	Unique identifier for Order as assigned by the client. Single session.
→	BlockOrderQtyData	Y		Qty	Quantity of order
40	*OrdType*	*Y*	*1 = Market* *2 = Limit* *K = Market with left over as limit(market order with unexecuted quantity becoming limit order at last price)* *3 = Stop Merval* *4 = Stop Limit* *z = Stop*	*Char*	*Order type*
44	*Price*	*C*		*Price*	*Order price. Required for limit, and stop limit orders.*
54	Side	Y	1 = Buy 2 = Sell	Char	Side of order.
18	ExecInst	N	Z = Cancel if not Best x = replace previous orders ROFEX	MultipleValueString	Instructions for order handling, Can contain multiple instructions, space delimited. x = if present, must cancel all previous

			indicator. G = All or None (AON)		orders, if they match in the following fields: account, side, symbol and security exchange. Z = indicates that the order will not be bookable (for products with the option of put or not in book). G = for orders in “all or none” products.
→	Block Instrument	Y			
→	Block Parties	Y			Insert here the set of "Parties". Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole.
59	TimeInForce	N	0 = Day (or session) 1 = Good Till Cancel (GTC) 3 = Immediate or Cancel (IOC) 4 = Fill or Kill (FOK)	Char	Specifies for how long the order remains in effect. For Bid and Offer: 0 = Day is used commonly For Buy and Sell: 3 = Immediate or Cancel is used commonly. Absence of this field is interpreted as DAY.
60	TransactTime	N		UTC Timestamp	Time of order creation.
99	StopPx	C		Price	Price per unit of quantity. Conditionally required when order type is 3,4,or z.
	Standard Trailer	Y			

Note: Tag 18 -> ExecInst = G, only valid in the following cases:

- For instruments “all or none” with TIFs GTC and DAY
- For instruments declared “not all or none” with TIFs IOC and FOK.
- For Bookable orders (not 18=Z flag specified) with TIFs DAY or GTC, in instruments declared “all or none” must specify ExecInst = G

For Bookable orders (not 18=Z flag specified) with TIFs DAY or GTC, in instruments declared “not all or none” with flag ExecInst = G will be rejected.

Order Cancel Request (MsgType = F)

The Order Cancel Request message requests the cancellation of all of the remaining quantity of an existing order. The request will only be accepted if the order can successfully be pulled back from the exchange book without executing. A cancel request is assigned a ClOrdID and is treated as a separate entity. If rejected, the ClOrdID of the Cancel Request will be sent in the Cancel Reject message, as well as the ClOrdID of the actual order in the OrigClOrdID field. The ClOrdID assigned to the cancel request must be unique amongst the ClOrdID assigned to regular orders and replacement orders. A successful Order Cancel Request is replied to with an Execution Report message. Note that the Order Cancel/Replace Request = G should be used to partially cancel (reduce) an order.

Possible Exchange's response messages: Execution Report (MsgType = 8), Reject – Session Level (MsgType = 3) or Order Cancel Reject (MsgType = 9).

Order Cancel Request (MsgType = F)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = F		
11	ClOrdID	Y		String	Unique ID of cancel request as assigned by the institution.
37	OrderId	C		String	Unique identifier for the order to be canceled as assigned by the server. Conditionally required if OrigClOrdId is not present. For cancel orders, simply send this identifier, will not be necessary to include ClOrdID or OrigClOrdID.
41	OrigClOrdID	C		String	The last accepted ClOrdID in an order chain. ClOrdID(11) of the previous non rejected order (generated by user) which will be canceled. Conditionally required if OrderId is not present.
54	Side	Y	1 = Buy 2 = Sell	Char	Side of order
60	TransactTime	Y		UTC Timestamp	Time of order creation.
1	Account	Y		String	
→	Block OrderQtyData	Y			Insert here the set of "OrderQtyData" fields defined in "Common Components Blocks of Application Messages"
→	Block Instrument	Y			In this case the Security Exchange field is mandatory

→	Block Parties	Y		Insert here the set of "Parties". Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole.
	Standard Trailer	Y		

Order Cancel/Replace Request (MsgType = G)

The Order Cancel Replace Request message is used to change the parameters of a previously entered order. It may be used to change attributes of an order (i.e. reduce/increase quantity, change price, etc.). The Cancel/Replace request will only be accepted if the order can successfully be pulled back from the exchange book without executing.

Do not use this message to cancel the remaining quantity of an outstanding order, use the Order Cancel Request message for this purpose.

Only the fields that are being changed need to be sent in the replacement message, (except required fields that must be sent anyway). Fields that are not sent are considered without changes.

If an order is successfully replaced, then it will generate a new OrderID for it, while the replaced order will be canceled.

For the moment may be changed only the following fields:

- OrderQty from OrderQtyData Block;
- Price
- ExecInst

Possible Exchange's response messages: Execution Report (MsgType = 8), Reject – Session Level (MsgType = 3) or Order Cancel Reject (MsgType = 9).

Order Cancel/ReplaceRequest (MsgType = G)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = G		
1	Account	Y		String(45)	Executing account mnemonic.
11	ClOrdID	Y		String	Unique identifier for the order to Cancel/Replace as assigned by the client.

POWERED BY PRIMARY

18	ExecInst	N	Z = Cancel if not Best x= replace previous orders ROFEX indicator.	MultipleValue String	Instructions for order handling, can be used to change the original order handling instructions. Can contain multiple instructions, space delimited. x = if present, must cancel all previous orders, if they match in the following fields: account, side, symbol, and security exchange. Z = indicates that the order will not be bookable (for products with the option of put or not in book). G = for orders in "all or none" products.
37	OrderId	C		String	Unique identifier for the order to Cancel/Request as assigned by the server. Conditionally required if OrigClOrdId is not present.
40	OrdType	Y	1 = Market 2 = Limit K = Market with left over as limit(market order with unexecuted quantity becoming limit order at last price) 3 = Stop Merval 4 = Stop Limit z = Stop	Char	Order type
41	OrigClOrdID	C		String	The last accepted ClOrdID in an order chain. ClOrdID(11) of the previous non rejected order (generated by user) which will be replaced. Conditionally required if OrderId is not present.
44	Price	C		Price	To indicate the new price of the order in case of modification.
54	Side	Y	1 = Buy 2 = Sell	Char	Side of order.
59	TimeInForce	C	0 = Day (or session) 1 = Good Till Cancel (GTC) 3 = Immediate or Cancel (IOC) 4 = Fill or Kill (FOK)	Char	Specifies for how long the order remains in effect. Absence of this field indicates Day order. Conditionally required if TIF is not "Day". For Bid and Offer "0 = Day" is used commonly For Buy and Sell "3 = Immediate or Cancel", is used commonly.

60	TransactTime	Y		UTCTimestamp	Time this order request was initiated/released by the trader or trading system.
→	Block Instrument	Y			In this case the Security Exchange field is mandatory
→	Block Parties	Y			Insert here the set of "Parties". Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole. Can be used among other things for requesting to market segments not belonging to ROFEX, information on the status of orders, filling the parties information with data of market member associated with the user.
→	Block OrderQtyData	C			Insert here the set of "OrderQtyData" fields defined in "Common Components Blocks of Application Messages". To indicate the new amount of the order in case of modification.
	Standard Trailer	Y			

POWERED BY PRIMARY

Order Cancel Reject (MsgType = 9)

The "Order Cancel Reject" message is issued by the exchange, upon receipt of a "Cancel Request", "Mass Cancel Request" or "Order Cancel Replace Request" (modification) message sent by client, which cannot be honored. Filled orders cannot be cancelled or modified.

When rejecting an "Order Cancel Request", the "Order Cancel Reject" message will provide the ClOrdID and OrigClOrdID values which were specified on the original message "Cancel/Mass Cancel/Replace Request" for identification.

Order Cancel Reject (MsgType = 9)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = 9		
11	ClOrdID	Y		String	ClOrdID of the Cancel Request or Cancel/Replace Request that is being rejected.
41	OrigClOrdID	Y		String	The last accepted ClOrdID in an order chain.
37	OrderID	Y		String	If CxlRejReason="Unknown order", specify "NONE".
39	OrdStatus	Y	0 = New 1 = Partially Filled 2 = Filled 4 = Canceled 8 = Rejected	Char	Identifies the current status of the order.
434	CxlRejResponseTo	Y	1 = Order Cancel Request	Char	Identifies the type of request this Cancel Reject is in response to.
102	CxlRejReason	Y	0 = Too late to Cancel 1 = Unknown Order 99 = Other	Int	Code to identify reason for cancel rejection.
58	Text	N		String	Provides the reason why the order was rejected.
	Standard Trailer	Y			

Order Status Request (MsgType = H)

The order status request message is used by the institution to generate an order status message back from the Exchange.

Order Status Request (MsgType = H)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = H		
790	OrdStatusReqID	N		String	Optional, can be used to uniquely identify a specific Order Status Request message. Echoed back on Execution Report if provided.
11	ClOrdID	C		String	The ClOrdID of the order whose status is being requested.. Conditionally required if OrderID is not provided.
37	OrderID	C		String	Conditionally required if ClOrdID(11) is not provided (Either OrderID or ClOrdID must be provided)
→	Block Instrument	Y			Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages".
→	Block Parties	C			Insert here the set of "Parties". Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole.
54	Side	Y	1 = Buy 2 = Sell	Char	Side of order.
	Standard Trailer	Y			

POWERED BY PRIMARY

Order Mass Status Request (MsgType = AF)

Message sent by the client to request status of orders meeting certain selection criteria.

Order Mass Status Request (MsgType = AF)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = AF		
584	MassStatusReqID	Y		String (10)	Unique identifier of this Order Mass Status Request message
585	MassStatusReqType	Y	7 = Status for all orders	Int	Mass Status Request Type.
	Block Parties	N			
1	Account	N		String	Can be used to specify the parties to whom the Order Mass Status Request should apply.
207	SecurityExchange	N		String	Security exchange identifier. Value defined **ROFX**
→	Block Instrument	N			
→→ **1151**	SecurityGroup	N	external (to refer to orders pertaining to contracts outer market segments)	String	An exchange specific name assigned to a group of related securities which may be concurrently affected by market events and actions.
→→ **965**	SecurityStatus		0= All, 1= Actives	String	When 1 is used for requesting orders in the active state, whereas when 0 all the states of orders are requested.
	Standard Trailer	Y			

POWERED BY PRIMARY

Order Mass Cancel Request (MsgType =q)

Message sent by the client to request the cancellation of orders that meet certain selection criteria.

Order Mass Cancel Request (MsgType = q)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = q		
530	MassCancelRequestType	Y	1 = Cancel orders for a security 4 = Cancel orders for a CFICode 7 = Cancel all orders	Char	Selection criteria.
11	ClOrdID	Y		String	Unique ID of Order Mass Cancel Request as assigned by the institution.
60	TransactTime	N		UTCTimestamp	Time this order request was initiated/released by the trader or trading system.
→	Block Instrument	C			Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages". Conditionally required if MassCancelRequestType = 1.
→→ ***461***	*CFICode*	*C*	*"FXXXSX" = Future* *"OPXXXS" = Option Put* *"OCXXXS" = Option Call* *"OXXXPS" = MERVAL Option* *"ESXXXX" = Stock* *"DBXXXX" = Bond* *"XXWXXX" = Swap* *"FXXXXX" = Futures Spread* *"EMXXXX"= CEDEAR* *"DTXXXX"= Financial Trust* *"RPXXXX"= Repurchase* *"MRIXXX"= Index* *"MXXXXX"= Undefined*	*String*	*Indicates the type of security using ISO 10962 standard, Classification of Financial Instruments (CFI code) values. It is recommended that CFICode be used instead of SecurityType for non-Fixed Income instruments.* *Conditionally required if MassCancelRequestType = 4.*

POWERED BY PRIMARY

54	Side	N	1 = Buy 2 = Sell	Char	Optional qualifier used to indicate the side of the market for which orders are to be cancelled. Absence of this field indicates that orders are to be cancelled regardless of side.
→	Block Parties	N			Insert here the set of "Parties". Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole. Used for massive cancelation of all orders belonging to a specified account.
1300	MarketSegmentID	Y		String	Cancel orders for a market segment.
	Standard Trailer	Y			

POWERED BY PRIMARY

Execution Report (MsgType = 8): New

The Execution Report message is used in the following scenarios:

- Confirm the receipt of an order;
- Confirm changes to an existing order (i.e. accept order cancel requests);
- Relay order status information;
- Relay fill information on working orders (trades);
- Reject orders.

Each execution report contains two fields which are used to communicate both the current state of the order as understood by the broker and the purpose of the message: OrdStatus (used to convey the current status of an order) and ExecType (used to identify the purpose of the Execution Report message).

Execution Report (MsgType = 8): New Response					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = 8		
1	Account	Y		String (32)	Executing account mnemonic.
6	AvgPx	Y	0	Price	This tag will always be 0.
11	ClOrdID	Y		String (32)	Unique identifier for New Order, Cancel, or Cancel/Replace that this Execution Report confirms.
14	CumQty	Y	0	Qty	Cumulative executed quantity.
17	ExecID	Y		String (32)	Unique Exchange identifier for message. This identifier is unique trading session.
18	ExecInst	N	Z = Cancel if not Best G = All or None (AON)	MultipleValueString	Instructions for order handling, Can contain multiple instructions, space delimited. Z = for products with the option of put or not in book, and non bookeable orders. G = for orders in “all or none” products. x = if present, must cancel all previous orders, if they match in the following fields: account, side, symbol and security exchange.
31	LastPx	Y	0	Price	Price of this (last) fill. This tag will always be 0.
32	LastQty	Y	0	Qty	Quantity (e.g. shares) bought/sold on this (last) fill. This tag will always be 0.
37	OrderID	Y		String (32)	Unique identifier for order as assigned by Exchange. This identifier is unique per trading session.

→	Block Parties	N			Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole.
→	Block OrderQtyData	Y		Qty	OrderQty submitted by the client.
39	OrdStatus	Y	0 = New	Char	Identifies the current status of an order. The value will be 0 for New if the original FIX message was New Order – Single.
40	OrdType	Y	2 = Limit K = Market with Left Over as Limit	Char	Type of order specified by individual entering the order.
41	OrigClOrdID	C		String (32)	Conditionally required for response to a Cancel or Cancel/Replace request (ExecType=PendingCancel, Replace, or Canceled) when referring to orders that where electronically submitted over FIX or otherwise assigned a ClOrdID(11). ClOrdID of the previous accepted order (NOT the initial order of the day) when canceling or replacing an order.
44	Price	C		Price	Order Price submitted by the client.
54	Side	Y	1 = Buy 2 = Sell	Char (1)	Side submitted by the client
→	Block Instrument	Y			In this case the Security Exchange field is mandatory
59	TimeInForce	N	0 = Day (or session) 1 = Good Till Cancel (GTC) 3 = Immediate or Cancel (IOC) 4 = Fill or Kill (FOK)	Char	Specifies how long the order remains in effect. If not present, DAY order is the default.
60	TransactTime	Y		UTC Timestamp	Time at which the order is accepted by the exchange.
150	ExecType	Y	0 = New	Char	The value will be 0 for New if the original FIX message was New Order – Single.
151	LeavesQty	Y		Qty (9)	Amount of stocks units open for further execution.
58	Text	N		String	
	Standard Trailer	Y			

Execution Report (MsgType = 8): Order Canceled Response

Execution Report (MsgType = 8): Order Canceled Response					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = 8		
1	Account	Y		String (32)	Executing account mnemonic.
6	AvgPx	Y		Price	AvgPx submitted with Cancel order. Calculated average price of all fills on this order.
11	ClOrdID	Y		String (32)	Unique identifier for Cancel order that this Execution Report confirms
14	CumQty	Y		Qty	CumQty submitted with Cancel order.
17	ExecID	Y		String (32)	Unique Exchange's identifier for message. This identifier is unique per trading session.
18	ExecInst	N	Z = Cancel if not Best G = All or None (AON)	MultipleValueString	Instructions for order handling, Can contain multiple instructions, space delimited. Z = for products with the option of put or not in book, and non bookeable orders. G = for orders in "all or none" products. Returned when OrderStatus is not Rejected.
31	LastPx	C		Price	Price of this fill. Required if ExecType = Trade.
32	LastQty	C		Qty	Quantity of stocks units bought/sold on this fill. Required if ExecType = Trade.
37	OrderID	Y		String (32)	Unique identifier for order as assigned by the exchange. This identifier is unique per trading session.
→	Block Parties	N			Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole.
→	Block OrderQtyData	Y		Qty	OrderQty submitted with Cancel order.
39	OrdStatus	Y	4 = Canceled	Char	Identifies the current status of an order.
40	OrdType	Y	2 = Limit K = Market with Left Over as Limit	Char	Type of order specified by individual entering the order.

41	OrigClOrdID	N		String (32)	The last accepted ClOrdID in an order chain.
44	Price	N		Price	Price submitted with Cancel order.
54	Side	Y	1 = Buy 2 = Sell	Char	Side submitted with Cancel order.
→	Block Instrument	Y			In this case the Security Exchange field is mandatory
59	TimeInForce	N	0 = Day (or session) 1 = Good Till Cancel (GTC) 3 = Immediate or Cancel (IOC) 4 = Fill or Kill (FOK)	Char	Specifies how long the order remains in effect. If not present, DAY order is the default.
60	TransactTime	Y		UTC Timestamp	Time at which the order is cancelled by the exchange.
150	ExecType	Y	4 = Canceled	Char	Describes the nature of the execution report while OrdStatus identifies the current order status.
151	LeavesQty	Y		Qty (9)	Amount of stocks units open for further execution.
58	Text	N		String	It always returns Canceled
	Standard Trailer	Y			

Execution Report (MsgType = 8): Order Replaced Response

Execution Report (MsgType = 8): Order Replaced Response					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = 8		
1	Account	Y		String (32)	Executing account mnemonic.
6	AvgPx	Y		Price	AvgPx submitted with Cancel/Replace order.
11	ClOrdID	Y		String (32)	Unique identifier for Cancel/Replace order that this Execution Report confirms.
14	CumQty	Y		Qty	CumQty submitted with Cancel/Replace order.
17	ExecID	Y		String (32)	Unique exchange identifier for message. This identifier is unique per trading session.
18	ExecInst	N	Z = Cancel if not Best	MultipleValueString	Instructions for order handling, Can contain multiple instructions, space delimited. Z = for products with the option of put or not in book, and non bookeable orders. Returned when OrderStatus is not Rejected.
31	LastPx	C		Price	Price of this fill. Required if ExecType = Trade.
32	LastQty	C		Qty	Quantity of stocks units bought/sold on this fill. Required if ExecType = Trade.
37	OrderID	Y		String (32)	Unique identifier for order as assigned by the exchange. This identifier is unique per trading session and identifies the replaced order.
→	Block Parties	N			Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole.
→	Block OrderQtyData	Y		Qty	OrderQty submitted with Cancel/Replace order.
39	OrdStatus	Y	0= New 1 = Partially Filled 2= Filled	Char	Identifies the current status of order.

40	OrdType	Y	2 = Limit K = Market with Left Over as Limit	Char	Type of order specified by individual entering the order.
41	OrigClOrdID	N		String (32)	The last accepted ClOrdID in an order chain.
44	Price	N		Price	Price submitted with Cancel/Replace order.
54	Side	Y	1 = Buy 2 = Sell	Char	Side submitted with Cancel/Replace order.
→	Block Instrument	Y			In this case the Security Exchange field is mandatory
60	TransactTime	Y		UTC Timestamp	Time at which the order is cancelled by the exchange.
150	ExecType	Y	5 = Replaced	Char	Describes the nature of the execution report while OrdStatus identifies the current order status.
151	LeavesQty	Y		Qty	Amount of stocks units open for further execution.
58	Text	N		String	
	Standard Trailer	Y			

POWERED BY PRIMARY

Execution Report (MsgType = 8): Order Filled/Partially Filled Response

This message will be sent to the customer as a result of an order matching leading to trade creation.

Execution Report (MsgType = 8): Order Fill/Partially Filled Response					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = 8		
1	Account	Y		String (32)	Executing account mnemonic.
6	AvgPx	Y		Price	Calculated average price of all fills on this order.
11	ClOrdID	Y		String (32)	Unique identifier for the order that this Execution Report references.
14	CumQty	Y		Qty	Total number of shares filled.
17	ExecID	Y		String (32)	Unique Exchange identifier for message. This identifier is unique per trading session.
18	ExecInst	N	Z = Cancel if not Best G = All or None (AON)	MultipleValueString	Instructions for order handling, Can contain multiple instructions, space delimited. Z = for products with the option of put or not in book, and non bookeable orders. G = for orders in "all or none" products.
31	LastPx	Y		Price	Price of this fill.
32	Last Qty	Y		Qty	Quantity of stocks units bought/sold on this fill.
37	OrderID	Y		String	Unique identifier for Order as assigned by the Exchange. This identifier is unique per trading session.
→	Block Parties	N			Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole.
→	Block OrderQtyData	Y		Qty	OrderQty submitted by the client.
39	OrdStatus	Y	1 = Partially Filled 2 = Filled	Char	Identifies the current status of an order.
40	OrdType	Y	1 = Market 2 = Limit K = Market with Left Over as Limit	Char	Type of order specified by individual entering the order.

POWERED BY PRIMARY

41	OrigClOrdID	N		String (32)	The last accepted ClOrdID in an order chain.
44	Price	N		Price	Price per share.
54	Side	Y	1 = Buy 2 = Sell	Char	Side submitted by the client.
→	Block Instrument	Y			In this case the Security Exchange field is mandatory
59	TimeInForce	N	0 = Day (or session) 1 = Good Till Cancel (GTC) 3 = Immediate or Cancel (IOC) 4 = Fill or Kill (FOK)	Char	Specifies how long the order remains in effect. If not present, DAY order is the default.
60	TransactTime	Y		UTC Timestamp	Time at which the order was filled.
150	ExecType	Y	F = Trade (Partial Fill or Fill)	Char	Describes the nature of the execution report while OrdStatus identifies the status of the order.
151	LeavesQty	Y		Qty	Amount of instrument units open for further execution.
58	Text	N		String	
	Standard Trailer	Y			

POWERED BY PRIMARY

Execution Report (MsgType = 8): Order Status Response – No orders

This message will be sent to the customer as the reply of an order mass status request or an order status request, in the case that if there are no associated orders.

Execution Report (MsgType = 8): Order Status Response					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = 8		
790	OrdStatusReqID	C		String	Required if responding to and if provided on the Order Status Request message. Echo back the value provided by the requester.
911	TotNumReports	C	0	Int	Can be used when responding to an Order Mass Status Request to identify the total number of Execution Reports which will be returned. It is related with the amount of active orders.
584	MassStatusReqID	C		Int	When responding to an Order Mass Status Request, corresponds to the unique identifier of Order Mass Status Request message
912	LastRptRequested	Y	Y = Last message	Boolean	Indicates that this is the last Execution Reports which will be returned as a result of the request.
6	AvgPx	Y	0	Price	Calculated average price of all fills on this order.
14	CumQty	Y	0	Qty	Total number of shares filled.
17	ExecID	Y	0	String (32)	Unique identifier for message. This identifier is unique per trading session.
37	OrderID	Y	0	String	Unique identifier for order as assigned by the exchange. This identifier is unique per trading session.
39	OrdStatus	Y	4 = Cancelled	Char	Identifies the current status of an order.
41	OrigClOrdID	N		String (32)	The last accepted ClOrdID in an order chain.
54	Side	Y	1 = Buy	Char	Side submitted by the client.
→	Block Instrument	Y			Symbol(55)=”N/A” and no Security Exchange
→	Block Parties	N			Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole.
60	TransactTime	Y		UTC	Time at which the order was filled.

				Timestamp	
150	ExecType	Y	I = Order Status	Char (1)	Describes the nature of the order status report while OrdStatus identifies the status of the order.
151	LeavesQty	Y	0	Qty	Amount of instrument units open for further execution.
58	Text	N		String	
	Standard Trailer	Y			

POWERED BY PRIMARY

Execution Report (MsgType = 8): Order Status Response – With orders

This message will be sent to the customer as the reply of an order mass status request or an order status request, in the case that there are at least one order associated that satisfies the request,

Execution Report (MsgType = 8): Order Status Response					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = 8		
790	OrdStatusReqID	C		String	Required if responding to and if provided on the Order Status Request message. Echo back the value provided by the requester.
911	TotNumReports	Y		Int	Can be used when responding to an Order Mass Status Request to identify the total number of Execution Reports which will be returned.
584	MassStatusReqID	C		Int	When responding to an Order Mass Status Request, corresponds to the unique identifier of Order Mass Status Request message
912	LastRptRequested	Y	N = Not last message Y= Last message	Boolean	Can be used when responding to an Order Mass Status Request to indicate that this is the last Execution Reports which will be returned as a result of the request.
1	Account	Y		String (32)	Executing account mnemonic.
6	AvgPx	Y		Price	Calculated average price of all fills on this order. It will be 0 in case of OrdStatus = 8 (Rejected)
11	ClOrdID	Y		String (32)	Unique identifier for the order that this Execution Report references.
17	ExecID	Y	0	String (32)	Unique identifier for message. This identifier is unique per trading session.
18	ExecInst	N	Z = Cancel if not Best G = All or None (AON)	MultipleValueString	Instructions for order handling, Can contain multiple instructions, space delimited. Z = for products with the option of put or not in book, and non bookeable orders. G = for orders in “all or none” products. Returned when OrderStatus is not Rejected.
→	Block Parties	N			Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole.

POWERED BY PRIMARY

31	LastPx	C		Price	Price of this (last) fill. Required if ExecType = Trade.
32	LastQty	C		Qty	Quantity of stocks units bought/sold on this (last) fill. Required if ExecType = Trade.
37	OrderID	Y		String	Unique identifier for order as assigned by the exchange. This identifier is unique per trading session.
→	Block OrderQtyData	Y		Qty	OrderQty submitted by the client. It will be 0 in case of OrdStatus = 8 (Rejected)
39	OrdStatus	Y	0 = New 1 = Partially Filled 2 = Filled 4 = Canceled 8 = Rejected	Char	Identifies the current status of an order.
40	OrdType	Y	1 = Market 2 = Limit K = Market with Left Over as Limit	Char	Type of order specified by individual entering the order.
41	OrigClOrdID	N		String	The last accepted ClOrdID in an order chain.
44	Price	Y		Price	Price per share. It will be 0 in case of OrdStatus = 8 (Rejected)
54	Side	Y	1 = Buy 2 = Sell	Char	Side submitted by the client.
→	Block Instrument	Y			In this case the Security Exchange field is mandatory
60	TransactTime	Y		UTC Timestamp	Time at which the order was filled.
103	OrdRejReason	N		Int	It will be 5= Unknown Order in case of OrdStatus = 8 (Rejected)
150	ExecType	Y	I = Order Status	Char (1)	Describes the nature of the order status report while OrdStatus identifies the status of the order.
151	LeavesQty	Y		Qty	Amount of instrument units open for further execution. It will be 0 in case of OrdStatus = 8 (Rejected)
14	CumQty	Y		Qty	Total number of shares filled. It will be 0 in case of OrdStatus = 8 (Rejected)
58	Text	N		String	Order Updated
	Standard Trailer	Y			

Execution Report (MsgType = 8): Reject Message Response

(The original FIX message sent by the customer was New Order – Single request.)

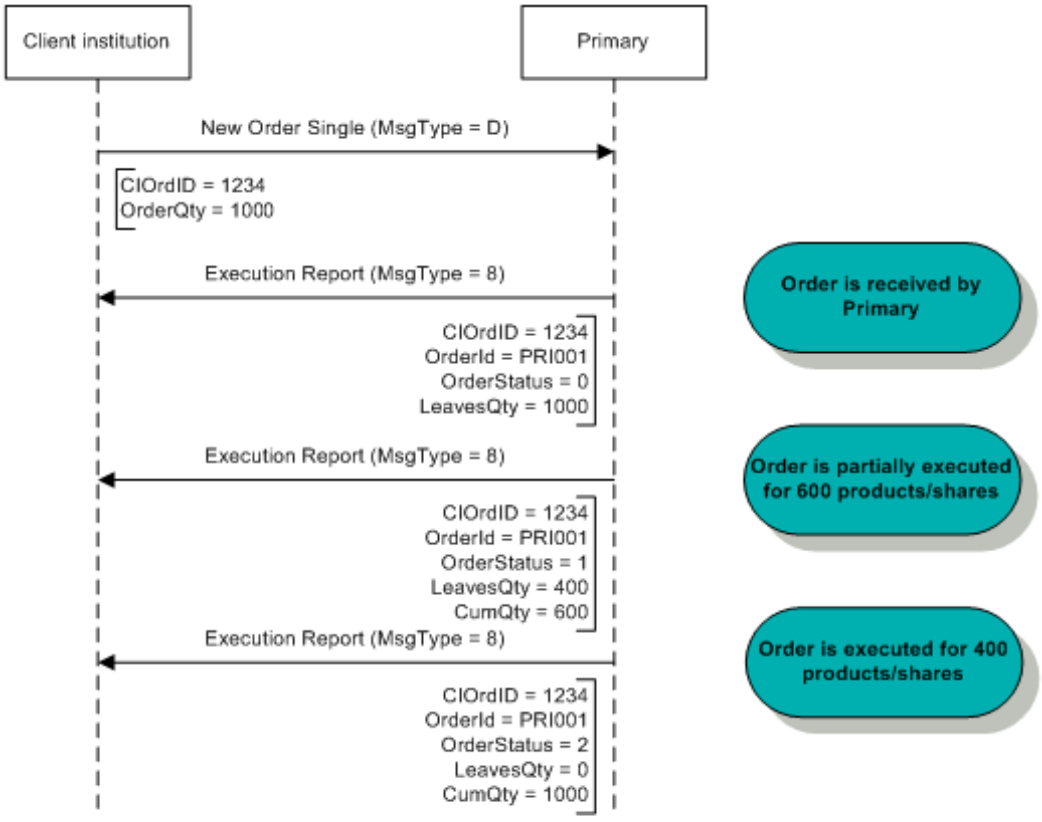
Execution Report (MsgType = 8): Reject Message Response					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = 8		
1	Account	Y		String (32)	Executing account mnemonic.
6	AvgPx	Y	0	Price	This tag will always be 0.
11	ClOrdID	Y		String (32)	Unique identifier for the order that the Execution Report references.
14	CumQty	Y	0	Qty	This tag will always be 0
17	ExecID	Y		String (32)	Unique Exchange identifier for message. This identifier is unique per trading session.
37	OrderID	Y	"NONE"	String (32)	Unique identifier for order as assigned by the Exchange. This identifier is unique per trading session. "NONE" in case of rejected order.
→	Block Parties	N			Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole.
→	Block OrderQtyData	Y		Qty	OrderQty submitted by the client.
39	OrdStatus	Y	8 = Rejected	Char	Identifies the current status of an order.
44	Price	N		Price	Price per share.
54	Side	Y	1 = Buy 2 = Sell	Char	Side submitted by the client.
→	Block Instrument	Y			Security Exchange field is always present
59	TimeInForce	N	0 = Day (or session) 1 = Good Till Cancel (GTC) 3 = Immediate or Cancel (IOC) 4 = Fill or Kill (FOK)	Char	Specifies how long the order remains in effect. If not present, DAY order is the default.
60	TransactTime	Y		UTC	Time at which the order was rejected.

				Timestamp	
150	ExecType	Y	8 = Rejected	Char (1)	The value will always be 8 for Rejected because the original FIX message was New Order – Single.
151	LeavesQty	Y	0	Qty	Amount of instrument units open for further execution. It will always be 0.
58	Text	N		String	
	Standard Trailer	Y			

Message Flow - Order Management

New Order – Single

In this example, an order is sent by the client institution. This order is partially filled and is completely filled afterwards.

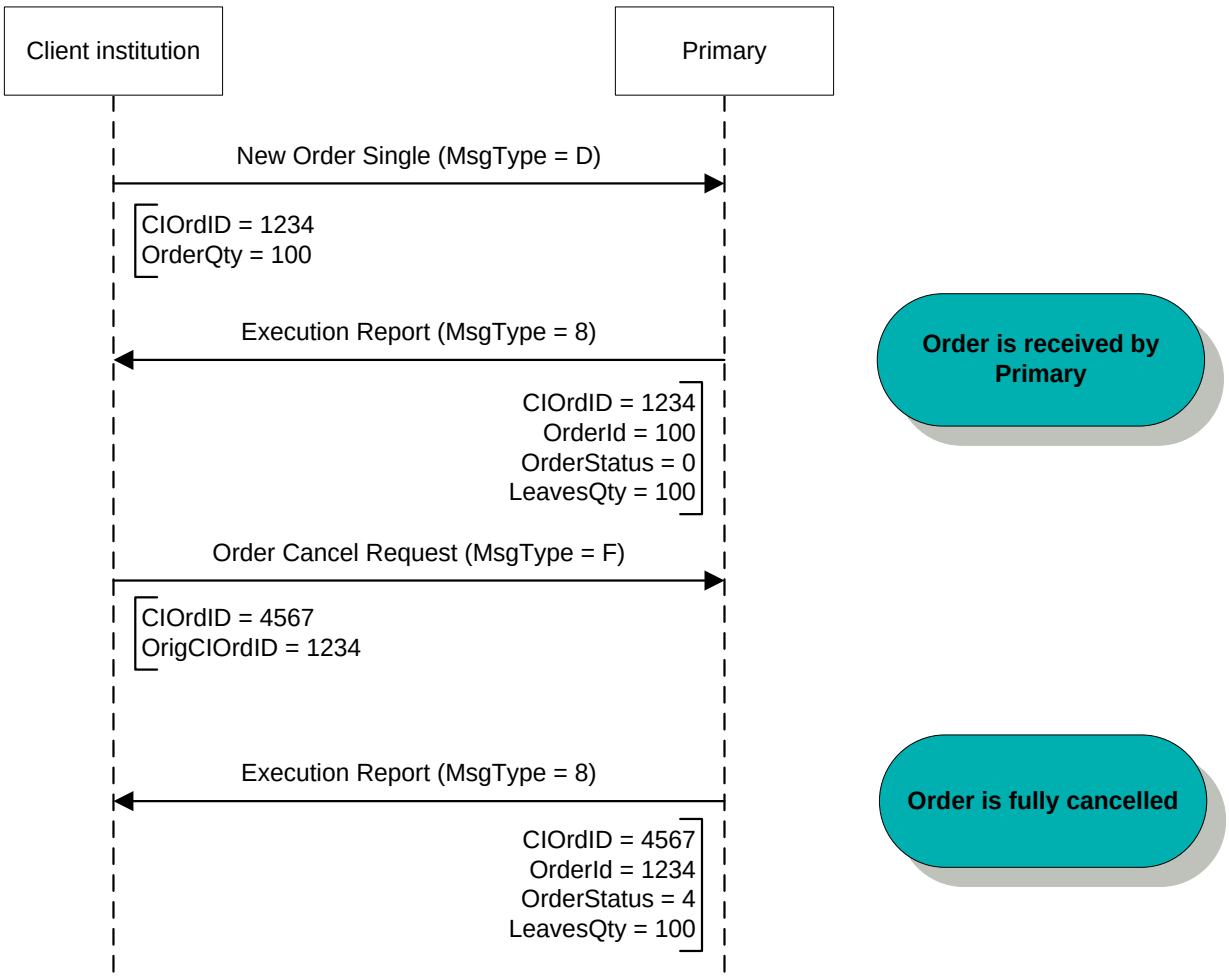


New Order, Partial Fill and Complete Fill

POWERED BY PRIMARY

Order Cancel Request

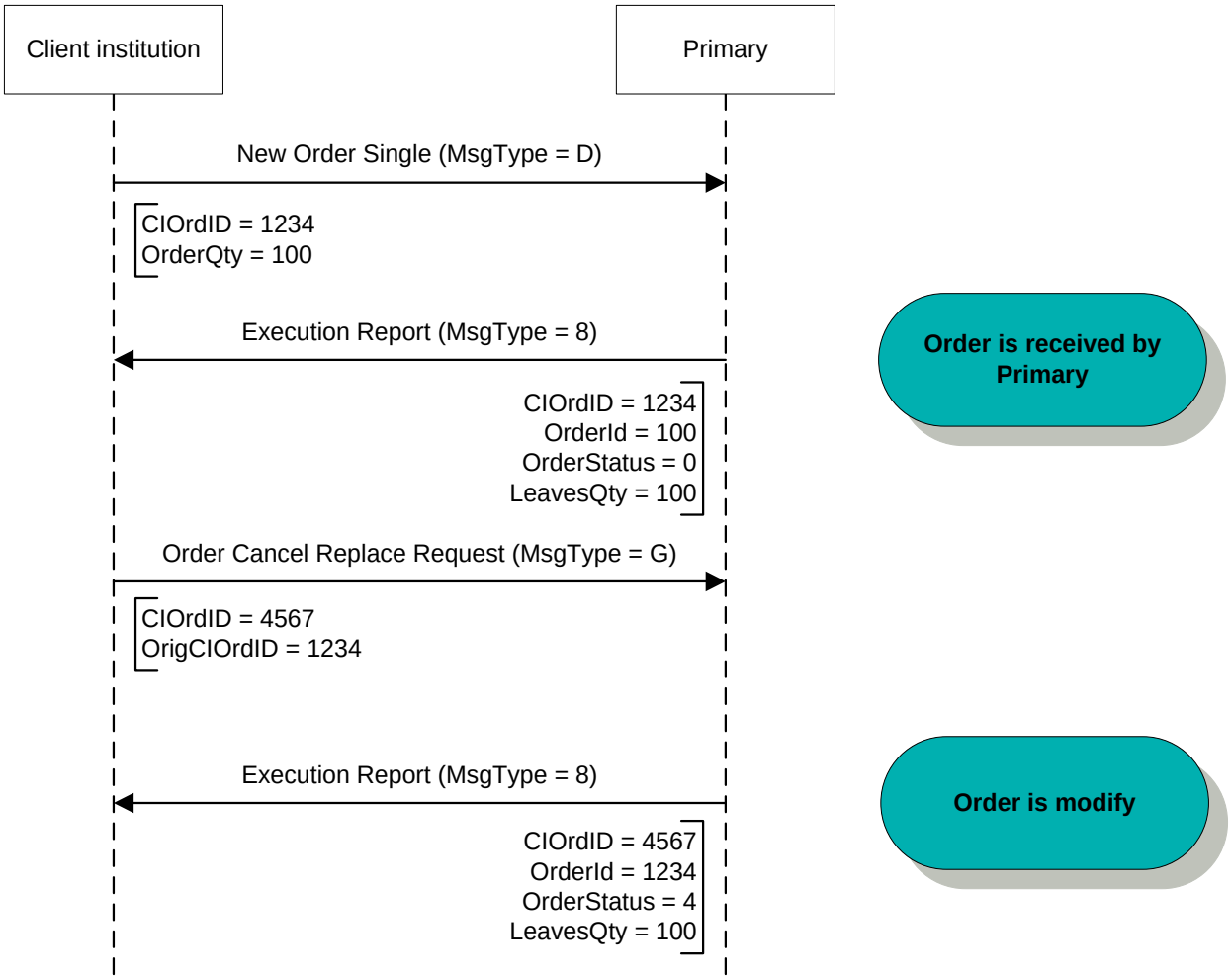
Once an order is accepted by the exchange, it is assigned a unique internal identifier by instrument, sent to the client in the tag OrderID in each Execution Report message. The client may take action on that order using the OrderID instead of the ClOrdID.



Order Cancellation

Order Cancel/Replace Request

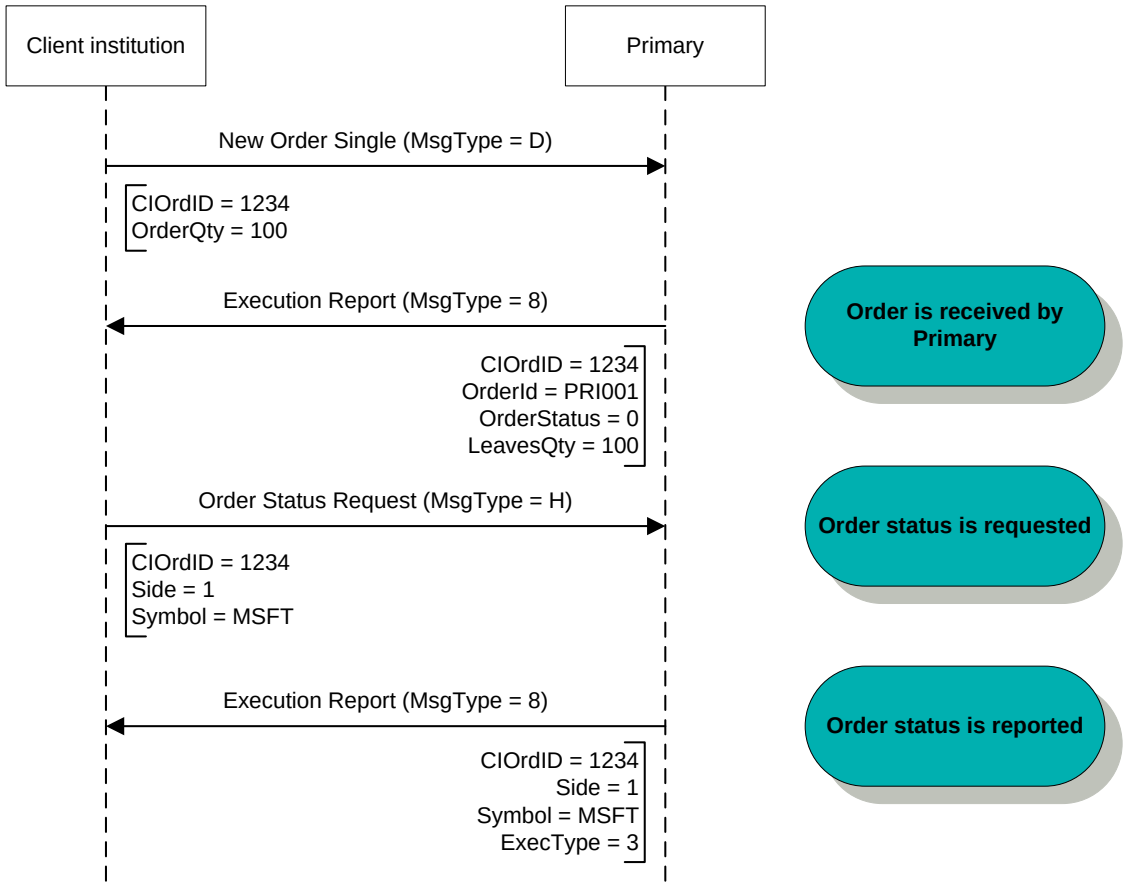
Once an order is accepted by the exchange, it is assigned a unique internal identifier by instrument, sent to the client in the tag OrderID in each Execution Report message. The client may take action on that order using the OrderID instead of the ClOrdID.



Order Modification

POWERED BY PRIMARY

Order Status Request



Order Status Request

Application Messages – Market Data

Message specification

Market Data Request (MsgType = V)

A Market Data Request is a general request for market data on a specific security. A successful Market Data Request returns one Market Data Full Snapshot message containing one or more Market Data Entries.

Possible Exchange response messages: Market Data – Snapshot / Full Refresh (MsgType = W) and Market Data Request Reject (MsgType = Y)

Market Data Request (MsgType = V)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = V		
262	MDReqID	Y		String (32)	Must be unique, or the ID of previous Market Data Request to disable if SubscriptionRequestType = Disable previous Snapshot + Updates Request.
263	SubscriptionRequestType	Y	0 = Snapshot 1 = Snapshot + Updates 2 = Disable Previous Snapshot + Update	Char	SubscriptionRequestType indicates to the other party what type of response is expected.
264	MarketDepth	Y	0 = Full Book 1 = Top of Book N >1 = Report best N price tiers of data	Int	Depth of market for Book Snapshot. Maximun depth 5.
265	MDUpdateType	C	0 = Full Refresh	Int	Conditional field when SubscriptionRequestType = 1 Specifies the type of Market Data update.
266	AggregatedBook	Y	Y = one book entry per side per price N = Multiple entries per side per price	Boolean	Specifies whether or not book entries should be aggregated.
→	***Block MDReqGrp***	Y			Number of MDEntryType fields requested

POWERED BY PRIMARY

267		NoMDEntryTypes	Y		NumInGroup (Int)	Number of MDEntryType fields requested.
→→	***269***	*MDEntryType*	*Y*	*0 = Bid* *1 = Offer* *2 = Trade* *4 = Opening Price* *5 = Closing Price* *6 = Settlement Price* *7 = Trading Session High Price* *8 = Trading Session Low Price* *B = TradeVolume* *C = Open Interest*	*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 requesting the Market Data is interested in receiving.*
→		***Block InstrumentMDReqGrp***	Y			
146		NoRelatedSym	Y		NumInGroup (Int)	Number of symbols requested
→→		Block Instrument	Y			
		Standard Trailer	Y			

Market Data – Snapshot / Full Refresh (MsgType = W)

The Market Data Snapshot/Full Refresh messages are sent as the response to a Market Data Request message. The message refers to only one Market Data Request. It will contain the appropriate MDReqID tag value to correlate the request with the response.

Market Data – Snapshot / Full Refresh (MsgType = W)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = W		
262	MDReqID	Y		String (32)	Unique identifier for Market Data Request
264	MarketDepth	N		Int	Can be used to define the current depth of the book.
→	Block Instrument	Y			
→	***Block MDFullGrp***	Y			Number of entries following.
268	NoMDEntries	Y		NumInGroup (Int)	Number of entries following.
→→	***269***	*MDEntryType*	*Y*	*0 = Bid* *1 = Offer* *2 = Trade* *4 = Opening Price* *5 = Closing Price* *6 = Settlement Price* *7 = Trading Session High Price* *8 = Trading Session Low Price* *B = TradeVolume* *C = Open Interest* *x = Nominal Volume* *w = Cash Volume*	*Int*	*Must be the first field in this repeating group. Identifies the type of this entry*
→→	***270***	*MDEntryPx*	*C*		*Price*	*Price of the Market Data Entry. Conditional field when MDEntryType is* *0 = Bid* *1 = Offer* *2 = Trade* *4 = Opening Price* *5 = Closing Price*

POWERED BY PRIMARY

						6 = Settlement Price *7 = Trading Session High Price* *8 = Trading Session Low Price* *w= Cash Volume*
→→	***271***	*MDEntrySize*	*C*		*Qty*	*Conditionally required if MDEntryType is* *0 = Bid* *1 = Offer* *2 = Trade* *B = TradeVolume y* *C= Open Interest* *x= Nominal Volume*
→→	***272***	*MDEntryDate*	*N*		*UTCDateOnly*	*Date of Market Data Entry. Date represented in UTC (Universal Time Coordinated, also known as "GMT") in YYYYMMDD format. This special-purpose field is paired with UTCTimeOnly to form a proper UTCTimestamp for bandwidth-sensitive messages. Valid values: YYYY = 0000-9999, MM = 01-12, DD = 01-31.*
→→	***273***	*MDEntryTime*	*N*		*UTCTimeOnly*	*Time of Market Data Entry. Field of type "Time-only" represented in UTC (Universal Time Coordinated, also known as "GMT") expressed in HH:MM:SS.sss (milliseconds) format, colons, and period required.* *Valid values are: HH = 00-23, MM = 00-59, SS = 00-5960 (60 only if UTC leap second), sss=000-999 (indicating milliseconds).*
→→	***277***	*Trade Condition*	*N*	*U = Exchange Last*	*String*	*Sent with the fields: TradeSide(7201), MDEntryPx(270), MDEntrySize(271), MDEntryDate(272), MDEntryTime(273), when MDEntryType(269) = Trade(2), to inform the "Last Trade" has occurred or not, at the moment when the message is sent.*
→→	**290**	MDEntryPositionNo	N		Int	Display position of a bid or offer, numbered from most competitive to least competitive, per market side, beginning with 1.
→→	***828***	*TrdType*	*N*	*0=Regular Trade* *1=Block Trade* *1001=Allocation* *1002=Give Up*	*Int*	*Specifies trade type when a trade is being reported. Unlike the "Exchange Last" or "Last Trade", here trades are reported when they occur, while the "Exchange Last" will be reported even if trade has not happened recently.*

				1003=Floor Trade		*Sent with the fields: TrdType (828), MDEntryPx(270), MDEntrySize(271), MDEntryDate(272), MDEntryTime(273), and MDEntryPositionNo (290) for MDEntryType(269) = Trade(2).*
→→	*<7201>*	*Trade Side*	*N*	*1 = Buy* *2 = Sell*	*Char*	*Side of the trade.*
		Standard Trailer	Y			

Market Data Request Reject (MsgType = Y)

The Market Data Request Reject will be issued by the Exchange when it cannot honor the Market Data Request, due to business or technical reasons.

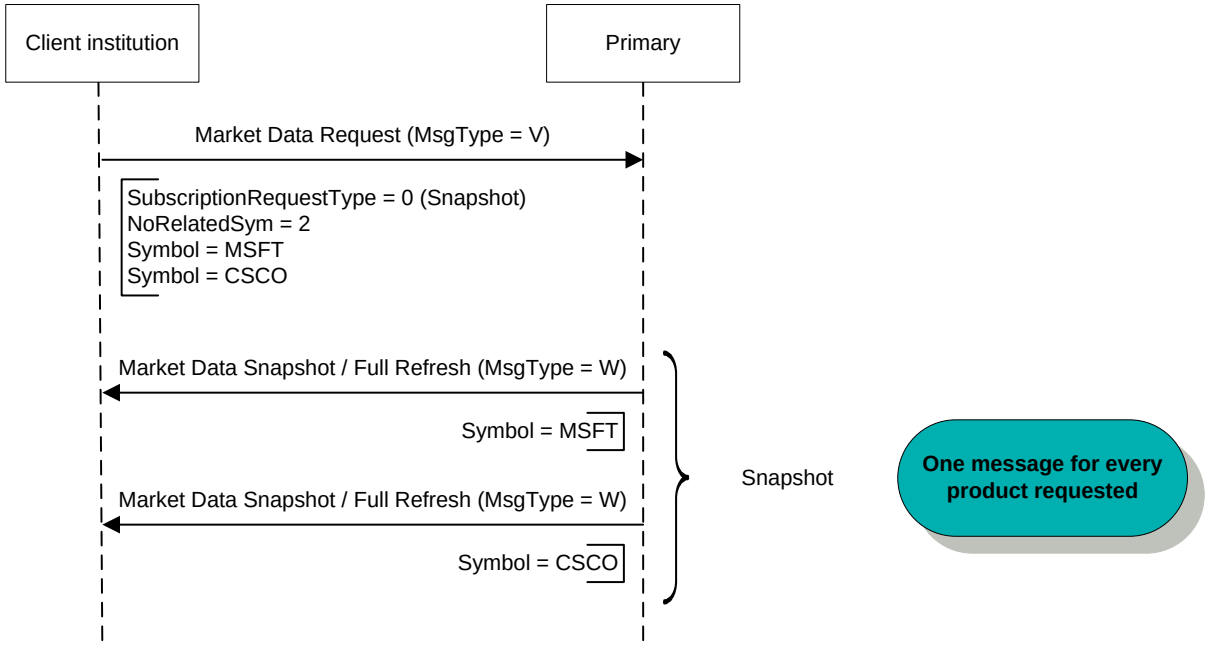
Market Data Request Reject (MsgType = Y)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = Y		

POWERED BY PRIMARY

262	MDReqID	Y		String (32)	Must refer to the MDReqID of the request.
281	MDReqRejReason	N	0 = Unknown symbol 1 = Duplicate MDReqID 2 = Insufficient Bandwidth 3 = Insufficient Permissions 4 = Unsupported Subscription Request Type 5 = Unsupported MarketDepth 6 = Unsupported MDUpdateType 7 = Unsupported AggregatedBook 8 = Unsupported MDEntryType	Char	Reason for the rejection of a Market Data request.
58	Text	N		String	
	Standard Trailer	Y			

Message Flow - Market Data

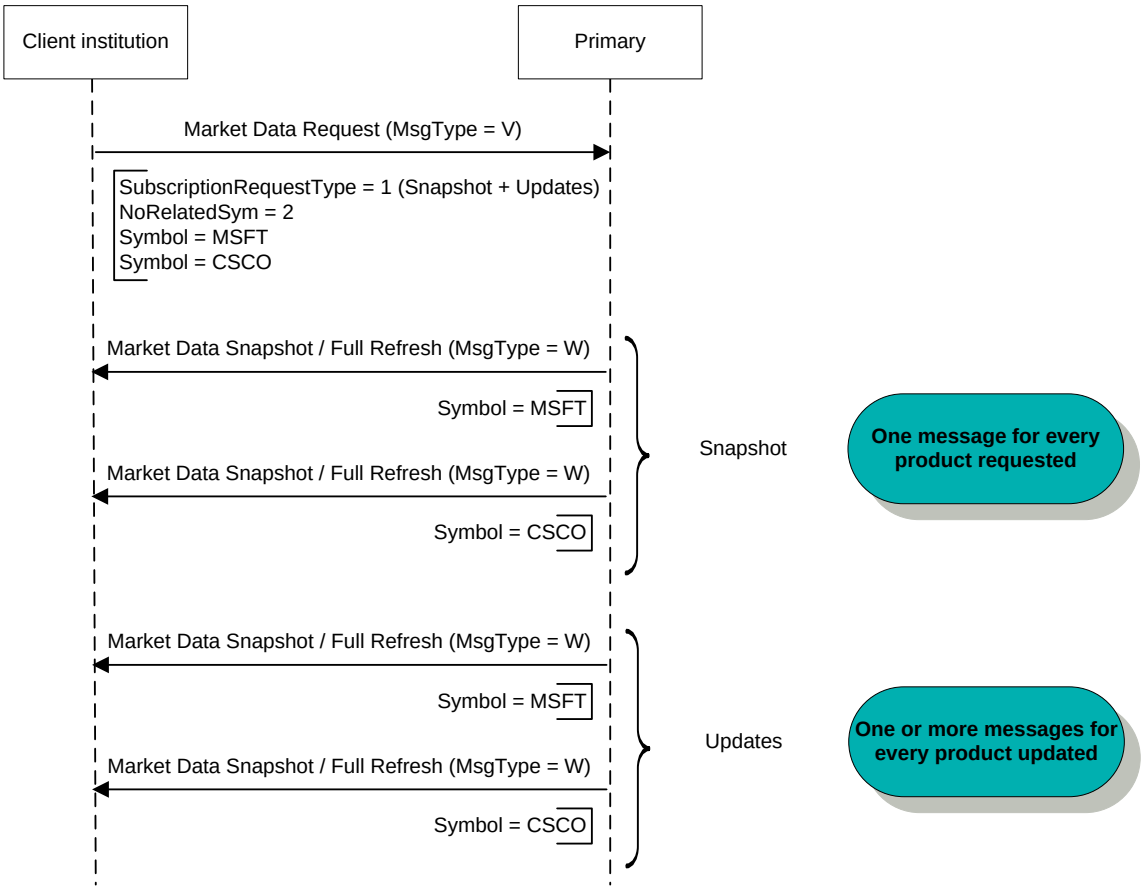
Market Data Request (Full Refresh) without updates



Market Data Request (Full Refresh) without updates

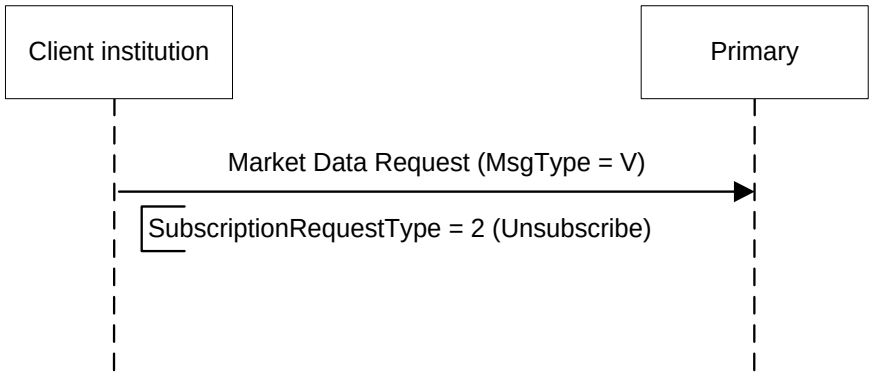
POWERED BY PRIMARY

Market Data Request (Full Refresh) with updates



Market Data Request (Full Refresh) with updates

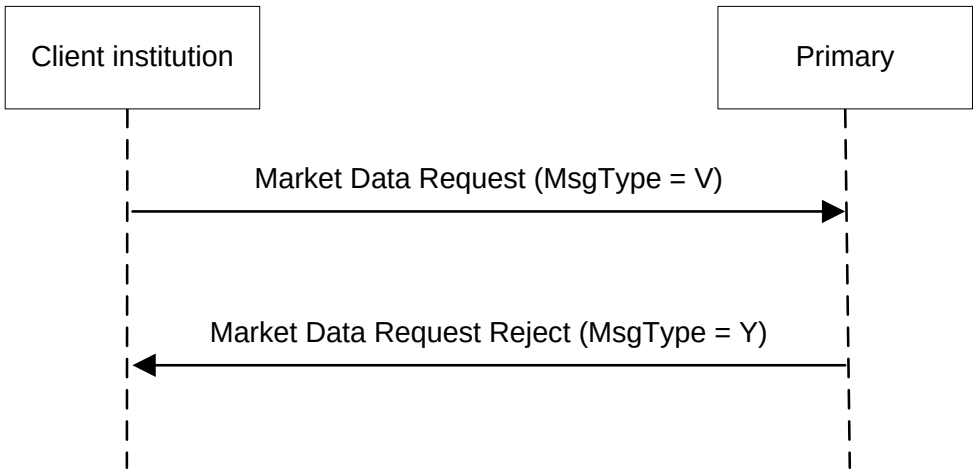
Market Data Unsubscribe



Market Data unsubscribe

POWERED BY PRIMARY

Market Data Request Incorrect



Market Data Request Incorrect

Application Messages – Security Definition

FIX messages are utilized so that the connecting parties are able to determine which instruments are negotiated at the Exchange. Instrument definition messaging is based on a subscription model, in which the client institutions subscribe to receive instrument definitions according to specific criteria, and optionally receive updates afterwards. The subscription may be cancelled at any time.

Message specification

Security List Request (MsgType = x)

Used by the client to request the instrument definitions.

Possible exchange's response messages: Security List (MsgType = y)

Security List Request (MsgType = x)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = x		
320	SecurityReqID	Y		String	Unique identifier for each Security List Request message
559	SecurityListRequestType	Y	0 = Symbol 1 = SecurityType or CFICode 2 = Product 4 = All Securities	Int	Selection criteria used
1301	MarketID	N	"ROFX"	String	Identifies the market which lists and trades the instrument
1300	MarketSegmentID	N	E.g.: "DDF", "DDA"	String	Identifies the market segment
→	Block Instrument	C			Conditional field where SecurityListRequestType is 0 = Symbol, in this case the security exchange-field is also expected.
→→ ***461***	*CFICode*	*C*	*"FXXXSX" = Future* *"OPXXXS" = Option Put* *"OCXXXS" = Option Call* *"OXXXPS" = MERVAL Option* *"ESXXXX" = Stock* *"DBXXXX" = Bond*	*String*	*Conditional field where SecurityListRequestType is 1 = CFICode*

POWERED BY PRIMARY

			"XXWXXX" = Swap *“FXXXXX” = Futures Spread* *“EMXXXX”= CEDEAR* *“DTXXXX”= Financial Trust* *“RPXXXX”= Repurchase* *“MRIXXX”= Index* *“MXXXXX”= Undefined*		
263	SubscriptionRequestType	N	0 = Snapshot 1 = Snapshot + Updates 2 = Disable Previous Snapshot + Update	Char	Defines the type of subscription. By default Snapshot subscription.
	Standard Trailer	Y			

POWERED BY PRIMARY

Security List (MsgType = y)

The Security List message is used to return a list of securities that matches the criteria specified in a Security List Request.

Security List (MsgType = y)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = y		
320	SecurityReqID	Y		String	Identifier of Security List Request message that it is replying to
322	SecurityResponseID	Y		String	Identifier for each Security List message
560	SecurityRequestResult	Y	0=Valid request 1=Invalid or unsupported request 2=No instruments found that match selection criteria 5=Request was rejected because the CFICode specified is not supported	Int	Result of request identified by SecurityReqID.
<559>	*SecurityListRequestType*	*N*	*4 = All Securities*	*Int*	*Type of Security List Request was made*
1301	MarketID	N	"ROFX"	String	Identifies the market which lists and trades the instrument
1300	MarketSegmentID	N	E.g.: "DDF", "DDA"	String	Identifies the market segment.
393	TotNoRelatedSym	Y		Int	Total number of securities for request. For use in fragmented messages.
893	LastFragment	Y	N = Not Last Message Y = Last Message	Boolean	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented. Currently the Security List message is sent fragmented, a message for each segment.

→		***Block SecListGrp***				Specifies the number of repeating symbols (instruments) specified
146		NoRelatedSym	C	> = 1	NumInGroup (Int)	Indicates the number of instruments contained in this message. It is omitted when there are no instruments that meet the selection criteria.
→→	*<9996>*	*ContractPositionNumber*	*N*		*Long*	*Indicates the order number for the instrument. This field provides a specific order (defined by the Exchange) for contracts, allowing their classification according to criteria: Segment - Type - Product among others.*
→→		Block Instrument	Y			
→→→	***107***	*SecurityDesc*	*Y*		*String*	*Security description. Can be used to provide an optional textual description for a financial instrument.*
→→→	***228***	*Factor*	*Y*		*Float*	*For Derivatives: Contract Value Factor by which price must be adjusted to determine the true nominal value of one derivatives contract.* *(Qty * Price) * Factor = Nominal Value.*
→→→	***461***	*CFICode*	*Y*	*"FXXXSX" = Future* *"OPXXXS" = Option Put* *"OCXXXS" = Option Call* *"OXXXPS" = MERVAL Option* *"ESXXXX" = Stock* *"DBXXXX" = Bond* *"XXWXXX" = Swap* *"FXXXXX" = Futures Spread* *"EMXXXX"= CEDEAR* *"DTXXXX"= Financial Trust* *"RPXXXX"= Repurchase* *"MRIXXX"= Index* *"MXXXXX"= Undefined*	*String*	*Classification of Financial Instruments values.* *If an Option: StrikePrice and StrikeCurrency are required*

→→→	***231***	*ContractMultiplier*	*C*		*Float*	*Indicates the ratio or multiplier to convert "nominal" units to total units. Present if the security has this information associated.*
→→→	***200***	*MaturityMonthYear*	*C*	*YYYYMM*	*Month-Year*	*Month and Year of the maturity. Applicable for standardized derivatives which are typically only referenced by month and year; ex: futures, options, bonds, stocks and futures spread.*
→→→	***541***	*MaturityDate*	*C*	*YYYYMMDD*	*LocalMkt Date*	*Specifies date of maturity (a full date). Present when MaturityMonthYear (=200) is present. Enrich the information in Field 200.*
→→→	***202***	*StrikePrice*	*C*		*Price*	*Required when, CFICode is OPXXXS or OCXXXS*
→→→	***947***	*StrikeCurrency*	*C*	*E.g.:* *ARS = Argentine pesos* *USD = U.S. dollars*	*String*	*Currency in which the StrikePrice is denominated. Required when, CFICode is OPXXXS or OCXXXS.*
→→→	***969***	*MinPriceIncrement*	*Y*		*Float*	*Minimum Pricing Increment.*
→→→	***<5023>***	*TickSize*	*Y*		*Qty*	*Minimum permitted size change. Cannot be 0 (zero)*
→→→	***<5514>***	*InstrumentPricePrecision*	*Y*		*Int*	*Number of decimals in prices.*
→→→	***<7117>***	*InstrumentSizePrecision*	*Y*		*Int*	*Number of decimals in size.*
→→	**15**	Currency	N	E.g: ARS = Argentine pesos USD = U.S. dollars	String	Identifies currency used for price. Absence of this field is interpreted as the default for the security. It is recommended that systems provide the currency value whenever possible
→→		***Block FinancingDetails***	C			
→→→	***917***	*EndDate*	*C*	*E.g: "20150507"*	*LocalMkt Date*	*End date of a financing deal, i.e. the date the seller reimburses the buyer and takes back control of the collateral.*

						Required only when CFI Code is RPXXXX (Repurchase).
→→	Block UndInstrmtGrp		N			Underlying security's Symbol.
→→	***Block SecurityTradingRules***					
→→→	***Block BaseTradingRules***		N			This block contains the base trading rules
→→→→	***1140***	*MaxTradeVol*	*N*		*Float*	*The maximum order quantity that can be submitted for a security.*
→→→→	***562***	*MinTradeVol*	*N*		*Float*	*The minimum order quantity that can be submitted for a security.*
→→→→	***Block LotTypeRules***		N			
1234	NoLotTypeRules				NumInGroup (Int)	Number of Lot Types
→→→→→	***1093***	*LotType*	*N*	*3 = Block Lot*	*Char*	*Defines the lot type assigned to the order.*
→→→→→	***1231***	*MinLotSize*	*C*		*Qty*	*Minimum lot size allowed based on lot type specified in LotType(1093) If LotType=3 means the min lot size for Block Trade orders.*
→→→→→	***<5515>***	*MaxLotSize*	*N*		*Qty*	*Maximum lot size allowed based on lot type specified in LotType(1093)* *If LotType=3 means the max lot size for Block Trade orders.*
→→→→	***Block PriceLimits***					
→→→→→	***1148***	*LowLimitPrice*	*C*		*Price*	*Minimum authorized price at which an instrument can trade. Present if the security has this information associated.*
→→→→→	***1149***	*HighLimitPrice*	*C*		*Price*	*Maximum authorized price at which an instrument can trade. Present if the security has this information associated.*

→→→	***Block TradingSessionRulesGrp***					This block contains the base trading rules
1309	NoTradingSessionRules				NumInGroup (Int)	
→→→→	***336***	*TradingSessionID*	*N*		*String*	*Identifier for the trading session* *Must be provided if NoTradingSessions > 0* *Set to [N/A] if values are not specific to trading session.*
→→→→	***Block TradingSessionRules***					Contains trading rules specified at the trading session level
→→→→→	***Block OrdTypeRules***		N			Specifies the order types that are valid for trading. The scope of the rule is determined by the context in which the component is used. In this case, the scope is trading session.
1237	NoOrdTypeRules		N		NumInGroup (Int)	Number of order types
→→→→→→	***40***	*OrdType*	*N*	*1 = Market* *2 = Limit* *K = Market With Left Over as Limit*	*Char*	*Indicates order types that are valid for the specified market segment.*
→→→→→	***Block TimeInForceRules***		N			Specifies the time in force rules that are valid for trading. The scope of the rule is determined by the context in which the component is used. In this case, the scope is trading session
1239	NoTimeInForceRules		N	1	NumInGroup (Int)	Number of time in force techniques
→→→→→→	***59***	*TimeInForce*	N	*0 = Day* *1 = Good Till Cancel* *3 = Inmediate or Cancel* *4 = Fill or Kill*	*Char*	*Indicates time in force techniques that are valid for the specified market segment*

POWERED BY PRIMARY

→→→→→	***Block ExecInstRules***	N				Specifies the execution instructions that are valid for trading. The scope of the rule is determined by the context in which the component is used. In this case, the scope is trading session
1232	NoExecInstRules	N	1	NumInGroup (Int)		Number of execution instructions
→→→→→→	***1308*** *ExecInstValue*	*N*	*G = All or None*	*Char*		*Indicates execution instructions that are valid for the specified market segment*
	Standard Trailer	Y				

Security Status Request (MsgType = e)

The Security Status Request message provides for the ability to request the status of a security. One or more Security Status messages are returned as a result of a Security Status Request message.

Security Status Request (MsgType = e)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = e		
324	SecurityStatusReqID	Y		String	Identifier of Security Status Request message. Must be unique, or the ID of previous Security Status Request to disable if SubscriptionRequestType = Disable previous Snapshot + Updates Request (2).
263	SubscriptionRequestType	Y	0 = Snapshot 1 = Snapshot + Updates (Subscribe) 2 = Disable previous Snapshot + Update Request (Unsuscribe)	String	Subscription Request Type
→	Block Instrument	Y			A
→→ *55*	*Symbol*	*Y*	*[N/A] = all symbols*	*String*	*Ticker symbol or [N/A] to request the security status for all symbols.*
	Standard Trailer	Y			

POWERED BY PRIMARY

Security Status (MsgType = f)

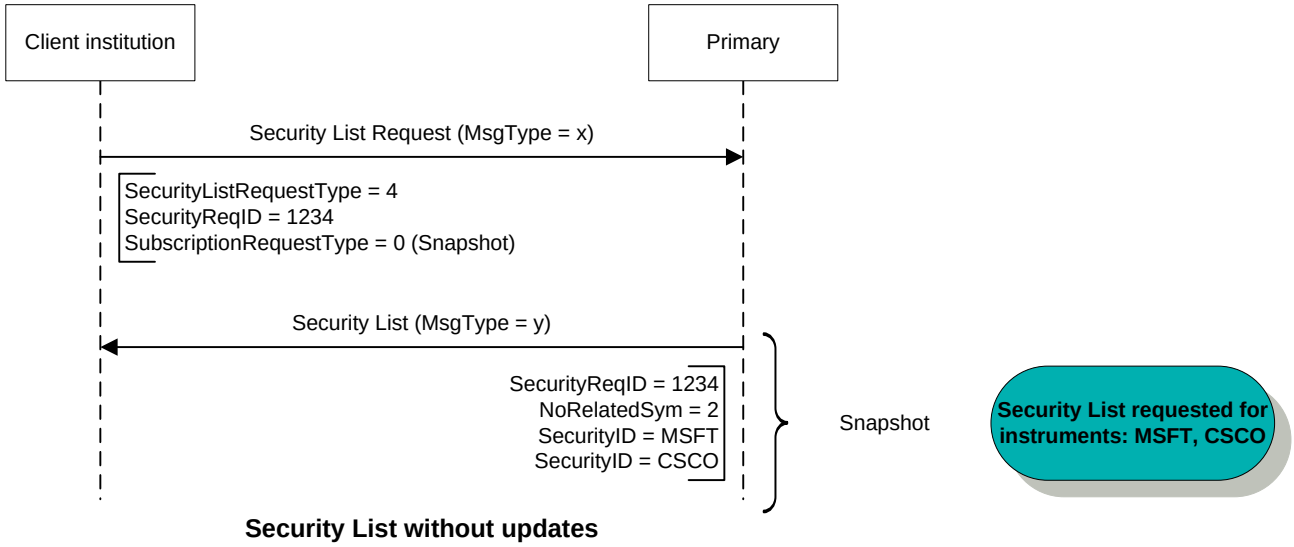
The Security Status message provides for the ability to report changes in status to a security. The Security Status message contains fields to indicate trading status, corporate actions, financial status of the company. The Security Status message is used by one trading entity (for instance an exchange) to report changes in the state of a security.

Security Status (MsgType = f)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = f		
324	SecurityStatusReqID	Y		String	Identifier of Security Status Request message that it is replying to. Must be unique, or the ID of previous Security Status Request to disable if SubscriptionRequestType = Disable previous Snapshot + Updates Request (2).
→	Block Instrument	Y			
→→ ***55***	*Symbol*	*Y*	*Ex.:DOOct2014*	*String*	*Ticker symbol*
326	SecurityTradingStatus	Y	2 = TRADING_HALT 3 = RESUME	Int	Identifies the trading status applicable to the transaction. When SecurityTradingStatus is 2 the symbol is suspended, if it is 3 the symbol is enabled to Trade.
	Standard Trailer	Y			

POWERED BY PRIMARY

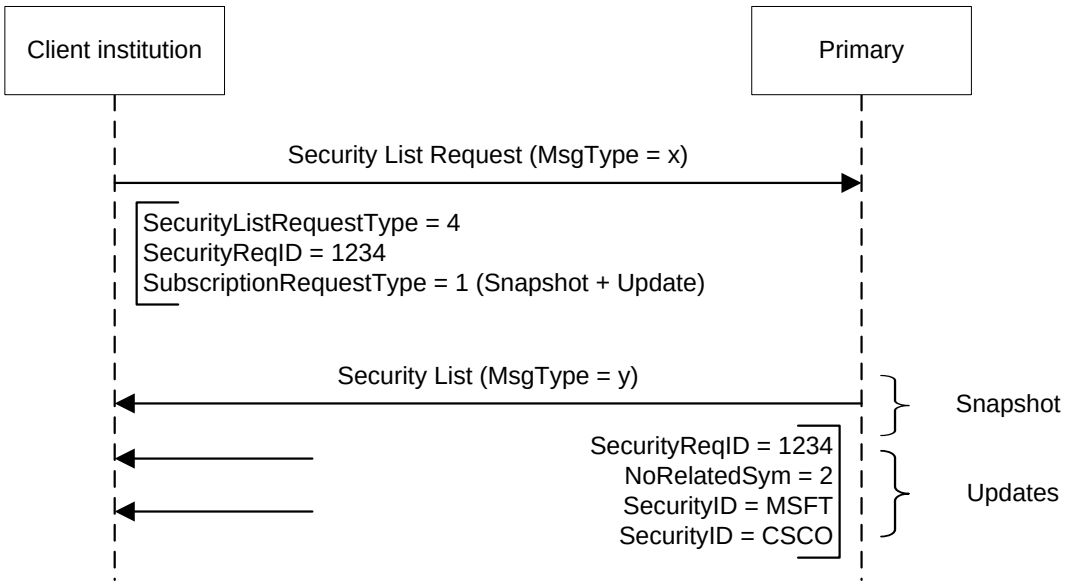
Message Flow - Security Definition

Security List without updates



Security List without updates

Security List with updates



Security List with updates

POWERED BY PRIMARY

Application Messages – Post trade messages

Message specification

Trade Capture Report Request (MsgType = AD): for Regular Trades by Account

The Trade Capture Report Request can be used to:

• Request one or more trade capture reports based upon selection criteria provided on the trade capture report request

Trade Capture Report Request (MsgType = AD)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = AD		
568	TradeRequestID	Y		String	Identifier for the trade request.
569	TradeRequestType	Y	1= MatchedTradesMatchingCriteria	Int	Type of "trade capture report".
828	TrdType	N	0=RegularTrade	Int	To request all trades of a specific trade type.
830	TransferReason	C	"AccountDetail"	String	To request all trades for a specific transfer reason, "AccountDetail" for all trades belonging a specific account.
→	Block Parties	C			Repeating group below should contain unique combinations of PartyID, PartyIDSource, and PartyRole. Conditionally required when Trade capture report is request by customer account.
453	NoPartyIDs	Y		NumInGroup	Number of PartyID (448), PartyIDSource (447), and PartyRole (452) entries
→→ ***448***	*PartyID*	*Y*	*Ex. "000100002"*	*String*	*Account name*
→→ ***447***	*PartyIDSource*	*Y*	*D-Proprietary*	*Char*	*Used to identify class source of PartyID value (e.g. BIC). Required if PartyID is specified. Required if NoPartyIDs > 0.*

POWERED BY PRIMARY

→→	*452*	*PartyRole*	*Y*	*24- Customer Account*	*Int*	*Identifies the type or role of the PartyID (448) specified.* *See "Appendix 6-G - Use of <Parties> Component Block"* *(see Volume : "Glossary" for value definitions)*
		Standard Trailer	Y			

Trade Capture Report (MsgType = AE): for Regular Trades by Account

The Trade Capture Report message can be:

• Used to report trades between counterparties.

• Sent as a reply to a Trade Capture Report Request.

Trade Capture Report (MsgType = AE)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = AE		
571	TradeReportID	Y		String	TradeReportID is conditionally required in a message-chaining model in which a subsequent message may refer to a prior message via TradeReportRefID. The alternative to a message-chain model is an entity-based model in which TradeID is used to identify a trade. In this case, TradeID is required and TradeReportID can be optionally specified.
568	TradeRequestID	N		String	Request ID if the Trade Capture Report is in response to a Trade Capture Report Request.
828	TrdType	Y	0=REGULAR_TRADE	Int	Type of Trade
60	TransactTime	N		UTCTimestamp	Timestamp when the business transaction represented by the message occurred. i.e. 20131230-19:36:59
570	PreviouslyReported	N	N= Not reported to counterparty	Boolean	Indicates if the trade capture report was previously reported to the counterparty.
748	TotNumTradeReports	N		int	Total number of trade reports returned.
912	LastRptRequested	N	N= Not last message Y= Last message	Boolean	Indicates whether this message is that last report message in response to a request, such as Order Mass Status Request
32	LastPx	Y		Price	Trade Price

31		LastQty	Y		Qty	Trade Quantity
→		Block Parties	N			Used to specify the parties for the trades to be returned (clearing firm, execution broker, trader id, etc.)
453		NoPartyIDs	Y		NumInGroup	Number of PartyID (448), PartyIDSource (447), and PartyRole (452) entries
→→	***448***	*PartyID*	*Y*	*Ex."jramirez".*	*String*	*User name*
→→	***447***	*PartyIDSource*	*Y*	*D-Proprietary*	*Char*	*Used to identify class source of PartyID value (e.g. BIC). Required if PartyID is specified. Required if NoPartyIDs > 0.*
→→	***452***	*PartyRole*	*Y*	*11-OrderOriginationTrader*	*Int*	*Identifies the type or role of the PartyID (448) specified.* *See "Appendix 6-G - Use of <Parties> Component Block"* *(see Volume : "Glossary" for value definitions)*
→		Block Instrument	Y			Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages".
→		Block TrdCapRptSideGrp	Y			
552		NoSides	Y	1 = One Side 2 = Both Sides		Number of Sides.
→→	***54***	*Side*	*Y*	*1= Buy, 2= Sell*	*char*	*Side of order*
→→	***1***	*Account*	*N*	*Ex. "000100023"*	*String*	*Account mnemonic as agreed between buy and sell sides, e.g. broker and institution or investor/intermediary and fund manager.* *"*" when account is from counterparty.*
		Standard Trailer	Y			

POWERED BY PRIMARY

Trade Capture Report Request (MsgType = AD): for Regular Trades by Symbol

The Trade Capture Report Request can be used to:

• Request one or more trade capture reports based upon selection criteria provided on the trade capture report request

Trade Capture Report Request (MsgType = AD)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = AD		
568	TradeRequestID	Y		String	Identifier for the trade request.
569	TradeRequestType	Y	1= MatchedTradesMatchingCriteria	Int	Type of "trade capture report".
828	TrdType	N	0=RegularTrade	Int	To request all trades of a specific trade type.
55	Symbol	C		String	Ticker symbol. Conditionally required when Trade capture report is request by symbol.
	Standard Trailer	Y			

Trade Capture Report (MsgType = AE): for Regular Trades by String

The Trade Capture Report message can be:

• Used to report trades between counterparties.

• Sent as a reply to a Trade Capture Report Request.

Trade Capture Report (MsgType = AE)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = AE		
571	TradeReportID	Y		String	TradeReportID is conditionally required in a message-chaining model in which a subsequent message may refer to a prior message via TradeReportRefID. The alternative to a message-chain model is an entity-based model in which TradeID is used to identify a trade. In this case, TradeID is required and TradeReportID can be optionally specified.
568	TradeRequestID	N		String	Request ID if the Trade Capture Report is in response to a Trade Capture Report Request.
828	TrdType	Y	0=REGULAR_TRADE	Int	Type of Trade
60	TransactTime	N		UTCTimestamp	Timestamp when the business transaction represented by the message occurred. i.e. 20131230-19:36:59
570	PreviouslyReported	N	N= Not reported to counterparty	Boolean	Indicates if the trade capture report was previously reported to the counterparty.
748	TotNumTradeReports	N		int	Total number of trade reports returned.
912	LastRptRequested	N	N= Not last message Y= Last message	Boolean	Indicates whether this message is that last report message in response to a request, such as Order Mass Status Request

32		LastPx	Y		Price	Trade Price
31		LastQty	Y		Qty	Trade Quantity
→		Block Instrument	Y			Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages".
→		Block TrdCapRptSideGrp	Y			
552		NoSides	Y	1 = One Side 2 = Both Sides		Number of Sides.
→→	*54*	*Side*	*Y*	*1= Buy, 2= Sell*	*char*	*Side of order*
→→		Block Parties	N			Used to specify the parties for the trades to be returned (clearing firm, execution broker, trader id, etc.)
453		NoPartyIDs	Y		NumInGroup	Number of PartyID (448), PartyIDSource (447), and PartyRole (452) entries
→→→	*448*	*PartyID*	*Y*	*Ex."jramirez".*	*String*	*User name*
→→→	*447*	*PartyIDSource*	*Y*	*D-Proprietary*	*Char*	*Used to identify class source of PartyID value (e.g. BIC). Required if PartyID is specified. Required if NoPartyIDs > 0.*
→→→	*452*	*PartyRole*	*Y*	*11-OrderOriginationTrader*	*Int*	*Identifies the type or role of the PartyID (448) specified.* *See "Appendix 6-G - Use of <Parties> Component Block"* *(see Volume : "Glossary" for value definitions)*
→→	*1*	*Account*	*N*	*Ex. "000100023"*	*String*	*Account mnemonic as agreed between buy and sell sides, e.g. broker and institution or investor/intermediary and fund manager.* *"*" when account is from counterparty.*
		Standard Trailer	Y			

POWERED BY PRIMARY

Trade Capture Report (MsgType = AE): for Block Trades

The Trade Capture Report message can be:

• Used to report trades between counterparties.

• Can be sent unsolicited between counterparties.

• Sent as a reply to a Trade Capture Report Request.

• Can be used to send a Block Trade to be confirmed by the involved parties

• Can be used to notify about the new Block Trade to be confirmed

• Can be used to notify the Block Trade acceptation, or declination.

Trade Capture Report (MsgType = AE)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = AE		
571	TradeReportID	Y		String	TradeReportID is conditionally required in a message-chaining model in which a subsequent message may refer to a prior message via TradeReportRefID. The alternative to a message-chain model is an entity-based model in which TradeID is used to identify a trade. In this case, TradeID is required and TradeReportID can be optionally specified.
572	TradeReportRefID	C		String	The TradeReportID that is being referenced for some action, such as correction or cancellation. Sent when TradeReportType = 2 or TradeReportType = 3.
487	TradeReportTransType		0 = New	Int	Identifies Trade Report message transaction type.
60	TransactTime	N		UTCTimestamp	Timestamp when the business transaction represented by the message occurred. i.e. 20131230-19:36:59
75	TradeDate	N		String	Indicates date of trade referenced in this message in YYYYMMDD format.
570	PreviouslyReported	N	N= Not reported to counterparty	Boolean	Indicates if the trade capture report was previously reported to the counterparty.

828	TrdType	Y	1= BLOCK_TRADE	Int	Type of Trade
856	TradeReportType	N	0=Submit 1=Alleged 2=Accept 3=Decline	Int	Type of Trade Report.
880	TrdMatchID	C		String	Identifier assigned to a trade by a matching system. Sent only when TradeReportType = 2 or TradeReportType = 3.
32	LastPx	Y		Price	Trade Price
31	LastQty	Y		Qty	Trade Quantity
150	ExecType	C	4 - Canceled F = Trade	char	Type of execution being reported. Sent the value = "F" when TradeReportType = 2, and "4" when the TradeReportType = 3.
→	Block RootParties	N			Insert here the set of "Root Parties" fields defined in "common components of application messages". Sent only when TradeReportType = 3.
→	Block Instrument	Y			Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages".
→	Block TrdCapRptSideGrp	Y			
552	NoSides	Y	2 = Both Sides		Number of Sides.
→→ ***54***	*Side*	*Y*	*1= Buy, 2= Sell*	*char*	*Side of order*
→→ ***1***	*Account*	*N*		*String*	*Account mnemonic as agreed between buy and sell sides, e.g. broker and institution or investor/intermediary and fund manager.*
→→	Block TradeReportOrderDetail				
→→→ ***37***	*OrderID*	*N*	*0, -*	*String*	*Unique identifier for Order as assigned by sell-side (broker, exchange, ECN). Currently not sent.*

POWERED BY PRIMARY

→→→	*40*	*OrderType*	*N*	*2 = Limit*	*Char*	*Order type from the order associated with the trade. Sent only when TradeReportType=0*
		Standard Trailer	Y			

TradeCaptureReportAck (MsgType = AR): for Block Trades

The Trade Capture Report Ack message can be:

• Used to acknowledge trade capture reports received from counterparty

• Used to inform about the Block Trades reception by the market

• Used to accept or reject a trade capture report received from a counterparty (this means accept or decline the Block Trade)

Trade Capture Report Ack(MsgType = AR)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = AR		
880	TrdMatchID	Y		String	Identifier assigned to a trade by a matching system.
571	TradeReportID	Y		String	Unique identifier of "trade capture report".
856	TradeReportType	Y	1=Alleged 3=Decline	Int	Type of trade report.
828	TrdType	Y	1 = Block Trade	Int	Type of Trade
→	Block Instrument	Y			Insert here the set of "Instrument" fields defined in "common components of application messages".
58	Text	C		String	Free format text string. Sent only when ExecType = 8, and used to inform about cause of rejection.
60	TransactTime	N		UTCTimestamp	Timestamp when the business transaction represented by the message occurred.
150	ExecType	N	0 = New 8 = Rejected	Char	Type of Execution being reported: Uses subset of ExecType for Trade Capture Reports.
	Standard Trailer	Y			

POWERED BY PRIMARY

Trade Capture Report Request (MsgType = AD): for allocations and giveups

The Trade Capture Report Request can be used to:

• Request one or more trade capture reports based upon selection criteria provided on the trade capture report request

• Subscribe for trade capture reports based upon selection criteria provided on the trade capture report request.

Currently used to request all trades made with “temporal accounts” available to do “allocations” or “giveups”.

Trade Capture Report Request (MsgType = AD)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = AD		
568	TradeRequestID	Y		String	Identifier for the trade request
569	TradeRequestType	Y	0= All Trades	Int	Type of “trade capture report”.
263	SubscriptionRequestType	N	0= Snapshot	Char	Used to subscribe / unsubscribe for trade capture reports If the field is absent, the value 0 will be the default (snapshot only - no subscription)
828	TrdType	N	1001= Allocation 1002= GiveUp	Int	Type of trade
	Standard Trailer	Y			

POWERED BY PRIMARY

Trade Capture Report (MsgType = AE): for allocations and giveup

The Trade Capture Report message can be:

• Used to report trades between counterparties.

• Can be sent unsolicited between counterparties.

• Sent as a reply to a Trade Capture Report Request.

• Can be used to ask for operations available to be allocated or giveup

Trade Capture Report (MsgType = AE)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = AE		
571	TradeReportID	Y		String	TradeReportID is conditionally required in a message-chaining model in which a subsequent message may refer to a prior message via TradeReportRefID. The alternative to a message-chain model is an entity-based model in which TradeID is used to identify a trade. In this case, TradeID is required and TradeReportID can be optionally specified.
572	TradeReportRefID	N		String	The TradeReportID that is being referenced for some action, such as correction or cancellation.
568	TradeRequestID	N		String	Request ID if the Trade Capture Report is in response to a Trade Capture Report Request.
828	TrdType	Y	0=REGULAR_TRADE 1= BLOCK_TRADE	Int	Type of Trade
856	TradeReportType	N	1=Alleged	Int	Type of Trade Report.
487	TradeReportTransType		0= New	Int	Identifies Trade Report message transaction type.
60	TransactTime	N		UTCTimestamp	Timestamp when the business transaction represented by the message occurred. i.e. 20131230-19:36:59

75	TradeDate	N		String	Indicates date of trade referenced in this message in YYYYMMDD format.
570	PreviouslyReported	N	N= Not reported to counterparty	Boolean	Indicates if the trade capture report was previously reported to the counterparty.
818	SecondaryTradeReportID	N		String	Used to send the operation ID.
880	TrdMatchID	N			Identifier assigned to a trade by a matching system.
748	TotNumTradeReports	N		int	Total number of trade reports returned.
912	LastRptRequested	N	N= Not last message Y= Last message	Boolean	Indicates whether this message is that last report message in response to a request, such as Order Mass Status Request
32	LastPx	Y		Price	Trade Price
31	LastQty	Y		Qty	Trade Quantity
150	ExecType	N	F = Trade	char	Type of execution being reported.
→	Block RootParties	N			Insert here the set of "Root Parties" fields defined in "common components of application messages".
→	Block Instrument	Y			Insert here the set of "Instrument" (symbology) fields defined in "Common Components of Application Messages".
→	Block TrdCapRptSideGrp	Y			
552	NoSides	Y	1 = One Side 2 = Both Sides		Number of Sides.

POWERED BY PRIMARY

→→	*54*	*Side*	*Y*	*1= Buy, 2= Sell*	*char*	*Side of order*
→→	*1*	*Account*	*N*		*String*	*Account mnemonic as agreed between buy and sell sides, e.g. broker and institution or investor/intermediary and fund manager.*
→→		Block TradeReportOrderDetail				
→→→	*37*	*OrderID*	*N*		*String*	*Unique identifier for Order as assigned by sell-side (broker, exchange, ECN).*
		Standard Trailer	Y			

POWERED BY PRIMARY

AllocationInstruction (MsgType = J)

The Allocation Instruction message provides the ability to specify how an order or set of orders should be subdivided amongst one or more accounts. Currenty used to request the allocation of the order to other of the broker accounts in case of an "*allocation";* or another account from a different broker in case of a "*giveup"*.

AllocationInstruction (MsgType = J)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = J		
70	AllocID	Y		String	Unique ID for this message
71	AllocTransType	Y	0 = New	Char	Identifies allocation transaction type. i.e. New, Cancel, Replace
626	AllocType	Y	1 = Calculated	Int	Specifies the purpose or type of Allocation message
→	Block Instrument	Y			Insert here the set of "Instrument" fields defined in "common components of application messages".
6	AvgPx	N		Price	Calculated average price of all fills on this order.
53	Quantity	Y		Qty	Total quantity (e.g. number of shares) allocated to all accounts, or that is Ready-To-Book
54	Side	Y	1 = Buy 2 = Sell	Char	Side of order
75	TradeDate	Y		LocalMktDate	Indicates date of trade referenced. i.e. Mon Dec 30 16:36:59 ART 2013
58	Text	N	"Allocation" "Giveup"	String	Free format text string
754	AutoAcceptIndicator		N = No	Boolean	Indicates if Allocation has been automatically accepted on behalf of the Carry Firm by the Clearing House.
828	TrdType	N	1001=Allocation 1002=Give Up	Int	Specifies trade type when a trade is being reported.
857	AllocNoOrdersType	N	1= ExplicitListProvided	Int	Indicates how the orders being booked and allocated by this message are identified, i.e. by explicit definition in the NoOrders group or not.

→		Block Alloc	C			Conditionally required except when AllocTransType = Cancel, or when AllocType = "Ready-to-book" or "Warehouse instruction"
→→	**78**	NoAllocs	N		NumIn Group (Int)	Number of repeating AllocAccount (79)/AllocPrice (366) entries.
→→	***79***	*AllocAccount*	*C*		*String*	*Required if NoAllocs > 0. Must be first field in repeating group.* *Conditionally required except when for AllocTransType="Cancel", or when AllocType= "Ready-To-Book" or "Warehouse instruction".*
→→	***366***	*AllocPrice*	*C*		*Price*	*AllocAccount plus AllocPrice form a unique Allocs entry.* *Executed price for an AllocAccount (79) entry*
→→	***80***	*AllocQty*	*C*		*Qty*	*Conditionally required except when for AllocTransType="Cancel", or when AllocType= "Ready-To-Book" or "Warehouse instruction".* *Quantity to be allocated to specific sub-account.*
→		Block OrdAlloc	C			Indicates number of orders to be combined for allocation. If order(s) were manually delivered set to 1 (one).Required when AllocNoOrdersType = 1
→→	**73**	NoOrders	N		NumIn Group (Int)	Indicates number of orders to be combined for average pricing and allocation.
→→	***11***	*ClOrdID*	*C*	*i.e. 16008*	*String*	*Order identifier assigned by client if order(s) were electronically delivered over FIX (or otherwise assigned a ClOrdID) and executed. If order(s) were manually delivered (or otherwise not delivered over FIX) this field should contain string "MANUAL". Note where an order has undergone one or more cancel/replaces, this should be the ClOrdID of the most recent version of the order.* *Required when NoOrders(73) > 0 and must be the first repeating field in the group.*

POWERED BY PRIMARY

➔➔	*37*	*OrderID*	*C*	*i.e.. 20029*	*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.*
		Standard Trailer	Y			

AllocationInstructionAck (MsgType = P)

The Allocation Instruction Ack message is used to acknowledge the receipt of and provide status for an Allocation Instruction message.Currently used only in case of allocations of Giveup.

AllocationInstructionAck (MsgType = P)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = P		
70	AllocID	Y		String	Unique ID for this message
87	AllocStatus	Y	3 = Received	Int	Identifies status of allocation.
58	Text	N	"Pendiente de confirmar"	String	Free format text string.
60	TransactTime	N		UTCTimestamp	Date/Time Allocation Instruction Ack generated. i.e. 20131230-21:04:26
	Standard Trailer	Y			

POWERED BY PRIMARY

Confirmation (MsgType = AK): for allocations and giveups

The Confirmation messages are used to provide individual trade level confirmations from the sell side to the buy side. Unlike the allocation message, the confirmation message operates at an allocation account (trade) level rather than block level, allowing for the affirmation or rejection of individual confirmations.

Confirmation (MsgType = AK)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = AK		
664	ConfirmID	Y	1	String	Unique ID for this message
772	ConfirmRefID	Y		String	Mandatory if ConfirmTransType is Replace or Cancel
665	ConfirmStatus	Y	1 = Received 4 = Confirmed 5 = RequestRejected	Int	Identifies the status of the Confirmation.
666	ConfirmTransType	Y	0 = New 2 = Cancel	Int	Identifies the Confirmation transaction type.
773	ConfirmType	Y	2 = Confirmation 3 = ConfirmationRequestRejected	Int	Identifies the type of Confirmation message being sent.
70	AllocID	Y		String	Used to refer to an earlier Allocation Instruction via its secondary identifier.
60	TransactTime	Y		UTCTimestamp	Timestamp when the business transaction represented by the message occurred. i.e. 20131230-19:40:11
75	TradeDate	Y		LocalMktDate	Indicates date of trade referenced in this message in YYYYMMDD format. i.e. 20131230
→	Block Instrument	Y			Insert here the set of "Instrument" fields defined in "common components of application messages".
58	Text	N	"Confirmada" "Cancelada por el usuario" "Ejecutada"	String	Free format text string.

POWERED BY PRIMARY

54	Side	Y	1 = Buy 2 = Sell	Int	Side of order
79	AllocAccount	Y		String	Account number for the trade being confirmed by this message
80	AllocQty	Y		Qty	Quantity to be allocated to specific sub-account.
6	AvgPx	Y		Price	Gross price for the trade being confirmed
381	GrossTradeAmt	Y	0	Amt	Total amount traded (i.e. quantity * price) expressed in units of currency. AllocQty(80) * AvgPx(6)
118	NetMoney	Y	0	Amt	Total amount due as the result of the transaction (e.g. for Buy order - principal + commission + fees) reported in currency of execution.
828	TrdType		1001= Allocation 1002 = Give Up	Int	Type of Trade
861	ReportedPx	N		Price	Reported price (may be different to AvgPx in the event of a marked-up or marked-down principal trade)
→	Block Parties	N			Insert here the set of "Parties" (firm identification) fields defined in "Common Components of Application Messages"
453	NoParties	C		NumIn Group (Int)	Indicates number of orders to be combined for allocation. If order(s) were manually delivered set to 1 (one).Sent only for GiveUps.
→→ ***448***	*PartyID*	*C*		*String*	*Party identifier/code.*
→→ ***447***	*PartyIDSource*	*C*	*D = Propietary*	*Char*	*Identifies class or source of the PartyID (448) value. Required if PartyID is specified. Note: applicable values depend upon PartyRole (452) specified.*
→→ ***452***	*PartyRole*	*C*	*11 =OrderOriginationTrader*	*Int*	*Identifies the type of PartyID (e.g. Executing Broker). Required if NoPartyIDs > 0.*
→	Block OrdAllocGrp	Y			

POWERED BY PRIMARY

	73	NoOrders	C		NumIn Group (Int)	Indicates number of orders to be combined for allocation. If order(s) were manually delivered set to 1 (one).Required when AllocNoOrdersType = 1
→→	***11***	*ClOrdID*	*C*		*String*	*Order identifier assigned by client if order(s) were electronically delivered over FIX (or otherwise assigned a ClOrdID) and executed. Required when NoOrders(73) > 0 and must be the first repeating field in the group.*
→→	***37***	*OrderID*	*C*		*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.* *This field is not used either for "allocations" or for "give ups".*
→		Block CapacityConf	Y	0		OrderCapacity repeating group instances.
→		Block UndInstrmt	N	0		Underlying repeating group instances.
→		Block InstrmtLeg	N	0		InstrumentLeg repeating group instances.
		Standard Trailer	Y			

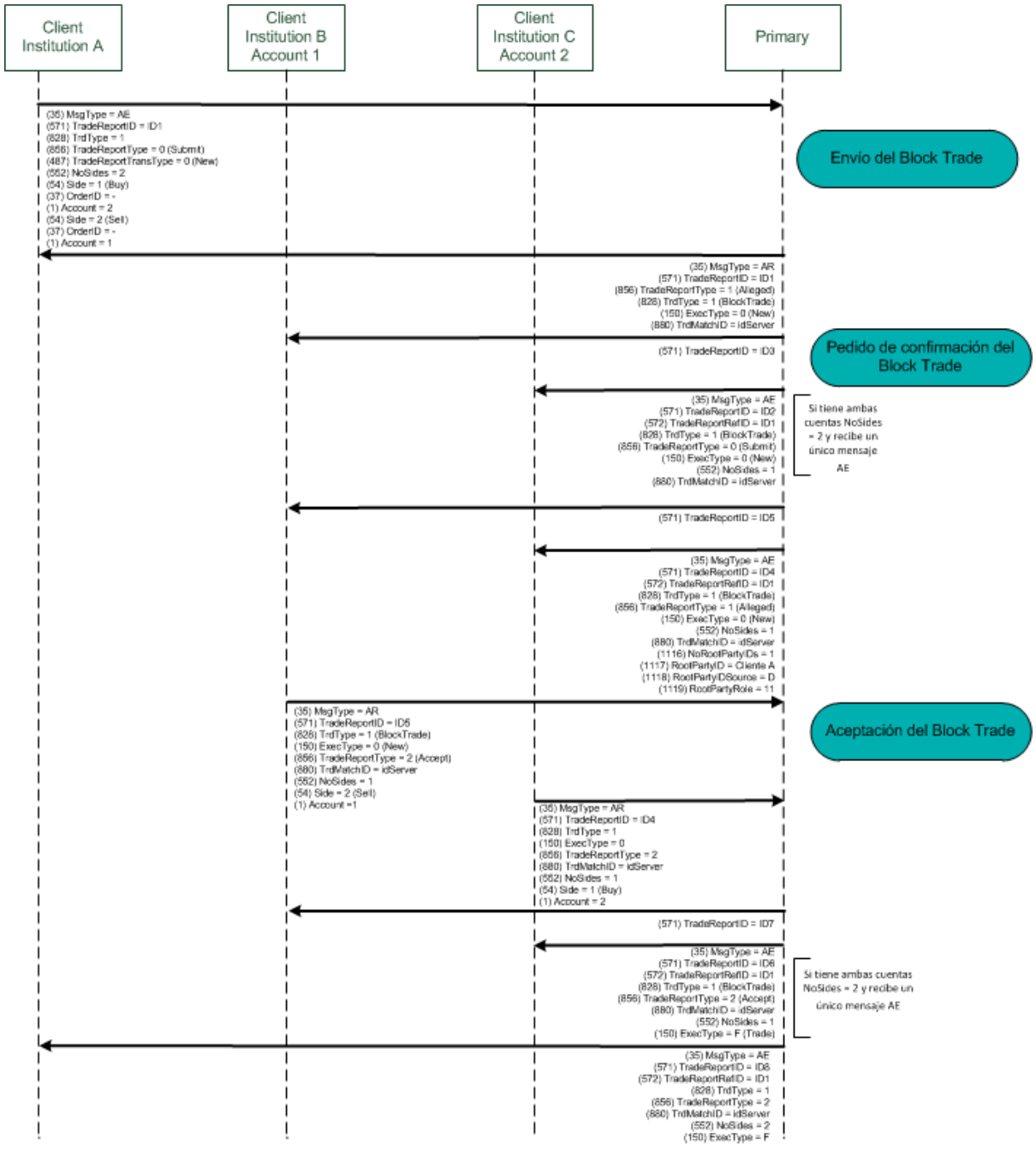
POWERED BY PRIMARY

ConfirmationAck (MsgType = AU)

The Confirmation Ack (aka Affirmation) message is used to respond to a Confirmation message. The ConfirmationAck message is used to confirm the acceptation o rejection for a “Giveup”.

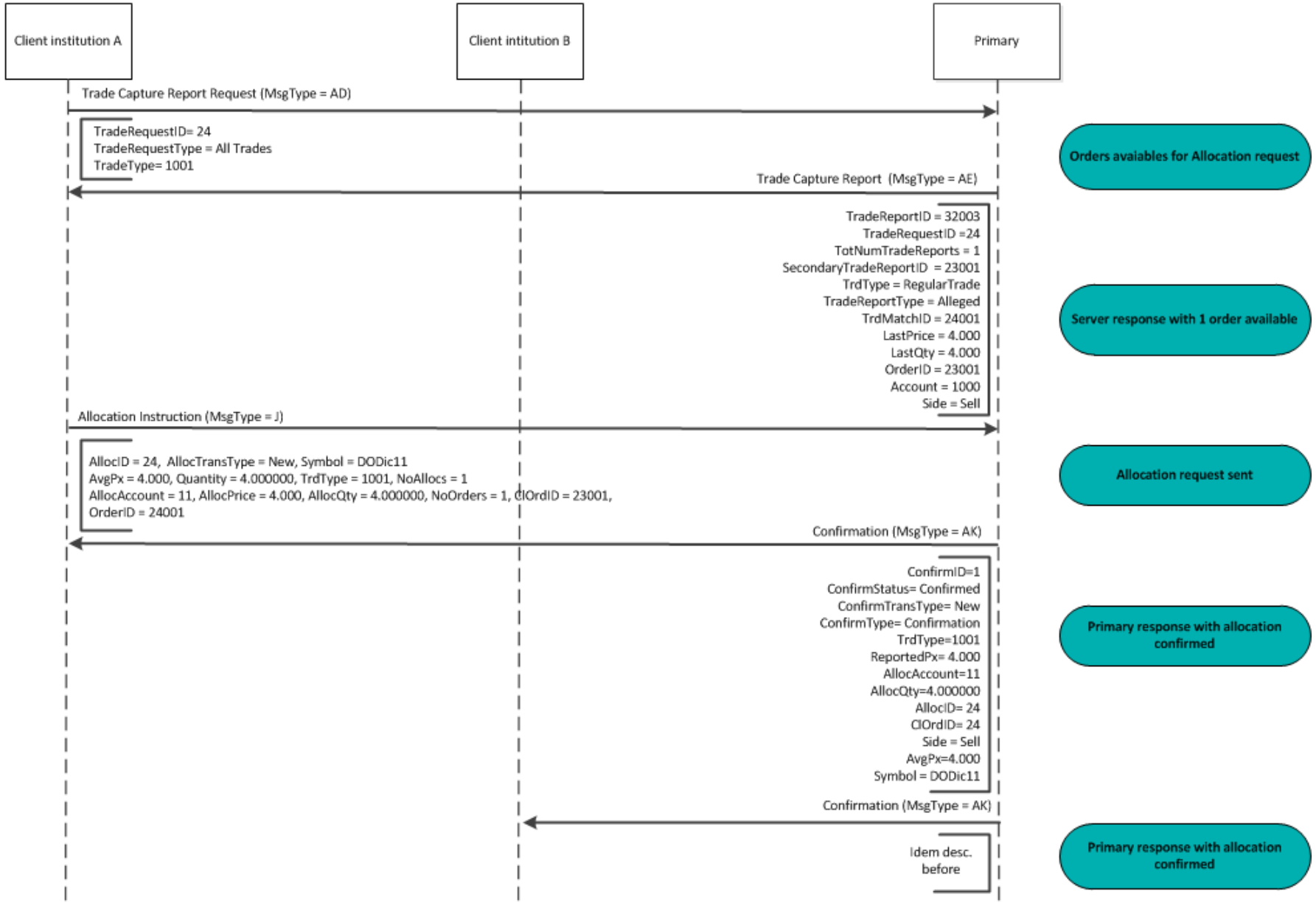
ConfirmationAck (MsgType = AU)					
Tag	**FIX Name**	**Req**	**Valid Values**	**Format**	**Description**
	Standard Header	Y	MsgType = AU		
664	ConfirmID	Y		String	Message reference for Confirmation
940	AfirmStatus	Y	2 = ConfirmRejected 3 = Affirmed	Int	Identifies the status of the ConfirmationAck.
75	TradeDate	Y	i.e. 20140103	LocalMktDate	Indicates date of trade referenced in this message in YYYYMMDD format.
60	TransactTime	Y		UTCTimestamp	Date/Time Confirmation Ack generated. i.e. 20131230-21:04:26
	Standard Trailer	Y			

POWERED BY PRIMARY

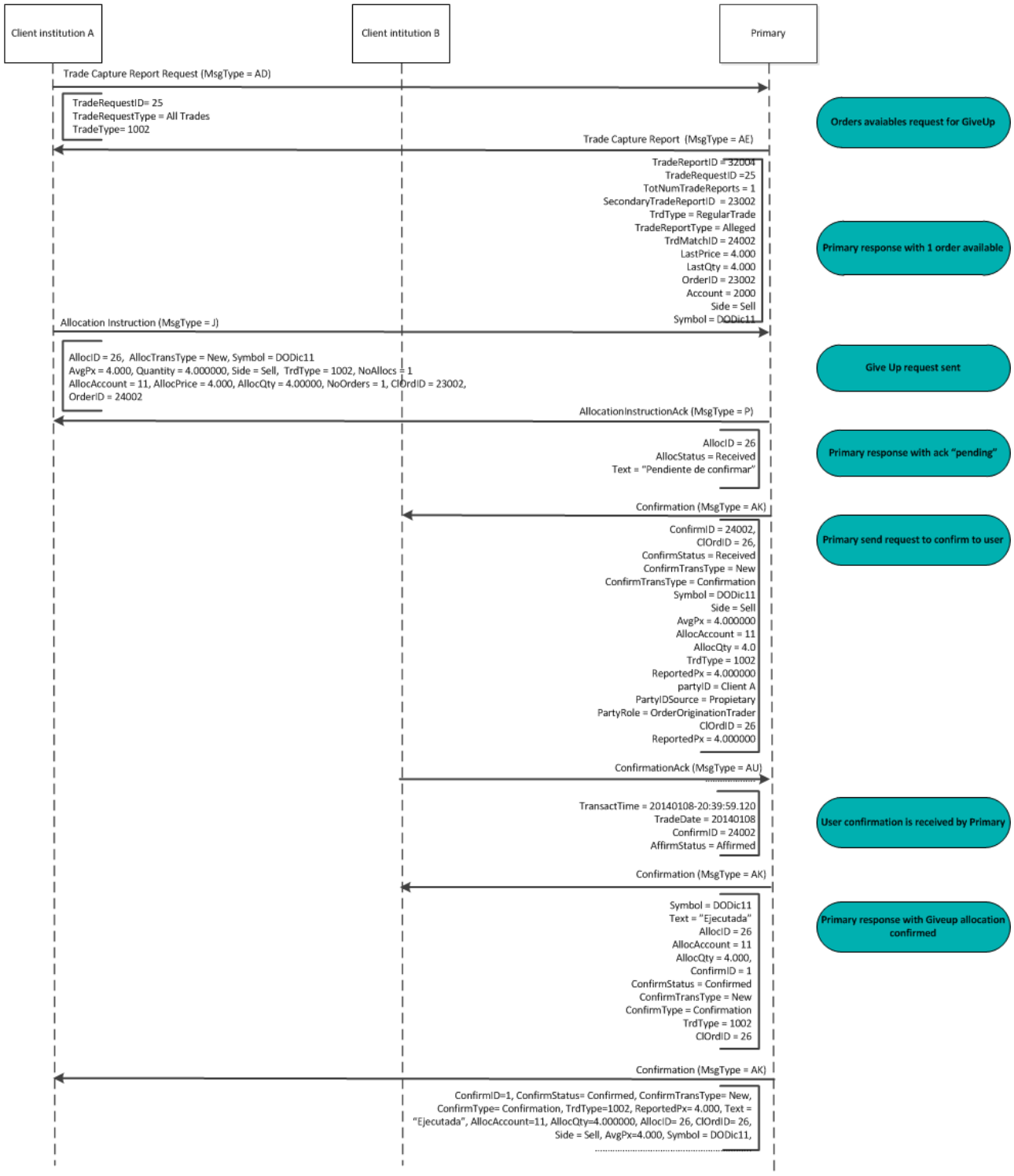


POWERED BY PRIMARY

Allocation Messages Flow



GiveUp Messages Flow



POWERED BY PRIMARY