Notices

This document is distributed for information purposes only and does not form part of or constitute an agreement with DeFinity. Current and prospective DeFinity products, services and processes are liable to change from time to time without notice and accordingly no liability of any nature (in contract, tort or otherwise, including for misrepresentation) can be accepted by DeFinity in relation to this document, unless otherwise agreed with DeFinity.

Usage of the GUI and API is subject to platform terms. A copy of the terms is available at

www.definitymarkets.com/agreement

Business Support

The DeFinity Business Support team operates 24/5.5 over three time zones and are available to deal with any problems or queries.

For support queries, please call or email

ops@definitymarkets.com

+44 (0) 20 7117 2517

Alternatively, you can submit a support ticket electronically:

https://www.definitymarkets.com/support

Information contained within this manual:

- must be kept confidential,
- must not be copied or distributed and
- may be used only by users of DeFinity

No employee or other representative of DeFinity has authority to incur liability on behalf of DeFinity in relation to this document.

Our electronic trading services are supplied exclusively in accordance with DeFinity Standard Platform Terms.

About this manual

This manual provides information required by Platform Users to access electronic foreign exchange and digital assets prices streamed by DeFinity. The Introduction chapter provides an overview of the API. The subsequent chapters provide instruction on the use of the various components of the system.

We work with multiple FIX API providers. For the purpose of digital assets price distribution we will connect with Platform Users through our Talos hub and FIX API.

Other products may require connectivity to a different API which we will communicate with Platform Users. For instance, certain products may be available through Fluent Technologies or other Systems.

It is intended that this manual is read on-line in pdf format using Adobe Acrobat reader. Cross reference links are highlighted in red text and provide a means of navigating to related information. It may also be printed.

Who this manual is for

This manual is designed for use by Platform Users of DeFinity. A trader uses the system to:

View prices, spread and depth from any of the pools of liquidity

Monitor positions and exposures in real-time view

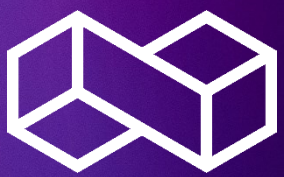
DeFinity solution overview

www.definitymarkets.com/solution

DEFINITY MARKETS® WORLD-CLASS DIGITAL ASSETS TRADING TOOLS

ELECTRONIC CRYPTO MARKETPLACE

Crypto FIX API CONNECTIVITY GUIDE



DeFinity FIX API

The FIX API provides the user with access to real-time market data subscription and orders execution. Additional data is available via the websocket API.

Connectivity

Connectivity is made to the {{ Customer }} API via the internet. The following will be provided to the client during onboarding:

- Hostname
- Port number

Configuration

{{ Customer }} acts as an acceptor for all FIX sessions.

FIX session information will be provided during onboarding:

- SenderCompID
- TargetCompID
- StartTime
- EndTime

Sessions can be configured with or without FIX session level message recovery. A typical setup includes 2 sessions:

- A FIX session for market data with recovery disabled.
- A FIX session for orders submission with recovery enabled.

Protocol

See the FIX.4.4 spec available at https://www.fixtrading.org/standards/fix-4-4/.

Timestamps

The API sends UTC timestamps with millisecond precision. For example: 20140515-19:49:56.659. The API accepts UTC timestamps with second, millisecond, or microsecond precision in the format YYYYddmm-HH:MM:SS.ffffff. For example, 20140515-19:49:56, 20140515-19:49:56.659, or 20140515-19:49:56.659441.

Market Data Requests and Updates

After FIX Logon, the client should send a MarketDataRequest to subscribe to market data snapshots. On subscription, the client can choose to subscribe to child markets individually, or subscribe to aggregated prices as price or size levels. After subscription, {{ Customer }} will stream market data full or incremental snapshot updates as the market prices change.

Market data delta updates will be supported in a future release.

Orders Submission and Updates

The orders API supports order submission. All orders can be amended and canceled via the API.

Specifying order strategy and algo parameters will be supported in a future release.

Cancel on Disconnect

By default, all open orders submitted via the FIX session will be canceled on disconnect. Cancel on Disconnect does not affect other orders for the same user that were submitted on other sessions, or orders that were explicitly placed as CancelOnDisconnect=N, see New Order Single.

Drop Copy

Sessions will receive a full drop copy for all execution reports associated with the session user. This includes orders submitted via the UI or other protocols (websocket, etc).

Header and Trailer

Standard Header

Tag	Field Name	Req'd	Comments
8	BeginString	Y	FIX.4.4. Must be the first field in the message
9	BodyLength	Y	Must be the second field in the message
35	MsgType	Y	Must be the third field in the message
49	SenderCompID	Y	Provided by {{ Customer }} during onboarding
56	TargetCompID	Y	Provided by {{ Customer }} during onboarding
34	MsgSeqNum	Y	
43	PossDupFlag	N	Always required for retransmitted messages, whether prompted by the sending system or as the result of a resend request.
97	PossResend	N	Required when the message may be duplicate of another message sent under a different sequence number.
52	SendingTime	Y	

122	OrigSendingTime	N	Required for message re-sent as a result of a ResendRequest. If data is not available set to same value as SendingTime <52>

Standard Header

Tag	Field Name	Req'd	Comments
10	CheckSum	Y	Three byte, simple checksum. See the FIX spec for details

Session Level Messages

Logon

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = A
108	HeartBtInt	Y	Note same value used by both sides
141	ResetSeqNumFlag	N	Indicates both sides of a FIX session should reset sequence numbers
96	RawData	Y	Contains signature (see below for creating a signature)
554	Password	Y	Contains the API Key for the Customer user
	<MessageTrailer>	Y	

Creating a Signature

The signature is created by concatenating

- SendingTime (52) as a string
- ASCII 01 value
- SeqNum (34) as a string
- ASCII 01 value
- SenderCompID (49)
- ASCII 01 value
- TargetCompID (56)

This string is then signed using the HMAC SHA-256 Algorithm and the API Secret for the API Key.

Sample Java code showing how this can be done programmatically. Note this sample relies on Apache Commons-Codec and QuickFIX for Java.

```
    /**
    *  Takes a Message, an apiKey and apiSecret and uses the HMAC-SHA256 algorithm to sign a FIX message by appending
    *  the  sending-time sequenceNumber, SenderCompID and TargetCompID with \u0001 separator and signing using the
    *  secret, putting it into the RawBytes in Base64 encoding.
    */
   private static void signLogon(final Message message,
                                 final SessionID sessionId,
                                 final String apiKey,
                                 final String apiSecret) throws FieldNotFound {
       final var senderCompId = sessionId.getSenderCompID();
       final var targetCompId = sessionId.getTargetCompID();
       final var sendingTime = message.getString(SendingTime.FIELD);
       final var seqNum = message.getInt(MsgSeqNum.FIELD);
       final var sep = "\\u0001";
       final var hmac = sign(apiSecret, sendingTime + sep + seqNum + sep + senderCompId + sep + targetCompId);
       message.setString(Password.FIELD, apiKey);
       message.setInt(RawDataLength.FIELD, hmac.length());
       message.setString(RawData.FIELD, hmac);
   }

   private static String sign(final String apiSecret, final String data) {
       final var mac = HmacUtils.getInitializedMac(HmacAlgorithms.HMAC_SHA_256, apiSecret.getBytes());
       final var encodedBytes = mac.doFinal(data.getBytes());
```

```
        final var encoder = Base64.getUrlEncoder();//URL Safe Base64

        return encoder.encodeToString(encodedBytes);

    }
```

Heartbeat

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = 0
112	TestReqID	Y	Required when the heartbeat is the result of a Test Request message
	<MessageTrailer>	Y	

Test Request

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = 1
112	TestReqID	Y	
	<MessageTrailer>	Y	

Resend Request

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = 2
7	BeginSeqNo	Y	
16	EndSeqNo	Y	
	<MessageTrailer>	Y	

Reject

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = 3
45	RefSeqNum	Y	MsgSeqNum of rejected message
371	RefTagID	N	The tag number of the FIX field being referenced

372	RefMsgType	N	The MsgType of the FIX message being referenced
373	SessionRejectReason	N	Code to identify reason for a session-level Reject message
58	Text	N	Where possible, message to explain reason for rejection
	<MessageTrailer>	Y	

Sequence Reset

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = 4
123	GapFillFlag	N	
36	NewSeqNo	Y	
	<MessageTrailer>	Y	

Logout

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = 5
58	Text	N	
	<MessageTrailer>	Y	

Market Data Messages

Market Data Request

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = V
262	MDReqID	Y	Must be unique, or the ID of previous Market Data Request to disable if SubscriptionRequestType(263) = 2 (Disable previous Snapshot + Updates Request).
263	SubscriptionRequestType	Y	0 = Snapshot, 1 = Snapshot + Updates (Subscribe) 2 = Unsubscribe
264	MarketDepth	Y	Depth of market for Book Snapshot / Incremental updates

			0 - full book depth 1 - top of book 2 and above - book depth (number of levels)
1070	MDQuoteType	N	Type of liquidity to be included 0 - Indicative 1 - Tradeable (i.e., *Firm*)
20020	NoSizeBucketsSegments	N	A list of sizes to return price levels for. Used to ask for prices of aggregated sizes. *Note: only supported with MDUpdateType(265) = 0 (Full refresh)*
→20021	SizeBucket	Y	Identifies a single size bucket
→55	Symbol	Y	The symbol of the currency pair to subscribe to. For example, *BTC-USD*
	<MessageTrailer>	Y	

***Note**: that exactly one of either Depth, or SizeBucket must be specified. For example, specifying both Depth and SizeBuckets will result in an error.*

Market Data Request Reject

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = Y
262	MDReqID	Y	Refers to the MDReqID of the request
58	Text	N	
	<MessageTrailer>	Y	

Market Data Snapshot Full Refresh

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = W
262	MDReqID	Y	Refers to the MDReqID of the request.
55	Symbol	Y	Symbol updated by the Market Data Entry.
268	NoMDEntries	Y	Number of entries following.
→269	MDEntryType	Y	Type of market data entry.

			0 = Bid 1 = Offer Required if NoMDEntries(268) > 0.
→270	MDEntryPx	Y	Price of the Market Data Entry.
→271	MDEntrySize	Y	Quantity represented by the Market Data Entry.
→272	MDEntryDate	Y	Date of Market Data Entry.
→273	MDEntryTime	Y	Time of Market Data Entry.
	<MessageTrailer>	Y	

Market Data Incremental Refresh

Tag	**Field Name**	**Req'd**	**Comments**
	<MessageHeader>	Y	MsgType <35> = X
262	MDReqID	Y	Refers to the MDReqID of the request.
55	Symbol	Y	Symbol updated by the Market Data Entry.
268	NoMDEntries	Y	Number of entries following.
→279	MDUpdateAction	Y	Type of Market Data update action. 0 = New 1 = Change 2 = Delete
→269	MDEntryType	Y	Type of market data entry. 0 = Bid 1 = Offer Required if NoMDEntries(268) > 0.
→270	MDEntryPx	Y	Price of the Market Data Entry.
→271	MDEntrySize	N	Quantity represented by the Market Data Entry.

→272	MDEntryDate	Y	Date of Market Data Entry.
→273	MDEntryTime	Y	Time of Market Data Entry.
	<MessageTrailer>	Y	

Orders Messages

New Order Single

The NewOrderSingle message is used to submit a new order. Market and Limit orders are currently supported.

Orders are routed using the default routing strategy. Specifying order strategy and algo parameters will be supported in a future release.

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = D
11	ClOrdID	Y	Unique identifier for Order as assigned by the client. Must be <= 36 characters. UUIDs are recommended.
55	Symbol	Y	The symbol of the currency pair to place an order on. For example, *BTC-USD*
54	Side	Y	Side of order 1 = Buy 2 = Sell
40	OrdType	Y	Order type 1 = Market 2 = Limit A = LimitAllIn, requested price/size includes fees
38	OrderQty	Y	Size of the order
15	Currency	N	The currency the quantity is specified in. If not specified, defaults to the base currency for the symbol.
44	Price	N	Price of the order. Required for limit orders.
59	TimeInForce	Y	Specifies how long the order remains in effect. 1 = Good Till Cancel (GTC)

			3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FOK)
60	TransactTime	Y	Business timestamp for this request.
126	ExpireTime	N	Optional expiration time for the order. Only valid on orders with TimeInForce=1 (GTC). UTC Timestamp.
168	EffectiveTime	N	Time at which this order will activate and begin sending orders to the market. UTC Timestamp.
20030	CancelOnDisconnect	N	Whether or not this order should be canceled when this session is disconnected, Y / N. Defaults to Y.
	<MessageTrailer>	Y	

Order Cancel Request

The OrderCancelRequest message is used to cancel an existing order.

Tag	**Field Name**	**Req'd**	**Comments**
	<MessageHeader>	Y	MsgType <35> = F
11	ClOrdID	Y	Unique ID of cancel request as assigned by the client. Must be <= 36 characters. UUIDs are recommended.
41	OrigClOrdID	Y	ClOrdID(11) of the previous non-rejected order.
55	Symbol	Y	The symbol of the currency pair of the order. For example, *BTC-USD*
54	Side	Y	Side of order 1 = Buy 2 = Sell
60	TransactTime	Y	Business timestamp for this request.
	<MessageTrailer>	Y	

Order Cancel Replace Request

The OrderCancelReplaceRequest message is used to amend the price, quantity, or markets for an existing order.

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = G
11	ClOrdID	Y	Unique ID of cancel request as assigned by the client. Must be <= 36 characters. UUIDs are recommended.
41	OrigClOrdID	Y	ClOrdID(11) of the previous non-rejected order.
55	Symbol	Y	The symbol of the currency pair of the order. For example, *BTC-USD*
54	Side	Y	Side of order 1 = Buy 2 = Sell
40	OrdType	Y	Order type 1 = Market 2 = Limit A = LimitAllIn, requested price/size includes fees
38	OrderQty	Y	Size of the order
44	Price	N	Price of the order. Required for limit orders.
60	TransactTime	Y	Business timestamp for this request.
	<MessageTrailer>	Y	

Execution Report

The ExecutionReport message is used to acknowledge/reject an order or changes to an order, and to relay order status and fill updates.

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = 8
11	ClOrdID	N	ClOrdID of the order.

41	OrigClOrdID	N	Set to the ClOrdID of the previous accepted order when responding to a Cancel or Cancel/Replace request
37	OrderID	Y	Unique identifier of most recent order as assigned by {{ Customer }}.
55	Symbol	Y	The symbol of the currency pair of the order. For example, *BTC-USD*
54	Side	Y	Side of order 1 = Buy 2 = Sell
38	OrderQty	Y	Size of the order
44	Price	N	Price of the order. Required for limit orders.
60	TransactTime	N	Business timestamp for this execution.
17	ExecID	Y	Unique identifier of execution message as assigned by {{ Customer }}.
527	SecondaryExecID	N	Unique identifier of execution message as assigned by the exchange or liquidity provider. Only specified when ExecType=Trade.
150	ExecType	Y	Describes the specific ExecutionRpt (e.g. Pending Cancel) while OrdStatus(39) will always identify the current order status (e.g. Partially Filled). 0 = New 3 = Done for day 4 = Canceled 5 = Replaced 6 = Pending Cancel (e.g. result of Order Cancel Request) 8 = Rejected A = Pending New E = Pending Replace (e.g. result of Order Cancel/Replace Request) F = Trade (partial fill or fill)
59	TimeInForce	Y	Specifies how long the order remains in effect.

			1 = Good Till Cancel (GTC) 3 = Immediate Or Cancel (IOC) 4 = Fill Or Kill (FOK)
40	OrdType	Y	Order type 1 = Market 2 = Limit
39	OrdStatus	Y	Describes the current state of the order, same scope as OrderQty, CumQty, LeavesQty, and AvgPx 0 = New 1 = Partially filled 2 = Filled 3 = Done for day 4 = Canceled 6 = Pending Cancel (i.e. result of Order Cancel Request) 8 = Rejected A = Pending New E = Pending Replace (i.e. result of Order Cancel/Replace Request)
151	LeavesQty	Y	Quantity open for further execution. If the OrdStatus(39) is = 4 (Canceled), 3 (Done For Day), or 8 (Rejected) (in which case the order is no longer active) then LeavesQty(151) could be 0, otherwise LeavesQty(151) = OrderQty(38) - CumQty(14).
14	CumQty	Y	Currently executed quantity for chain of orders.
6	AvgPx	Y	Calculated average price of all fills on this order.
103	OrdRejReason	N	For optional use with ExecType = 8 (Rejected) 0 = Broker / Exchange option 1 = Unknown symbol 2 = Exchange closed 3 = Order exceeds limit

			5 = Unknown order 6 = Duplicate Order (e.g. dupe ClOrdID) 7 = Duplicate of a verbally communicated order 8 = Stale order
31	LastPx	N	Price of this (last) fill. Required if ExecType(150) = ExecType = F (Trade)
32	LastQty	N	Quantity bought/sold on this (last) fill. Required if ExecType(150) = F (Trade)
381	GrossTradeAmt	N	Last amount traded (i.e. quantity * price) expressed in units of counter currency. Required if ExecType(150) = F (Trade)
4015	CumAmt	N	Total amount traded (i.e. quantity * price) expressed in units of counter currency.
4016	CumFee	N	Total fee expressed in units of FeeCurrency.
20030	CancelOnDisconnect	N	Whether or not this order will be canceled when this session is disconnected, Y / N
20032	DecisionStatus	N	The current decision-status of the order. A = Active (default) P = Paused U = Pending Pause R = Pending Resume W = Waiting for StartTime T = Waiting for Trigger Paused / Staged orders can be edited and resumed from the {{ Customer }} UI. Pending statuses are intermediate states.
58	Text	N	
	<MessageTrailer>	Y	

Order Cancel Reject

The OrderCancelReject message is used to reject invalid cancel or cancel/replace requests.

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = 9

11	ClOrdID	Y	ClOrdID(11) which could not be canceled/replaced.
41	OrigClOrdID	Y	ClOrdID of the previous accepted order
37	OrderID	Y	Unique identifier of most recent order as assigned by {{ Customer }}. If CxlRejReason="Unknown order", then will be "NONE".
60	TransactTime	N	Business timestamp for this reject.
39	OrdStatus	Y	Identifies current status of order. 0 = New 1 = Partially filled 2 = Filled 4 = Canceled 6 = Pending Cancel (i.e. result of Order Cancel Request) 8 = Rejected A = Pending New E = Pending Replace (i.e. result of Order Cancel/Replace Request)
434	CxlRejResponseTo	Y	Identifies the type of request that a Cancel Reject is in response to. 1 = Order cancel request 2 = Order cancel/replace request
102	CxlRejReason	N	Code to identify reason for cancel rejection. 0 = Too late to cancel 1 = Unknown order 2 = Broker / Exchange Option 3 = Order already in Pending Cancel or Pending Replace status
58	Text	N	
	<MessageTrailer>	Y	

Security List Messages

Security List Request

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = X
320	SecurityReqID	Y	Unique ID to identify this request
559	SecurityListRequestType	Y	Identifies the type of Security List Request 4 = ALL_SECURITIES
	<MessageTrailer>	Y	

Security List

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

Tag	Field Name	Req'd	Comments
	<MessageHeader>	Y	MsgType <35> = Y
320	SecurityReqID	Y	Unique ID to identify this request
322	SecurityResponseID	Y	Identifier for the SecurityList message
560	SecurityRequestResult	Y	Result of the Security Request Identified by the SecurityReqID 0 = Valid Request 1 = invalid or Unsupported Request 2 = No instruments found that match the selection criteria 3 = Not authorized to retrieve instrument data 4 = Instrument data temporarily unavailable 5 = Request for instrument data not supported
393	TotalNoRelatedSym	N	Used to indicate the total number of securities being returned for this request. Used in the event that message fragmentation is required
893	LastFragment	N	Indicates whether this is the last fragment in a sequence of message fragments. Only required where message has been fragmented
146	NoRelatedSym		
	Component Block <Instrument>	Y if 560=0	

→55	Symbol	Y	Common “human understood” representation of the security
→48	SecurityID	Y	Internal {{ Customer }} ID for Security - for information only
→561	RoundLot	Y	The trading lot size of a security
→562	MinTradeVol	Y	Minimum Trading Volume for a security
→969	MinPriceIncrement	Y	The smallest difference between quoted prices for this instrument
→1140	MaxTradeVol	Y	Maximum Size for an order on this security
	<MessageTrailer>	Y	

Order State Change Matrices

Filled order

Time	Request (ClOrdID,OrigClOrdID)	Response (ClOrdID,OrigClOrdID)	ExecType	OrdStatus	OrderQty	CumQty	LeavesQty	LastQty	Comment
1	NewOrderSingle(A)				10				
2		ExecutionReport(A)	PendingNew	PendingNew	10	0	10		
3		*ExecutionReport(A)*	*Rejected*	*Rejected*	*10*	*0*	*10*		*If order is rejected*
3		ExecutionReport(A)	New	New	10	0	10		
4		ExecutionReport(A)	Trade	PartiallyFilled	10	2	8	2	
5		ExecutionReport(A)	Trade	PartiallyFilled	10	3	7	1	
6		ExecutionReport(A)	Trade	Filled	10	10	0	7	

Canceled order

Time	Request (ClOrdID,OrigClOrdID)	Response (ClOrdID,OrigClOrdID)	ExecType	OrdStatus	OrderQty	CumQty	LeavesQty	LastQty	Comment
1	NewOrderSingle(A)				10				
2		ExecutionReport(A)	PendingNew	PendingNew	10	0	10		
3		*ExecutionReport(A)*	*Rejected*	*Rejected*	*10*	*0*	*10*		*If order is rejected*
3		ExecutionReport(A)	New	New	10	0	10		

4	OrderCancelRequest(B,A)								
5		*OrderCancelRejected(B,A)*		*New*	*10*	*0*	*10*		*If cancel is rejected*
5		ExecutionReport(B,A)	PendingCancel	PendingCancel	10	0	10		
6		*OrderCancelRejected(B,A)*		*New*	*10*	*0*	*10*		*If cancel is rejected*
7		ExecutionReport(B,A)	Canceled	Canceled	10	0	0		

Replace to increase quantity

Time	Request (ClOrdID,OrigClOrdID)	Response (ClOrdID,OrigClOrdID)	ExecType	OrdStatus	OrderQty	CumQty	LeavesQty	LastQty	Comment
1	NewOrderSingle(A)				10				
2		ExecutionReport(A)	PendingNew	PendingNew	10	0	10		
3		ExecutionReport(A)	New	New	10	0	10		
4	OrderCancel/ ReplaceRequest(B,A)				11				
5		*OrderCancelRejected(B,A)*		*New*	*10*	*0*	*10*		*If replace is rejected*
5		ExecutionReport(B,A)	PendingReplace	PendingReplace	10	0	10		
6		*OrderCancelRejected(B,A)*		*New*	*10*	*0*	*10*		*If replace is rejected*
6		ExecutionReport(B,A)	Replaced	New	11	0	0		
7		ExecutionReport(B)	Trade	PartiallyFilled	11	1	10	1	

Replace during fill

Time	Request (ClOrdID,OrigClOrdID)	Response (ClOrdID,OrigClOrdID)	ExecType	OrdStatus	OrderQty	CumQty	LeavesQty	LastQty	Comment
1	NewOrderSingle(A)				10				
2		ExecutionReport(A)	PendingNew	PendingNew	10	0	10		
3		ExecutionReport(A)	New	New	10	0	10		
4	OrderCancel/ ReplaceRequest(B,A)				8				
5		ExecutionReport(A)	Trade	PartiallyFilled	10	1	9	1	Fill before replace is received
6		ExecutionReport(B,A)	PendingReplace	PendingReplace	10	1	9		

7		ExecutionReport(A)	Trade	PendingReplace	10	3	7	2	Fill before replace is processed
8		ExecutionReport(B,A)	Replaced	New	8	3	5		
9		ExecutionReport(B)	Trade	Filled	8	8	0	5	

Example Messages

Login

Logon

```
8=FIX.4.4^A9=143^A35=A^A34=1^A49=CUSTOMER^A52=20220915-18:29:58.756^A56={{
Customer
}}^A95=44^A96=ZduZiNxyxS7_4UPDesOryd9KVEecg9LAqqTRR79Pp20=^A98=0^A108=30000
0^A141=Y^A554=Daniel^A10=248^A
```

Logon Response

```
8=FIX.4.4^A9=78^A35=A^A34=1^A49={{ Customer }}^A52=20220915-
18:29:58.765^A56=CUSTOMER^A98=0^A108=300000^A141=Y^A10=255^A
```

Security List

SecurityList Request

```
8=FIX.4.4^A9=88^A35=x^A34=3^A49=CUSTOMER^A52=20220930-13:28:16.451^A56={{
Customer }}^A320=id-220929160342682-27^A559=4^A10=156^A
```

SecurityList Response

```
8=FIX.4.4^A9=243^A35=y^A34=3^A49={{ Customer }}^A52=20220930-
13:28:16.514^A56=CUSTOMER^A146=2^A55=ETH-
USD^A48=2^A561=0.00000001^A562=0.00000001^A969=0.00001^A1140=1000.00^A55=BT
C-USD^A48=24^A561=0.00000001^A562=0^A969=0.01^A1140=1000.00^A320=id-
220929160342682-27^A322=1BQM22QR01G00^A560=0^A10=153^A
```

Orders

NewOrderSingle

```
8=FIX.4.4^A9=147^A35=D^A34=2^A49=CUSTOMER^A52=20220915-18:30:01.335^A56={{
Customer }}^A11=id-220912164936074-
1152^A38=1^A40=2^A44=1630.123^A54=2^A55=ETH-USD^A59=1^A60=20220915-
14:30:01^A10=228^A
```

ExecutionReport (Reject)

```
8=FIX.4.4^A9=366^A35=8^A34=2^A49={{ Customer }}^A52=20220915-
18:30:01.393^A56=CUSTOMER^A1=default^A6=0^A11=id-220912164936074-
1152^A14=0^A17=ba22dbe1-cb33-4fcb-9613-921006afdebd^A37=94869628-b862-43e4-
bd2d-be85e4295bd4^A38=1.00000000^A39=8^A40=2^A44=1630.123^A54=2^A55=ETH-
USD^A58=Unable to submit market order - please try again
later.^A59=1^A60=20220915-
14:30:01.000^A103=0^A150=8^A151=1.00000000^A20030=N^A20032=A^A10=205^^A
```

Market Data

MarketDataSnapshot Request

```
8=FIX.4.4^A9=186^A35=V^A34=2^A49=CUSTOMER_MD^A52=20220930-
13:28:06.146^A56={{ Customer }}^A262=id-220929160342682-
26^A263=1^A264=0^A265=0^A1070=1^A146=1^A55=ETH-
USD^A267=2^A269=0^A269=1^A20020=4^A20021=1^A20021=2^A200
21=5^A20021=10^A10=226^A
```

MarketDataSnapshot Response

```
8=FIX.4.4^A9=608^A35=W^A34=2^A49={{ Customer }}^A52=20220930-
13:28:06.454^A56=CUSTOMER_MD^A55=ETH-USD^A262=id-220929160342682-
26^A268=8^A269=0^A270=1320.43^A271=1.00000000^A272=20220930^A273=13:28:06.4
01^A269=0^A270=1
320.43^A271=2.00000000^A272=20220930^A273=13:28:06.401^A269=0^A270=1320.43^
A271=5.00000000^A272=20220930^A273=13:28:06.401^A269=0^A270=1320.43^A271=10
.00000000^A272=20220930^A273=13:28:06.401^A269=1^A270=1323
.42^A271=1.00000000^A272=20220930^A273=13:28:06.401^A269=1^A270=1323.42^A27
1=2.00000000^A272=20220930^A273=13:28:06.401^A269=1^A270=1323.42^A271=5.000
00000^A272=20220930^A273=13:28:06.401^A269=1^A270=1323.42
^A271=10.00000000^A272=20220930^A273=13:28:06.401^A10=247^A
```